Міністерство освіти і науки України
Університет митної справи та фінансів

ТЕРМІНОЛОГІЧНИЙ АНАЛІЗ ТА КЛАСИФІКАЦІЯ ПОНЯТЬ У НЕЙРОМЕРЕЖЕВИХ МЕТОДАХ УПРАВЛІННЯ РИЗИКАМИ КІБЕРФІЗИЧНИХ СИСТЕМ

Прокопович-Ткаченко Дмитро Ігорович

Монографія

Дніпро
2025

УДК 004.891.2:004.056:62-50

Рекомендовано до друку
вченою радою Університету митної справи та фінансів
(протокол № 6 від 29.12.2025 р.)

Рецензенти:
Ігор Рубан, доктор технічних наук, професор, ректор Харківського національного університету радіоелектроніки;
Вячеслав Харченко, завідувач кафедри комп'ютерних систем, мереж і кібербезпеки Національного аерокосмічного університету "Харківський авіаційний інститут", Лауреат Державної премії України у галузі науки і техніки, заслужений винахідник України, член-кореспондент НАН України

Автор: Прокопович-Ткаченко Дмитро Ігорович, кандидат технічних наук, доцент, завідувач кафедри кібербезпеки та інформаційних технологій Університету митної справи та фінансів, старший науковий співробітник Державної наукової установи «Інститут інформаційної безпеки і права» Національної академії правових наук України

Термінологічний аналіз та класифікація понять у нейромережевих методах управління ризиками кіберфізичних систем / Прокопович-Ткаченко Д. І. / за ред. О. Корченка, д. т. н., проф., члена-кореспондента НАН України, Заслуж. діяча науки і техніки України, лауреата Державної премії України в галузі науки і техніки, першого проректора Державного університету інформаційно-комунікаційних технологій. Дніпро : Університет митної справи та фінансів, 2025. 351 с.

ISBN 978-966-328-254-1

Метою цієї монографії є комплексний аналіз і систематизація термінів і понять, які використовуються у нейромережевих методах управління ризиками кіберфізичних систем, а також розробка рекомендацій щодо їх стандартизації та уніфікації.

ISBN 978-966-328-254-1

ЗМІСТ

ПЕРЕЛІК НЕРОЗШИФРОВАНИХ СКОРОЧЕНЬ

Таблиця 1

Перелік нерозшифрованих скорочень – Частина 1.1 (A–G)

Абревіатура	Розділ	Розшифрування
ABBR	Розділ 1	Abbreviation – абревіатура
ACCESS	Вступ	Advanced Computer Control for Energy and System Security
ACM	Розділ 1	Association for Computing Machinery – Асоціація обчислювальної техніки
ACM/IEEE	Вступ	ACM / Institute of Electrical and Electronics Engineers
AD	Розділ 1	Active Directory – служба каталогів
ADAM	Розділ 1	Adaptive Moment Estimation – метод оптимізації Adam
ADS	Розділ 1	Anomaly Detection System – система виявлення аномалій
AE	Розділ 1	Autoencoder – автоенкодер
AE/VAE	Вступ	Autoencoder / Variational Autoencoder
AI	Вступ, Розділ 1, Розділ 2	Artificial Intelligence – штучний інтелект
AINS	Вступ	Artificial Immune System – штучна імунна система
AMI	Розділ 1	Advanced Metering Infrastructure – інфраструктура інтелектуального обліку
ANSI	Розділ 2	American National Standards Institute
API	Вступ, Розділ 1, Розділ 2	Application Programming Interface
APK	Вступ	Android Package
APPROVED	Розділ 1	Статус «схвалено» у специфікаціях
APT	Розділ 1	Advanced Persistent Threat
AR	Вступ	Augmented Reality
ASIC	Розділ 1	Application-Specific Integrated Circuit
ASOC	Вступ	Adaptive System on Chip
AWS	Вступ	Amazon Web Services
BERT	Розділ 1	Bidirectional Encoder Representations from Transformers

Продовження таблиці 1

BI	Вступ	Business Intelligence
BMC	Розділ 1	Baseboard Management Controller
BMS	Розділ 1, Розділ 2	Building Management System
BOW	Вступ	Bag of Words
CAN	Вступ, Розділ 1	Controller Area Network
CARPATHIANCC	Вступ	Carpathian Communication Conference
CCTA	Вступ, Розділ 1	Central Computer and Telecommunications Agency
CDSS	Вступ, Розділ 1	Clinical Decision Support System
CICIDS/UNSW	Вступ	CICIDS / UNSW – набори даних
CIO	Розділ 2	Chief Information Officer
CISA	Розділ 2	Cybersecurity and Infrastructure Security Agency
CMMC	Розділ 2	Cybersecurity Maturity Model Certification
CNC	Розділ 1	Computer Numerical Control
COLING	Розділ 1	International Conference on Computational Linguistics
CPS	Розділ 2	Cyber-Physical Systems

Перелік нерозшифрованих скорочень – Частина 1.2 (A–G)

Абревіатура	**Розділ**	**Розшифрування**
CRC	Вступ	Cyclic Redundancy Check
CSIRT	Розділ 1	Computer Security Incident Response Team
CSIT	Вступ	Computer Science and Information Technology
CSRC	Розділ 2	Computer Security Resource Center (NIST)
CSV	Розділ 1	Comma-Separated Values
CT	Розділ 1	Computed Tomography / Critical Technology
DAC	Розділ 1	Discretionary Access Control
DBN	Розділ 1	Deep Belief Network
DBSCAN	Вступ	Density-Based Spatial Clustering with Noise
DCS	Розділ 1	Distributed Control System
DDS	Вступ	Data Distribution Service

Продовження таблиці 1

DER	Вступ, Розділ 1	Distributed Energy Resources
DL	Розділ 1	Deep Learning
DOI	Розділ 1, Розділ 2	Digital Object Identifier
DSM	Вступ, Розділ 1	Demand Side Management
DSTU	Розділ 1	ДСТУ – Державний стандарт України
DT	Вступ, Розділ 1	Digital Twin
DSS	Вступ	Decision Support System
EHR	Вступ, Розділ 1	Electronic Health Record
EMS	Розділ 1	Energy Management System
EN	Розділ 2	European Norm
ENISA	Вступ, Розділ 1	European Union Agency for Cybersecurity
ERP	Розділ 1	Enterprise Resource Planning
ETSI	Розділ 2	European Telecommunications Standards Institute
FHIR	Розділ 1	Fast Healthcare Interoperability Resources
FMS	Розділ 1	Fleet Management System
FPGA	Розділ 1	Field-Programmable Gate Array
GDPR	Вступ, Розділ 1	General Data Protection Regulation
GPS	Розділ 1	Global Positioning System
GRU	Вступ, Розділ 1	Gated Recurrent Unit

Перелік нерозшифрованих скорочень – Частина 1.3 (A–G)

Абревіатура	**Розділ**	**Розшифрування**
HMI	Розділ 1	Human-Machine Interface
HTTP	Вступ, Розділ 1	Hypertext Transfer Protocol

Перелік нерозшифрованих скорочень – Частина 2.1 (H–N)

Абревіатура	**Розділ**	**Розшифрування**
IAM	Вступ	Identity and Access Management
ICM	Вступ	Integrated Computational Materials / Industrial Control Model
ICMLA	Розділ 1	International Conference on Machine Learning and Applications

Продовження таблиці 1

ICS	Розділ 2	Industrial Control System
IDF	Розділ 1	Intermediate Distribution Frame / Israel Defense Forces
IDMZ	Вступ	Industrial Demilitarized Zone
IDS	Вступ, Розділ 1	Intrusion Detection System
IDS/IPS	Вступ	Intrusion Detection / Prevention System
IEC	Вступ, Розділ 1, Розділ 2	International Electrotechnical Commission
IEEE	Вступ, Розділ 1, Розділ 2	Institute of Electrical and Electronics Engineers
IETF	Розділ 1, Розділ 2	Internet Engineering Task Force
IP	Вступ, Розділ 2	Internet Protocol
IP/IDMZ	Вступ	Internet Protocol / Industrial Demilitarized Zone
IPS	Розділ 1	Intrusion Prevention System
ISMS	Розділ 1	Information Security Management System
ISO	Вступ, Розділ 1, Розділ 2	International Organization for Standardization
ISO/IEC	Вступ, Розділ 1, Розділ 2	Joint standards of ISO and IEC
ISO/NIST	Вступ, Розділ 1	Joint approaches of ISO and NIST
IT	Вступ	Information Technology
ITS	Розділ 1	Intelligent Transportation Systems
ITU	Розділ 1, Розділ 2	International Telecommunication Union
JAMA	Розділ 1	Journal of the American Medical Association
JMIR	Розділ 1	Journal of Medical Internet Research
JSON	Розділ 1, Розділ 2	JavaScript Object Notation
JSON/CSV	Розділ 1, Розділ 2	Combined JSON and CSV formats
JWT	Розділ 1, Розділ 2	JSON Web Token
JTC	Вступ, Розділ 1	Joint Technical Committee (ISO/IEC)
KPI	Розділ 1	Key Performance Indicator

Продовження таблиці 1

LAN	Розділ 2	Local Area Network
LSTM	Вступ	Long Short-Term Memory
MAGERIT	Розділ 1	Spanish methodology for risk analysis and management
MCU	Розділ 1	Microcontroller Unit
MEDINFO	Розділ 1	International Conference on Medical and Health Informatics
MEDLINE	Розділ 1	Medical Literature Analysis and Retrieval System Online
MES	Вступ, Розділ 1	Manufacturing Execution System
MES/ERP	Вступ	Manufacturing Execution System / Enterprise Resource Planning
MFA	Розділ 1, Розділ 2	Multi-Factor Authentication
MITM	Вступ, Розділ 1	Man-in-the-Middle
ML	Розділ 1	Machine Learning
MQTT	Вступ, Розділ 1, Розділ 2	Message Queuing Telemetry Transport
MQTT/AMQP	Вступ	MQTT / Advanced Message Queuing Protocol
MSE	Розділ 1	Mean Squared Error
MVP	Розділ 1	Minimum Viable Product
NFC	Розділ 1	Near Field Communication
NICCS	Розділ 2	National Initiative for Cybersecurity Careers and Studies
NIS	Вступ, Розділ 1	Network and Information Security Directive (EU)
NIST	Вступ, Розділ 1, Розділ 2	National Institute of Standards and Technology (US)
NTP	Розділ 1	Network Time Protocol

Перелік нерозшифрованих скорочень – Частина 3 (O–Z)

Абревіатура	**Розділ**	**Розшифрування**
OMG	Розділ 1	Object Management Group – консорціум OMG
OPC	Вступ, Розділ 1, Розділ 2	OLE for Process Control – промисловий стандарт

Продовження таблиці 1

OWE	Розділ 2	Opportunistic Wireless Encryption
PACS	Розділ 1	Picture Archiving and Communication System
PKI	Вступ, Розділ 1, Розділ 2	Public Key Infrastructure
PLM	Розділ 1	Product Lifecycle Management
PLC	Розділ 2	Programmable Logic Controller
PLC/RTU	Вступ, Розділ 1	PLC / Remote Terminal Unit
PMU	Розділ 1	Phasor Measurement Unit
POODLE	Розділ 2	Padding Oracle On Downgraded Legacy Encryption
PROFINET	Вступ	Industrial Ethernet Standard (Siemens)
QIP	Розділ 2	Quality Improvement Program
QJM	Розділ 1	Quarterly Journal of Medicine
RDF	Розділ 1	Resource Description Framework
RDF/SKOS	Розділ 1	RDF / Simple Knowledge Organization System
REST	Розділ 1	Representational State Transfer
RMLR	Розділ 1	Risk-Managed Learning Rate
RNN	Вступ, Розділ 1	Recurrent Neural Network
RTOS	Розділ 1	Real-Time Operating System
RTU	Розділ 1	Remote Terminal Unit
SAFE	Розділ 1	Security Assessment Framework
SCADA	Розділ 2	Supervisory Control And Data Acquisition
SCADA/HMI	Вступ	SCADA / Human-Machine Interface
SCADA/MES	Вступ	SCADA / Manufacturing Execution System
SDK	Розділ 1	Software Development Kit
SG	Розділ 1	Smart Grid
SIEM	Вступ, Розділ 1	Security Information and Event Management
SKOS	Розділ 1	Simple Knowledge Organization System
SME	Розділ 1	Subject Matter Expert / Small and Medium Enterprise
SNOMED	Розділ 2	Systematized Nomenclature of Medicine

Продовження таблиці 1

SOC	Вступ, Розділ 1	Security Operations Center
SP	Вступ, Розділ 1, Розділ 2	Service Provider / Security Policy
SPARQL	Розділ 1	SPARQL Protocol and RDF Query Language
SSL	Вступ	Secure Sockets Layer
TF	Розділ 1	TensorFlow
TIL	Розділ 2	Today I Learned
TMS	Вступ, Розділ 1	Transportation Management System
TLS	Розділ 2	Transport Layer Security
TLS/VPN	Вступ	TLS / Virtual Private Network
TS	Розділ 2	Technical Specification
TSDB	Вступ	Time Series Database
TSN	Розділ 1	Time-Sensitive Networking
TTL	Розділ 1	Time To Live
UA	Розділ 1	Unified Architecture
UA/JSON	Вступ	Unified Architecture / JSON
UI	Розділ 1	User Interface
URI	Розділ 1	Uniform Resource Identifier
URI/DOI	Розділ 1	URI / Digital Object Identifier
URL	Розділ 2	Uniform Resource Locator
URN	Розділ 1	Uniform Resource Name
VAE	Розділ 1	Variational Autoencoder
VPN	Розділ 2	Virtual Private Network
WAN	Вступ, Розділ 1	Wide Area Network
WEB	Розділ 1	World Wide Web
XAI	Розділ 1	Explainable Artificial Intelligence
ZTA	Розділ 1	Zero Trust Architecture
ЄС	Вступ, Розділ 2	Європейський Союз
ІБ	Вступ, Розділ 1	Інформаційна безпека
ІТ	Вступ, Розділ 1	Інформаційні технології

Закінчення таблиці 1

АСОЕ	Вступ, Розділ 1	Автоматизована система обробки енергії
ДК	Розділ 2	Державний комітет
ДП	Вступ, Розділ 1	Державне підприємство
ДСТУ	Розділ 2	Державний стандарт України
КФС	Розділ 2	Кіберфізичні системи
КФС/ШНМ	Розділ 1	Кіберфізичні системи / Штучні нейронні мережі
ООН	Вступ, Розділ 1	Організація Об’єднаних Націй
ПЗ	Вступ	Програмне забезпечення
ПЛК	Вступ	Програмований логічний контролер
РОЗДІЛ	Розділ 1	Позначення структурного розділу
США	Розділ 2	Сполучені Штати Америки
ТОВ	Розділ 1	Товариство з обмеженою відповідальністю
ЦОД	Вступ	Центр обробки даних
ШІ	Розділ 1	Штучний інтелект
ШНМ	Вступ, Розділ 1	Штучні нейронні мережі

ВСТУП

Актуальність дослідження

На сучасному етапі розвитку технологій кіберфізичні системи (КФС) є ключовими елементами цифровізації промисловості, енергетики, медицини та транспорту. Вони формують нову парадигму функціонування критичної інфраструктури [1, 2]. Глибока інтеграція фізичних та кібернетичних компонентів дозволяє таким системам здійснювати безперервне керування складними процесами в реальному часі, підвищуючи їхню ефективність, надійність та оперативність реагування на будь-які зміни зовнішніх умов [3, 4]. Водночас стрімке поширення КФС супроводжується зростанням ризиків, пов’язаних із кіберзагрозами. Складність архітектури таких систем значно розширює можливі вектори атак, підвищуючи вразливість до цілеспрямованих і масованих кібердарів [5, 6].

Зокрема, проблемою є асиметричність загроз, мультиплатформність і різнорідність компонентів КФС, а також необхідність постійної роботи систем у реальному часі, що унеможливлює зупинку для усунення наслідків атак [7, 8]. Унаслідок цього виникає нагальна потреба у розробці та впровадженні ефективних методів управління ризиками, здатних забезпечувати високий рівень кібербезпеки таких систем. Останні наукові дослідження підкреслюють особливе значення нейронних мереж та інших технологій штучного інтелекту (ШІ) у розв’язанні цієї проблеми. Зокрема, вони забезпечують можливість швидкого аналізу великих обсягів даних, оперативне виявлення аномалій і потенційних загроз, а також прогнозування ризиків [9, 10].

Утім, ефективність впровадження нейромережевих рішень суттєво залежить від узгодженості термінологічної бази, що наразі відсутня. Наявність неоднозначностей у тлумаченні ключових понять значно ускладнює комунікацію між фахівцями, гальмує розвиток нормативно-правового регулювання та стандартизацію в галузі кіберфізичних систем [11, 12].

Таким чином, особливо актуальним є завдання розробки гармонізованого термінологічного підґрунтя, що сприятиме ефективнішій інтеграції сучасних технологій у практичну діяльність і забезпечить узгоджене застосування нейромережевих методів управління ризиками у КФС [13, 14].

Мета і завдання монографії

Метою цієї монографії є комплексний аналіз і систематизація термінів та понять, які використовуються у нейромережевих методах управління ризиками кіберфізичних систем, а також розробка рекомендацій щодо їх стандартизації та уніфікації.

Основні завдання монографії

Проведення аналізу наявних визначень і термінів у сфері кіберфізичних систем та штучного інтелекту з використанням міжнародного досвіду та стандартів ISO, IEC, NIST [15, 16].

Розробка класифікаційної структури основних понять та ієрархічних взаємозв'язків між ними, що дозволить забезпечити системність і зрозумілість термінологічної бази [17, 18].

Дослідження міжнародних підходів до стандартизації термінології та їх адаптація до умов української нормативно-правової та науково-технічної бази [19, 20].

Наукове обґрунтування ефективності застосування нейронних мереж у процесах оцінки ризиків та управління безпекою КФС [21, 22].

Формулювання конкретних рекомендацій щодо впровадження узгоджених термінів у нормативні документи та освітні програми [23, 24].

Методологія дослідження

Методологічна основа монографії включає комплексний підхід, що поєднує теоретичні, емпіричні та лінгвістичні методи дослідження. Теоретичні методи передбачають глибокий аналіз наукових джерел і міжнародних стандартів у сфері кіберфізичних систем, штучного інтелекту та кібербезпеки. Використано контент-аналіз і бібліометричний аналіз для виявлення тенденцій розвитку термінології та класифікації ризиків [25, 26]. Емпіричні методи включають аналіз реальних кейсів використання нейромереж у КФС, проведення експертних опитувань для уточнення та перевірки термінів і понять, а також імітаційне моделювання сценаріїв кібератак із використанням спеціалізованого програмного забезпечення (MATLAB, Python, Tensor Flow) [27, 28]. Лінгвістичний аналіз спрямований на систематизацію та узгодження термінологічної бази шляхом порівняння й гармонізації понять, що використовуються в різних наукових школах і міжнародних стандартах. Це дозволить уникнути семантичних неоднозначностей і забезпечити цілісність понятійного апарату [29, 30].

Наукова новизна та практичне значення

Наукова новизна отриманих результатів полягає у створенні уніфікованої системи термінів і понять, що стосуються нейромережевих методів управління ризиками кіберфізичних систем. Уперше запропоновано інтегровану класифікацію понять з урахуванням міжнародних стандартів та української специфіки [31, 32]. Розроблено нові критерії класифікації загроз і ризиків та запропоновано концепцію динамічного ризик-профілю для оперативного аналізу ситуації у реальному часі [33, 34]. Практичне значення дослідження полягає в

упровадженні результатів у наукові проєкти, практичну діяльність підприємств, освітні програми та нормативні документи. Запропонований глосарій і рекомендації щодо використання термінів дозволять забезпечити єдину термінологічну базу для ефективної комунікації між фахівцями різних галузей, що суттєво підвищить якість та оперативність ухвалення рішень у сфері безпеки кіберфізичних систем [35, 36].

Відтак результати монографії забезпечують науково-методичне підґрунтя для подальших досліджень та ефективної стандартизації у сфері кіберфізичних систем, що має важливе значення як для національної безпеки, так і для інтеграції України у міжнародний науково-технічний простір [37, 38].

Джерела:

1. AlHarmali, A., Ali, S., Aman, W., Hussain, O. Cyber Risk Assessment for Cyber-Physical Systems: A Review of Methodologies and Recommendations for Improved Assessment Effectiveness. CSIT. 2024. DOI: 10.5121/csit.2024.141608.
2. Gao, X., Ali, M., Sun, W. A Risk Assessment Framework for Cyber-Physical Security in Distribution Grids with Grid-Edge DERs. Energies. 2024. №17(7). DOI: 10.3390/en17071587.
3. Cassottana, B., Roomi, M. M., Mashima, D., Sansavini, G. Resilience Analysis of Cyber-Physical Systems: A Review of Models and Quantitative Assessment. Risk Analysis. 2023. DOI: 10.1111/risa.14089.
4. Amro, A., Gkioulos, V. Evaluation of a Cyber Risk Assessment Approach for Cyber-Physical Systems: Maritime- and Energy-Use Cases. Journal of Marine Science and Engineering. 2023. №11(4). DOI: 10.3390/jmse11040744.
5. Kayan, H., Nunes, M., Rana, O., Burnap, P., Perera, C. Cybersecurity of Industrial Cyber-Physical Systems: A Review. arXiv preprint, 2021.
6. Lun, Y. Z., D'Innocenzo, A., Malavolta, I., Di Benedetto, M. D. Cyber-Physical Systems Security: A Systematic Mapping Study. arXiv preprint, 2016.
7. Neural Network Approach to Assessing Cybersecurity Risks in Large-Scale Systems. IEEE/ACM. 2019–2020. DOI: 10.1145/ 3433174.3433603.
8. Safety and Security Risk Assessment in Cyber-Physical Systems. IET Cyber-Physical Systems: Theory & Applications. 2018. DOI: 10.1049/ iet-cps.2018.5068.
9. Model-Based Risk Assessment for Cyber Physical Systems Security. Computers & Security. 2020. DOI: 10.1016/j.cose.2020.101930.
10. Assessing Cyber Risk in Cyber-Physical Systems Using Enhanced FMECA-ATT&CK Approach. ACM Digital Library. 2023. DOI: 10.1145/3571733.
11. Simeone, A. A Human-Cyber-Physical System for Operator 5.0 Smart Risk Assessment. International Journal of Advanced Manufacturing Technology. 2023. DOI: 10.1007/s00170-023-12012-5.

12. Feng, C., Tian, P. Time Series Anomaly Detection for Cyber-Physical Systems via Neural System Identification and Bayesian Filtering. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD). 2021. DOI: 10.1145/3447548.3467238.

13. Paredes, C. M., Martínez-Castro, D., Ibarra-Junquera, V., González-Potes, A. Detection and Isolation of DoS and Integrity Cyber Attacks in Cyber-Physical Systems with a Neural Network-Based Architecture. Electronics. 2021. Vol. 10, No. 18. Article 2238. DOI: 10.3390/electronics10182238.

14. Abshari, D., Sridhar, M. A Survey of Anomaly Detection in Cyber-Physical Systems. arXiv preprint. 2025. DOI: 10.48550/arXiv.2502.13256.

15. Moriano, P., Hespeler, S. C., Li, M., Mahbub, M. Adaptive Anomaly Detection for Identifying Attacks in Cyber-Physical Systems: A Systematic Literature Review. Soft Computing. 2025. DOI: 10.1007/s10462-025-11292-w.

16. Luo, Y., Xiao, Y., Cheng, L., Yao, D. D. Deep Learning-Based Anomaly Detection in Cyber-Physical Systems: Progress and Opportunities. arXiv preprint. 2020. DOI: 10.48550/arXiv.2003.13213.

17. Gao, X., B., D. Deep Learning-Driven Anomaly Detection for Cyber-Physical Systems: Enhancing Real-Time Security and Resilience. International Research Journal of Innovations in Engineering and Technology. 2024. DOI: 10.47001/IRJIET/2023.706034.

18. Gao, X., Liang, W., Zhou, X. Deep Learning-Based Cyber-Physical Feature Fusion for Anomaly Detection. Mathematics. 2022. Vol. 10, Issue 22. Article 4373. DOI: 10.3390/math10224373.

19. Jadidi, Z., Pal, S., Nayak, N. N. Security of Machine Learning-Based Anomaly Detection in Cyber-Physical Systems. arXiv preprint. 2022. DOI: 10.48550/arXiv.2206.05678.

20. Ullah, I., Mahmoud, Q. H. IADF-CPS: Intelligent Anomaly Detection Framework Towards Cyber-Physical Systems. Computer Communications. 2022. Vol. 188. P. 81–89. DOI: 10.1016/j.comcom.2022.02.022.

21. Jeffrey, N., Tan, Q., Villar, J. R. A Hybrid Methodology for Anomaly Detection in Cyber-Physical Systems. Neurocomputing. 2024. Vol. 568. Art. 127068. DOI: 10.1016/j.neucom.2023.127068.

22. Schneider, P., Böttinger, K. High-Performance Unsupervised Anomaly Detection for Cyber-Physical System Networks. Proceedings of ACM Conference. 2019. DOI: 10.1145/3359789.3359798.

23. Duo, W. L., Zhou, M. C., Abusorrah, A. A Survey of Cyber Attacks on Cyber-Physical Systems: Recent Advances and Challenges. IEEE/CAA Journal of Automatica Sinica. 2022. Vol. 9, Issue 5. P. 721–741. DOI: 10.1109/JAS.2022.105548.

24. Humayed, A., Lin, J., Li, F., Luo, B. Cyber-Physical Systems Security – A Survey. IEEE Internet of Things Journal. 2017. Vol. 4, Issue 6. P. 1802–1831. DOI: 10.1109/JIOT.2017.2703172.

25. Kayan, H., Nunes, M., Rana, O., Burnap, P., Perera, C. Cybersecurity of Industrial Cyber-Physical Systems: A Review. ACM Computing Surveys (CSUR). 2022. Vol. 54, Issue 11. Art. 232. DOI: 10.1145/3433174.3433603.

26. Segovia-Ferreira, M., Cassottana, B., Sansavini, G. Resilience Analysis of Cyber-Physical Systems: A Review of Models and Quantitative Assessment. Risk Analysis. 2023. Vol. 43, Issue 6. P. 1187–1207. DOI: 10.1111/risa.14089.

27. Reeja, Y., Kumar, R. Deep Learning Method for Anomaly Detection in CPS. Taylor & Francis Book Chapter. 2021. DOI: 10.1201/9781003176309-7.

28. Feng, C., Tian, P. Neural System Identification and Bayesian Filtering for Time Series Anomaly Detection in CPS. KDD 2021 Conference Proceedings. 2021. DOI: 10.1145/3447548.3467238.

29. Paredes, C. M., Martínez-Castro, D., Ibarra-Junquera, V., González-Potes, A. Detection and Isolation of DoS and Integrity Cyber Attacks in CPS Using Neural Networks. Electronics. 2021. Vol. 10, No. 18, 2238. DOI: 10.3390/electronics10182238.

30. Abshari, D., Sridhar, M. Anomaly Detection Methods in Cyber-Physical Systems: Comprehensive Review. arXiv preprint. 2025. DOI: 10.48550/arXiv.2502.13256.

31. Moriano, P., Hespeler, S. C., Li, M., Mahbub, M. Adaptive Anomaly Detection in Cyber-Physical Systems: Literature Review. Soft Computing. 2025. DOI: 10.1007/s10462-025-11292-w.

32. Luo, Y., Xiao, Y., Cheng, L., Yao, D. D. Deep Learning-Based Anomaly Detection in CPS: Current Progress and Opportunities. arXiv preprint. 2020. DOI: 10.48550/arXiv.2003.13213.

33. Gao, X., Liang, W., Zhou, X. Deep Learning-Based Cyber-Physical Feature Fusion for CPS Anomaly Detection. Mathematics. 2022. Vol. 10, Issue 22. Art. 4373. DOI: 10.3390/math10224373.

34. Ullah, I., Mahmoud, Q. H. Intelligent Anomaly Detection Framework Towards Cyber-Physical Systems. Computer Communications. 2022. Vol. 188. P. 81–89. DOI: 10.1016/j.comcom.2022.02.022.

35. Jadidi, Z., Pal, S., Nayak, N. N. Security Considerations in Machine Learning-Based CPS Anomaly Detection. arXiv preprint. 2022. DOI: 10.48550/arXiv.2206.05678.

36. Simeone, A., Caggiano, A., Caiazzo, F. A Human-Cyber-Physical System for Operator 5.0 Smart Risk Assessment. International Journal of Advanced Manufacturing Technology. 2023. Vol. 131, Issue 3–4. P. 1201–1225. DOI: 10.1007/s00170-023-12012-5.

37. IEEE-ACM. Neural Network Approach to Cybersecurity Risk Assessment in Large-Scale Systems. ACM Computing Surveys (CSUR). 2020. Vol. 54, Issue 1. Art. 16. DOI: 10.1145/3433174.3433603.

38. Amro, A., Gkioulos, V. Cyber Risk Assessment Methodology Evaluation for CPS: Maritime and Energy Cases. Journal of Marine Science and Engineering. 2023. Vol. 11, No. 4. Art. 744. DOI: 10.3390/jmse11040744.

РОЗДІЛ 1
ТЕОРЕТИЧНІ ЗАСАДИ КІБЕРФІЗИЧНИХ СИСТЕМ

Умови функціонування сучасних кіберфізичних систем дедалі більше залежать від здатності оперативно та ефективно реагувати на різнотипні загрози, що виникають на перетині фізичного та цифрового середовища. Кіберфізичні комплекси, які керують виробничими процесами, інфраструктурними об'єктами, енергетичними мережами, транспортом чи медичними приладами, характеризуються високим ступенем інтеграції, динамічністю структур, а також критичністю наслідків порушення їхньої стабільної роботи. У такому контексті управління ризиками набуває стратегічного значення, стаючи основою забезпечення стійкості, передбачуваності та життєздатності всієї інженерної екосистеми.

Системний підхід до управління ризиками в кіберфізичних системах передбачає не лише локалізацію потенційних вразливостей, але й глибоке розуміння контексту їхнього виникнення, оцінювання ймовірнісних сценаріїв розвитку подій, визначення зон найвищої критичності, а також формування адаптивних стратегій реагування. Від традиційної парадигми безпеки, яка фокусувалася переважно на запобіганні інцидентам, новітні концепції ризик-менеджменту відрізняються орієнтацією на проактивне прогнозування, багаторівневу діагностику та здатність до автономного відновлення після порушень функціональності. Інституційна практика демонструє все більшу зацікавленість щодо впровадження таких підходів, як побудова цифрових близнюків, контекстно-чутливий моніторинг, багатомодельна оцінка ризиків, інтеграція машинного навчання та формування моделей реакції на інциденти в реальному часі. Це спричинено потребою у скороченні часу між виявленням загрози і вжиттям відповідних заходів, а також у забезпеченні прозорості прийнятих рішень в умовах обмеженої інформації.

Особливого значення набуває розроблення стандартів, які регламентують методики ризик-орієнтованого управління з урахуванням специфіки кіберфізичних процесів. На відміну від класичних ІТ-систем, де об'єктом захисту є переважно дані, у КФС необхідно враховувати матеріальні процеси, динаміку зовнішніх чинників, фізичну інерцію систем і складну взаємозалежність між компонентами, що потребує поєднання інженерних, кібернетичних та поведінкових підходів. Управління ризиками розглядається як безперервний процес, що включає в себе етапи ідентифікації небезпек, оцінки ризиків, планування реагування, реалізації заходів мінімізації, контролю ефективності та коригування на основі нових даних. Застосування цього підходу у масштабі кіберфізичних систем (КФС) потребує підтримки як автоматизованого збору даних, так і аналітичних платформ, здатних обробляти великі обсяги гетерогенних даних у режимі реального часу. Актуальність розробки та впровадження методів управління ризиками в кіберфізичних системах зумовлені також

стрімким зростанням складності сучасних технологічних ландшафтів. Поєднання в одній системі таких технологій, як Інтернет-речі, хмарні сервіси, периферійні обчислення, мобільні агентні платформи та алгоритми штучного інтелекту, створює складну багатовимірну парадигму, в якій ризики можуть набувати складної, каскадної або латентної форм.

Розвиток методів аналізу ризиків у кіберфізичних системах супроводжується впровадженням інструментів, здатних не лише кількісно оцінювати ймовірність інциденту, а також моделювати його потенційні наслідки, вплив на технологічний процес, час відновлення функціональності, вартість втрат та соціальний резонанс. Застосування мультиагентних систем, байєсівських мереж, сценарного моделювання, нечіткої логіки та обґрунтованого компромісу між безпекою і продуктивністю стає основою нового покоління рішень у сфері безпечної інженерії.

Таким чином, потреба щодо формування цілісної, формалізованої та технологічно верифікованої методології управління ризиками в кіберфізичних системах визначає завдання цього розділу. Його метою є виклад і систематизація ключових підходів, що використовуються для виявлення, оцінки, моделювання, прогнозування, мінімізації та реагування на ризики в контексті функціонування складних КФС-рішень. Особливу увагу приділено огляду адаптивних, когнітивних, цифрово-бінарних та онтологічно-узгоджених моделей, здатних забезпечити не лише захист, але й стійкий розвиток високотехнологічних інфраструктур майбутнього.

1.1. Історичний розвиток кіберфізичних систем

Історія розвитку кіберфізичних систем (КФС) є результатом тривалого еволюційного процесу інтеграції фізичних та інформаційних технологій. Вона починається з простих автоматизованих механізмів і поступово трансформується в складні інтелектуальні комплекси, здатні керувати цілими виробничими процесами та великими інфраструктурними об’єктами в режимі реального часу.

Перший фундаментальний етап цього розвитку бере початок ще в середині XX століття, коли були закладені теоретичні засади кібернетики, що сформулював Норберт Вінер у 1948 році [39]. Вінер визначив кібернетику як науку про управління та зв’язок у живих організмах і машинах. Саме тоді з’явилися перші концепції використання принципів зворотного зв’язку, які уможливили автоматичне регулювання роботи різних технічних систем. Цей підхід став революційним, адже він відкрив можливість створення механізмів, що могли автономно реагувати на зміни зовнішніх умов, підтримуючи стабільність та ефективність роботи. Наступний етап, який суттєво вплинув на розвиток КФС, пов’язаний із виникненням перших SCADA-систем (Supervisory Control and Data Acquisition) у 1960–1970-х роках [40]. Ці системи призначалися для централізованого контролю та збору даних із віддалених виробничих

об’єктів. Важливою складовою цього етапу стали архітектури типу СО-SCADA, що інтегрували віддалені термінальні блоки (RTU–Remote Terminal Unit) та програмовані логічні контролери (PLC–Programmable Logic Controller) [41]. Завдяки таким системам уперше стало можливим дистанційне управління складними промисловими процесами, включаючи моніторинг стану обладнання, контроль технологічних параметрів і оперативне втручання у виробничий процес безпосередньо з центрального пункту керування. З розвитком мікропроцесорної техніки у 1980–1990-х роках можливості кіберфізичних систем значно розширилися. Компактні, потужні та доступні мікропроцесори дозволили створити більш складні та функціонально багаті системи управління, а також забезпечили масове впровадження робототехніки у виробничі процеси [42]. Цей період також відзначається появою стандартизованих мережевих інтерфейсів, таких як Modbus і PROFIBUS, що сприяли уніфікації та взаємодії обладнання різних виробників. Упровадження стандартів дало змогу спростити інтеграцію нових компонентів у вже існуючі системи управління, забезпечуючи їх гнучкість і масштабованість.

У 1990-х роках значний вплив на формування сучасних КФС мав бурхливий розвиток Інтернету. Високошвидкісний зв’язок та загальна доступність глобальних мереж уможливили створити інтегровані системи моніторингу та управління, що поєднували різноманітні фізичні об’єкти та обчислювальні ресурси в єдине ціле [43]. Саме тоді почали з’являтися перші прототипи сучасних кіберфізичних систем, які використовували мережеві технології для передачі великих обсягів інформації, необхідної для прийняття ефективних управлінських рішень. Офіційний термін «кіберфізичні системи» (Cyber-Physical Systems, CPS) був уперше запропонований близько 2006 року за ініціативи Хелен Гілл, яка працювала у Національному науковому фонді США (NSF) [44]. Саме тоді кіберфізичні системи почали виокремлювати як самостійну галузь наукових досліджень. Початок 2000-х років характеризувався активними теоретичними розробками, серед яких особливо вирізняються концепції гібридних автоматів (hybrid automata), запропоновані Томасом Хензінгером у 1990-х роках [45], а також проєкти PRET та Ptides, спрямовані на створення детерміністичних моделей для ефективного управління КФС [46]. Ці підходи дали можливість закласти науковий фундамент для подальших досліджень і розробок у галузі CPS. Другий потужний імпульс розвитку кіберфізичних систем відбувся після 2010 року, що пов’язано з упровадженням концепції Industry 4.0, або четвертої промислової революції [47]. Це був період, коли КФС почали широко застосовувати у різних сферах, включаючи виробництво, енергетику, транспорт, медицину, аграрний сектор та інфраструктуру «розумних» міст. Особливу роль у цей період відіграли технології Інтернету речей (IoT), які дозволили значно розширити можливості кіберфізичних систем щодо збору, передачі й аналізу даних у режимі реального часу [48]. З появою

хмарних обчислень і технологій Big Data в другій половині 2010-х років управління та моніторинг кіберфізичних систем вийшли на якісно новий рівень [49]. Великі масиви інформації могли оброблятися централізовано, що дозволило значно підвищити оперативність і точність діагностики, прогнозування та ухвалення управлінських рішень.

У цей час також відбулося активне впровадження штучного інтелекту (ШІ) та технологій машинного навчання у сферу кіберфізичних систем [50]. Зокрема, ШІ забезпечував можливість створення адаптивних і самонавчальних систем, які могли швидко реагувати на зміну зовнішніх умов, самостійно оптимізувати свої робочі параметри та прогнозувати можливі несправності і збої. Завдяки цьому підвищилася автономність та ефективність роботи CPS, що зробило їх ще більш привабливими для різноманітних сфер застосування. Разом з цим розвитком відбулося зростання ризиків, пов'язаних із кібербезпекою кіберфізичних систем. Одним із найвідоміших прикладів таких ризиків став вірус Stuxnet, виявлений у 2010 році [51]. Це був перший широко відомий випадок, коли шкідливе програмне забезпечення змогло фізично вплинути на роботу промислових систем через PLC-контролери. Stuxnet призвів до серйозних порушень у роботі іранських об'єктів збагачення урану, тим самим продемонструвавши вразливість критичних інфраструктур перед кіберзагрозами.

Саме у відповідь на зростання таких загроз були розроблені міжнародні стандарти безпеки для автоматизованих систем, серед яких найважливішими є стандарти серії ISA99/IEC 62443 [52]. Ці стандарти визначають комплексні заходи та вимоги щодо захисту кіберфізичних систем, які охоплюють різні рівні – від апаратних компонентів до програмних засобів управління. Дотримання цих стандартів стало критично важливим для забезпечення безперервності та безпеки роботи сучасних CPS. Сучасні кіберфізичні системи являють собою складні інтегровані комплекси, що включають в себе технології Інтернету речей (IoT), хмарні обчислення, великі дані (Big Data) та штучний інтелект (ШІ). Вони використовуються в різноманітних сферах суспільної діяльності, включаючи "розумне" виробництво, енергетику, транспорт, медицину, логістику та управління містами [53]. Завдяки високому ступеню автоматизації та можливості оперативного збору й аналізу інформації ці системи значно підвищують ефективність, надійність та економічність процесів, що ними керують. Однією з найважливіших сфер застосування кіберфізичних систем є промислове виробництво. Тут CPS забезпечують інтегроване управління технологічними процесами, автоматизацію виробничих ліній, моніторинг стану обладнання та прогнозування збоїв. Це дозволяє підприємствам мінімізувати простої, зменшувати витрати на обслуговування та забезпечувати стабільно високу якість продукції. В енергетиці CPS відіграють ключову роль у забезпеченні стабільності роботи електричних мереж, управлінні розподіленими джерелами

генерації енергії та інтеграції відновлюваних джерел енергії. Вони також сприяють оптимізації енергоспоживання, забезпечуючи баланс між виробництвом і споживанням енергії в режимі реального часу. В транспортних системах CPS використовуються для управління транспортними потоками, автоматизації руху безпілотних автомобілів та забезпечення безпеки дорожнього руху. Такі системи дозволяють знижувати аварійність, покращувати логістику і забезпечувати комфорт пасажирів. Медицина також активно використовує переваги кіберфізичних систем. Вони дозволяють здійснювати дистанційний моніторинг стану пацієнтів, автоматизувати процеси діагностики та лікування, забезпечуючи більш точну і своєчасну медичну допомогу. Перспективи подальшого розвитку кіберфізичних систем пов'язані з інтеграцією новітніх технологій, таких як квантові обчислення, блокчейн, передові алгоритми штучного інтелекту та методи глибокого навчання (deep learning) [54]. Квантові обчислення, зокрема, здатні значно підвищити швидкість і точність розрахунків, що дозволить CPS ще ефективніше управляти складними системами з великою кількістю змінних параметрів. Блокчейн-технології можуть забезпечити високий рівень захисту інформації, підвищити прозорість і довіру до обміну даними між різними компонентами CPS, що особливо важливо для критично важливих інфраструктур та виробничих процесів. Використання блокчейну дозволить створити надійні та незмінні журнали реєстрації подій і транзакцій, що зробить CPS більш безпечними і прозорими. Методи глибокого навчання та передові алгоритми штучного інтелекту відкривають нові можливості для розвитку адаптивних і автономних систем, які зможуть не лише реагувати на зміни умов роботи, але й ефективно навчатися на основі накопиченого досвіду. Це забезпечить новий рівень автономності CPS, дозволивши їм працювати майже без людського втручання, що є критично важливим у складних умовах. Сучасний розвиток кіберфізичних систем характеризується посиленою увагою до забезпечення їхньої безпеки, стійкості та надійності, оскільки зростає кількість і складність загроз, що виникають внаслідок взаємодії фізичних та інформаційних компонентів. Сучасні CPS повинні не тільки ефективно функціонувати у стабільних умовах, але й оперативно реагувати на зовнішні загрози, забезпечуючи при цьому неперервність критичних процесів [55]. Зокрема, кібербезпека стала важливою складовою розробки та експлуатації КФС. Це зумовлено тим, що системи, які керують критичною інфраструктурою, виробничими процесами, транспортом, медициною, енергетикою, повинні бути захищеними від кіберзагроз різного рівня складності. Відповідно, проводиться велика кількість досліджень у сфері захисту CPS від потенційних атак, включаючи розробку методик виявлення вторгнень, управління інцидентами та реагування на них у режимі реального часу.

Особливу увагу приділяють розробці нових підходів до управління ризиками в CPS. Це включає створення моделей оцінки ризиків, що враховують специфіку взаємодії фізичних і цифрових компонентів. Такі моделі дозволяють визначити критичні точки в системах, оцінити потенційні наслідки кібератак і сформулювати стратегії щодо мінімізації їх впливу. Зростає також роль інноваційних технологій у забезпеченні стійкості кіберфізичних систем до збоїв та відмов. Наприклад, використання цифрових близнюків (digital twins) дозволяє створювати точні віртуальні копії фізичних об'єктів і процесів, завдяки чому можна проводити комплексне тестування різних сценаріїв роботи системи, прогнозувати її поведінку в разі виникнення збоїв і аварій, а також оперативно ухвалювати рішення щодо необхідних змін у її функціонуванні. Важливим напрямом розвитку CPS є також забезпечення взаємної інтеграції різних технологічних платформ. У цьому контексті активно розробляються стандарти і протоколи, що забезпечують сумісність різних компонентів і систем незалежно від їхнього виробника чи архітектури. Це дозволяє створювати єдині екосистеми, що поєднують різноманітні фізичні та інформаційні компоненти в єдину функціональну систему. Перспективним напрямом досліджень є застосування кіберфізичних систем у контексті сталого розвитку та зниження екологічного впливу. Завдяки CPS стає можливим точний моніторинг та управління ресурсами, що дозволяє знижувати споживання енергії, оптимізувати використання водних і природних ресурсів та зменшувати викиди шкідливих речовин у довкілля. У сфері міського управління кіберфізичні системи забезпечують функціонування «розумних» міст, що включає інтегроване управління транспортними потоками, освітленням, водопостачанням та іншими міськими службами. Такі системи дозволяють ефективно розподіляти ресурси, знижувати витрати на обслуговування інфраструктури та покращувати якість життя мешканців. Інтеграція штучного інтелекту в CPS відкриває нові можливості у сфері автоматизації прийняття управлінських рішень. Сучасні системи вже здатні самостійно аналізувати великі масиви інформації, виявляти закономірності, передбачати розвиток ситуацій та пропонувати оптимальні рішення, мінімізуючи участь людини у цих процесах. Зростає також інтерес до застосування кіберфізичних систем у сільському господарстві, де вони використовуються для точного землеробства. Завдяки сенсорам, безпілотникам та автоматизованим системам управління фермери можуть точно контролювати стан посівів, ефективно використовувати добрива та засоби захисту рослин, що дозволяє підвищити врожайність і знизити витрати. Таким чином, історія розвитку кіберфізичних систем демонструє поступовий перехід від простих автоматизованих механізмів до складних інтегрованих інтелектуальних комплексів. Майбутні перспективи розвитку CPS пов'язані з подальшою інтеграцією передових технологій, розширенням сфер їхнього застосування та зростанням ролі у забезпеченні

сталого розвитку та підвищення якості життя суспільства. Це ставить перед дослідниками і розробниками нові виклики, пов’язані з необхідністю забезпечення високого рівня надійності, безпеки та ефективності таких систем у різних умовах їхнього функціонування.

1.2. Визначення та структура кіберфізичних систем

Сучасні кіберфізичні системи (КФС) демонструють високу залежність від архітектурного підходу, який обумовлює їхню здатність до масштабування, інтеграції, безпеки та адаптивності. За останні роки дослідники приділили значну увагу класифікації архітектур КФС, оскільки саме архітектура визначає базову функціональність системи та її здатність до взаємодії з іншими елементами цифрово-фізичного довкілля [55]. Питання архітектурного проєктування КФС стає ключовим у контексті промислового Інтернету речей (IIoT) та смарт-виробництва. У праці [56] автори підкреслюють важливість багаторівневої моделі проєктування, яка поєднує логічні, комунікаційні та фізичні рівні, при цьому акцентується на обов’язковому розділенні зон безпеки. Архітектура, побудована з урахуванням ієрархії управління, дозволяє ефективно реалізовувати як вертикальну, так і горизонтальну інтеграцію компонентів. Інший підхід представлено у дослідженні [57], де описано модульну архітектуру наступного покоління КФС із чіткою демаркацією функцій та сервісів. Автори наголошують на важливості відокремлення рівнів управління, збору даних, передавання повідомлень та прийняття рішень. Такий розподіл дозволяє підвищити масштабованість і полегшує обслуговування системи у складному динамічному довкіллі. Водночас архітектурні підходи мають бути формалізовані, щоб забезпечити сумісність та відтворюваність компонентів у різних цифрових ландшафтах. Як вказано в [58], архітектурне проєктування КФС повинно спиратися на загальні принципи корпоративної архітектури, що дозволяє зменшити фрагментацію системи й забезпечити узгоджене управління життєвим циклом. Для підвищення формалізованості архітектурних рішень запропоновано методи опису інтерфейсів КФС через подання дискретно-подійних специфікацій, як зазначено у [59]. Такий підхід дозволяє математично моделювати взаємодію між підсистемами, перевіряти логічну узгодженість структур і автоматизувати процес верифікації. Зростаюча складність CPS-архітектур потребує застосування засобів штучного інтелекту для їх аналізу, прогнозування та автоматизованого керування. Як зазначено у [60], поєднання моделей машинного навчання з IoT-архітектурами дає змогу не лише підвищити рівень автономності, але й оперативно реагувати на загрози, дефекти або зміни у довкіллі. У праці [61] представлено огляд загальних принципів проєктування КФС включно з питаннями визначення сервісів, ролей агентів, способів взаємодії між ними та підсистемами аналізу даних. Підхід орієнтований на формування

сервіс-орієнтованих архітектур (SOA), що сприяє гнучкості адаптації до нових вимог довкілля. У дослідженні [62] запропоновано сервісно-орієнтовану архітектуру для КФС, де сервіси виступають в ролі об’єктів першого класу. Автори стверджують, що такий підхід спрощує повторне використання компонентів, дозволяє динамічно перепідключати модулі та суттєво підвищує рівень реінжинірингу систем. У контексті забезпечення жорстких часових вимог актуальним стає використання архітектур, заснованих на Time-Sensitive Networking (TSN). Дослідження [63] демонструє застосування розподілених мережевих архітектур, що дають можливість синхронізувати критичні вузли та гарантувати передачу даних із визначеною затримкою. Забезпечення інформаційної безпеки КФС потребує формування окремих архітектур із фокусом на захист каналів, даних і контролерів. У праці [64] описано безпечну референсну архітектуру, яка охоплює контроль доступу, багаторівневе шифрування та захист від атак типу Man-in-the-Middle. Існують також загальні аналітичні моделі КФС, які описують взаємодію фізичних об’єктів, цифрових агентів і середовищ взаємодії. Такі моделі, як зазначено в [65], дозволяють формалізувати процеси адаптації системи до змін у структурі довкілля. У сфері смарт-виробництва особливе місце займають архітектури, що поєднують датчики, приводи, аналітику і хмарні сервіси. Праця [66] ілюструє практичну реалізацію такої архітектури на прикладі розумного заводу, де всі компоненти працюють у режимі реального часу під контролем загального DSS-модуля. У монографії [67] розглянуто фундаментальні засади проєктування CPS, зокрема визначення моделей узгодження поведінки агентів, часових автоматів та впливу вхідних перешкод на стан системи. У дослідженні обґрунтовано необхідність використання комбінованих моделей у поєднанні з машинним навчанням для побудови адаптивних автономних систем. Дослідження [68] повертається до концепції модульної архітектури, розглядаючи додаткові аспекти її реалізації в умовах обмежених обчислювальних ресурсів.

Автори пропонують стратегію стиснення інформаційних потоків у КФС за допомогою ієрархічного кодування, що дозволяє зменшити навантаження на комунікаційні шини. Подальший розвиток архітектур КФС пов’язаний із впровадженням концепцій гетерогенних систем, у яких взаємодіють різнотипові агенти, включаючи фізичні сенсори, цифрові контролери, хмарні сервіси та інтелектуальні модулі. Дослідження [69] підкреслює, що забезпечення узгодженої поведінки в таких ландшафтах потребує синхронізації розподілених часових моделей і формалізованої логіки обміну подіями. Окремо вивчається проблема інтероперабельності в мультисегментних КФС. У праці [70] описано протокольні механізми, які дозволяють координувати передачу даних між модулями з різними стандартами обміну (наприклад, OPC UA, MQTT, DDS). Такий підхід сприяє побудові платформонезалежних інфраструктур. Прогрес у галузі вбудованих систем дозволив розширити функціональність КФС на рівні

«edge», забезпечуючи попередню обробку даних без потреби передавання їх у хмару. Як зазначено у [71], архітектури типу fog-CPS дають можливість зменшити латентність, знизити навантаження на центральні обчислювальні вузли та підвищити стійкість до відмов мережі. Із розвитком смарт-міст та урбаністичних КФС зростає потреба в адаптивних платформах, які здатні динамічно реагувати на зміни трафіка, попиту на енергоресурси, погодні умови. У [72] запропоновано концепцію самоконфігурованих ландшафтів CPS, які базуються на інтелектуальній маршрутизації інформаційних потоків і автоматичному балансуванні навантаження. У праці [73] розглянуто застосування цифрових близнюків (Digital Twins) для побудови віртуальних копій фізичних процесів у режимі реального часу. Такий підхід дозволяє не лише здійснювати прогнозування відмов, але й активно випробовувати нові стратегії управління без ризику для фізичних об’єктів. Безпека в архітектурі КФС залишається критичним чинником. Автори [74] обґрунтовують потребу в багаторівневих схемах виявлення загроз, що поєднують сигнатурні методи, поведінковий аналіз і аномалії в потоках керування.

Універсальні безпекові фреймворки мають вбудовуватись на рівні як комунікаційного шару, так і логіки прийняття рішень. Особливу увагу приділено проблемам валідації складних архітектур. У дослідженні [75] подано формальний підхід до перевірки цілісності конфігурації CPS шляхом моделювання сценаріїв на основі правил Petri Net. Автори наголошують на необхідності автоматизованого тестування моделей ще до їх реального розгортання. Технології візуального програмування та low-code-інструменти знаходять широке застосування в контексті архітектурного розгортання КФС. У праці [76] зазначено, що візуальне налаштування логіки агентів дозволяє значно скоротити час впровадження та мінімізувати людські помилки в налаштуваннях. Нові підходи також включають використання гібридних логік у розподілених КФС. Як зазначено в [77], поєднання логіки скінченних автоматів із часовими обмеженнями забезпечує більш точне відтворення поведінки системи в умовах режиму реального часу, особливо в критичних сценаріях. У сфері адаптивного управління розглядається застосування нейро-фаззі-систем, які дозволяють КФС самостійно налаштовувати алгоритми реагування на підставі попереднього досвіду. У праці [78] описано приклади реалізації таких підходів у смарт-енергетичних мережах. Дослідники також акцентують на ролі семантичної інтероперабельності. Згідно з [79], для досягнення повноцінної взаємодії між модулями КФС різного походження необхідно впроваджувати онтології, які описують типи даних, подій, процесів і контекстів уніфікованим способом. У межах концепції сталого розвитку актуальним стає впровадження «зелених» архітектур КФС. У праці [80] представлено метод енергетичного профілювання компонентів CPS для динамічного вимкнення неактивних модулів у фазі очікування без шкоди для загальної

функціональності. З точки зору управління життєвим циклом, сучасні КФС інтегрують фреймворки DevOps та CI/CD для автоматичного оновлення функціональних блоків. Дослідження [81] демонструє, що впровадження таких практик дозволяє значно зменшити час між ітераціями та знизити кількість збоїв під час розгортання оновлень. Нарешті, у [82] аналізується архітектура CPS-платформ на основі мікросервісів, які дозволяють створювати гнучкі, масштабовані системи, що легко адаптуються до зміни завдань або середовищ функціонування. Вдосконалення моделювання взаємодії між фізичними об'єктами, цифровими агентами та середовищами обміну даними є критичним аспектом для підвищення адаптивності сучасних КФС. Як зазначено в [65], загальні аналітичні моделі КФС дають можливість формалізувати адаптивні процеси систем у змінному інформаційному середовищі, створюючи основу для автоматизованої конфігурації та керування. У сфері смарт-виробництва домінує архітектурний підхід, що поєднує сенсори, виконавчі пристрої, хмарну аналітику та DSS-модулі в єдину екосистему, яка працює в режимі реального часу. У дослідженні [66] ілюструється практична реалізація подібної архітектури для індустріального підприємства, де кожен компонент інтегровано в контекст загального цифрового планування та оптимізації. Монографія [67] систематизує фундаментальні засади проєктування адаптивних CPS, зокрема моделі узгодження поведінки агентів на основі часових автоматів. Тут розглянуто також вплив стохастичних збурень та перешкод на стабільність роботи систем, що є особливо важливим у проєктуванні автономних КФС у непередбачуваних обставинах. Питання ефективної реалізації модульної архітектури в умовах обмежених ресурсів досліджується у [68]. Автори пропонують використання ієрархічного кодування для зменшення обсягу інформаційного потоку, що дозволяє знизити навантаження на комунікаційні шини, зберігаючи при цьому точність та актуальність даних.

Подальша еволюція КФС пов'язана з інтеграцією гетерогенних компонентів – сенсорів, контролерів, хмарних сервісів та ШІ-модулів – у єдину когерентну інфраструктуру. Згідно з [69], забезпечення узгодженої поведінки в таких системах потребує формального опису подієвого обміну та синхронізації часових моделей, що є передумовою для коректної колективної дії агентів у режимі реального часу. Інтероперабельність між модулями з різними протоколами обміну (наприклад, OPC UA, MQTT, DDS) є предметом дослідження [70]. Тут описано протокольні механізми трансляції повідомлень і динамічного узгодження форматів, що дозволяє реалізувати платформонезалежну інфраструктуру обміну даними між гетерогенними системами. На тлі зростання вимог до швидкодії систем все більшого поширення набувають fog-архітектури, які забезпечують попередню обробку інформації на рівні edge. Як зазначено у [71], це уможливлює значно знизити латентність,

розвантажити центральні вузли обчислень і підвищити стійкість до збоїв мережевого зв’язку. У праці [72] запропоновано концепцію самоконфігурованих середовищ CPS, де адаптація до умов середовища здійснюється шляхом динамічного маршрутизування інформаційних потоків та автоматичного балансування обчислювального навантаження між вузлами. Технологія цифрових близнюків стає одним із ключових інструментів прогнозування поведінки складних систем. У [73] описано створення віртуальних копій фізичних процесів, що дає змогу не лише передбачати відмови, але й експериментально перевіряти нові алгоритми керування без втручання в реальну інфраструктуру. Безпека у КФС залишається критичною складовою архітектурних рішень. У дослідженні [74] окреслено багаторівневі підходи до виявлення аномалій, що включають сигнатурний аналіз, моделювання поведінки та виявлення нетипових патернів у потоках керування. Ці методи забезпечують своєчасну ідентифікацію загроз без шкоди для продуктивності системи. Формальна верифікація складних CPS-архітектур потребує застосування математичних моделей. У [75] автори демонструють використання Petri Net для симуляції сценаріїв конфігурації, що дозволяє виявляти конфлікти, помилки та невідповідності ще до етапу реального впровадження.

1.3. Функціональні компоненти

Кіберфізичні системи (КФС) є багаторівневими технолого-інформаційними комплексами, що поєднують у собі фізичні пристрої, обчислювальні модулі, програмне забезпечення і комунікаційні мережі, здатні до автономного прийняття рішень та взаємодії з довкіллям. Основу функціональної структури КФС становлять чотири ключові компоненти: сенсорний рівень, рівень управління, мережевий рівень та виконавчий рівень, які взаємодіють у межах зворотного зв’язку в режимі реального часу [83].

Сенсорний рівень

Сенсорна підсистема КФС виконує функцію безперервного збору даних з фізичного довкілля, включаючи механічні, термічні, електромагнітні, акустичні та інші параметри. Висока щільність розміщення сенсорів дозволяє досягти максимальної точності в моніторингу об’єктів керування. Важливими характеристиками сенсорних модулів є: швидкість вимірювання, енергоефективність, точність калібрування та здатність до самодіагностики [84]. У контексті індустрії 4.0 особливої актуальності набувають інтелектуальні сенсори, здатні не лише передавати, але й попередньо обробляти інформацію безпосередньо на місці її виникнення (edge-computing).

Рівень управління

Керуюча підсистема КФС виконує функції обробки отриманих даних, оцінки поточного стану системи, прогнозування її поведінки та генерації команд для виконавчих механізмів. У ній можуть бути реалізовані як класичні ПІД-регулятори, так і адаптивні чи когнітивні алгоритми управління на основі штучного інтелекту (ШІ) [85]. Сучасні керуючі модулі КФС найчастіше реалізуються на базі вбудованих мікропроцесорних платформ або ПЛІС (FPGA), які забезпечують одночасну обробку великих потоків інформації із низькою затримкою. Інтеграція моделей глибокого навчання (deep learning) у керуючі ланки дозволяє формувати самонавчальні контролери, здатні до адаптації до зміни зовнішніх умов та внутрішніх параметрів системи. У випадку критичних інфраструктур, таких як енергетика або транспорт, впровадження таких інтелектуальних систем підвищує рівень автономності та знижує залежність від операторів [86].

Мережевий рівень

Ключовим елементом функціональності КФС є надійна комунікація між компонентами системи. Мережевий рівень забезпечує обмін даними між сенсорами, контролерами, виконавчими модулями та зовнішніми системами керування. Типова реалізація включає як дротові протоколи (Ethernet, CAN, Modbus, PROFIBUS), так і бездротові стандарти (Wi-Fi, ZigBee, LoRa, 5G), які оптимізуються під реальні умови роботи КФС [87].

Зростання вимог до детермінованості передачі інформації у системах режиму реального часу зумовлює впровадження мереж із жорсткими часовими обмеженнями (Time-Sensitive Networking, TSN), які дозволяють гарантувати затримку і пропускну здатність при одночасній підтримці синхронізації між модулями [88].

Виконавчий рівень

Виконавча підсистема реалізує фізичний вплив на об’єкт управління відповідно до отриманих команд. До її складу входять електромеханічні приводи, гідравлічні системи, маніпулятори, виконавчі роботи, інтелектуальні пристрої типу actuator-as-a-service [89]. Ефективність виконавчих механізмів прямо впливає на здатність системи до швидкого реагування, динамічної стабілізації та оптимізації процесів. У сучасних реалізаціях акцент робиться на енергоефективності та зниженні інерційності, що особливо актуально в умовах гнучкого виробництва та мобільної робототехніки. У випадках використання автономних агентів (наприклад, дронів), виконавча частина включає модулі планування маршруту, уникнення перешкод та підтримки стійкості польоту [90].

Рівень інтеграції та цифрового представлення

Окрему роль у функціональній структурі КФС відіграє рівень цифрових моделей, який забезпечує абстрактне подання фізичних об'єктів у вигляді цифрових близнюків (Digital Twins). Такі моделі дозволяють симулювати сценарії, прогнозувати відмови, тестувати зміни у віртуальному середовищі перед їх реальним впровадженням [91]. Цифрові близнюки використовуються також для оптимізації енергоспоживання, прогнозування навантажень, управління ризиками і формування стратегій технічного обслуговування. У цьому контексті важливим є використання хмарних платформ (AWS IoT, Azure Digital Twins), які забезпечують масштабованість і інтеграцію з іншими елементами КФС [92].

Модуль інтерфейсів і безпеки

У сучасних КФС обов'язковою є реалізація функціонального модуля, що відповідає за інтерфейси взаємодії з людиною (HMI), зовнішніми системами (API), а також захист комунікацій (TLS, VPN, Zero Trust Architectures). Надійна і безпечна взаємодія є критичною умовою під час роботи в інфраструктурах критичного значення або в умовах активного кібервпливу [93]. У цьому модулі також реалізуються системи контролю доступу, шифрування конфіденційної інформації, детекції аномалій та реагування на інциденти, що дає змогу зменшити ризики порушення цілісності та доступності системи [94].

Інтелектуальні керувальні компоненти

Інтеграція технологій штучного інтелекту в структуру КФС дозволила суттєво розширити функціональні можливості керувальних підсистем. Сучасні системи управління часто будуються на принципах когнітивного контролю, що включає самонавчання, адаптацію до середовищ, прогнозування відхилень і оптимізацію дій на основі накопиченого досвіду [85]. У цьому контексті все частіше застосовуються нейромережеві моделі, здатні до реального навчання під час роботи, включаючи глибокі згорткові мережі (CNN), рекурентні (RNN, LSTM), автокодери і нейрофаззі-системи. У промислових КФС інтелектуальні контролери дозволяють передбачати збої, автоматично оновлювати логіку реагування на події, а також здійснювати діагностику причин порушення функціональності без втручання оператора. Наприклад, упровадження контролерів на основі нейронних Q-мереж (DQN) у смарт-енергомережах дає змогу динамічно балансувати навантаження і уникати перевантажень у розподілених топологіях [86]. Особливу роль інтелектуальні керувальні компоненти відіграють у системах мобільної робототехніки, де необхідне динамічне

переобчислення маршрутів, уникнення перешкод, адаптація до змін рельєфу та погодних умов.

Периферійна та хмарна обробка

Класичні КФС тривалий час використовували централізовані обчислювальні моделі (cloud-based). Проте з розвитком fog computing і edge AI обробка даних усе частіше переноситься ближче до джерела їх виникнення. Це дозволяє мінімізувати затримки, знизити навантаження на мережеві канали та підвищити відмовостійкість систем [87]. Периферійні вузли (edge nodes) тепер обладнані не тільки сенсорами і виконавчими механізмами, але й інтелектуальними процесорами, які дозволяють виконувати алгоритми класифікації, прогнозування або виявлення аномалій у режимі реального часу.

Наприклад, використання графових процесорів (GPU) на периферії забезпечує виконання моделей CNN у задачах візуального моніторингу індустріальних процесів. Fog-архітектури, зі свого боку, реалізують проміжний рівень між периферією та хмарою. Вони здатні агрегувати інформацію з кількох edge-вузлів, формувати аналітичні зведення та локально виконувати складні розрахунки з високою частотою оновлення [88].

Цифрові близнюки в управлінні життєвим циклом

Цифрові близнюки (Digital Twins, DT) є віртуальними аналогами фізичних об’єктів, процесів або систем, які оновлюються в режимі реального часу. У складі КФС вони виконують роль інтелектуального інтегратора, забезпечуючи візуалізацію стану, симуляцію сценаріїв, аналіз відмов і автоматичне планування технічного обслуговування [89]. Використання DT особливо ефективне в галузях, де важливе тестування рішень до їх впровадження, наприклад в енергетиці, авіації, транспортній логістиці та фармацевтиці. Через віртуалізацію відбувається об’єднання даних із сенсорів, аналітичних моделей, машинного навчання та операторського інтерфейсу в єдину когерентну картину функціонування системи. Платформи, що підтримують DT (Siemens MindSphere, Azure Digital Twins), дозволяють управляти життєвим циклом КФС – від розгортання, до модернізації та виведення з експлуатації – з використанням автоматизованих CI/CD-процесів, характерних для DevOps-підходів [90].

Хмарна інфраструктура та масштабовані обчислення

У сучасних КФС хмара виконує не лише функцію централізованого сховища, але й стає основою для масштабованих аналітичних обчислень. Завдяки хмарним сервісам з’являється можливість реалізації таких

функцій, як глобальна координація дій між віддаленими об’єктами, централізований аналіз Big Data, глибоке навчання на кластерах GPU та обслуговування DT-екземплярів [91]. Сучасні платформи IoT (наприклад, AWS IoT Greengrass, Google Cloud IoT Core) надають повну інфраструктуру для розгортання КФС за моделлю «інфраструктура як код» (IaC), дозволяючи автоматизувати налаштування, оновлення та моніторинг мільйонів вузлів.

Переваги хмарної інтеграції включають:

масштабованість – можливість динамічного додавання або видалення обчислювальних ресурсів;

глобальну доступність – підтримка мультирегіональних систем;

уніфікацію безпеки – через централізовану політику контролю доступу, журналювання та шифрування.

Інтероперабельність та стандартизація компонентів

Ефективність функціонування КФС залежить не лише від рівня технічної реалізації окремих модулів, але й від інтероперабельності – здатності різнорідних компонентів взаємодіяти у єдиному середовищі. Це особливо актуально в умовах гетерогенних інфраструктур, де задіяні пристрої від різних виробників, з різними протоколами та форматами даних. Основу забезпечення інтероперабельності становить впровадження єдиних моделей даних, онтологій (semantic models) та стандартизованих протоколів обміну, таких як OPC UA, DDS, MQTT або CoAP.

Використання таких протоколів дозволяє зменшити витрати на інтеграцію та підтримку, а також підвищити гнучкість адаптації до змін [83]. На рівні системного моделювання активно впроваджуються фреймворки на кшталт IEC 61499, які дають можливість описувати логіку функціонування компонентів у формі функціональних блоків з чіткою структурою вхідних та вихідних сигналів. Це сприяє уніфікації реалізації функціональних підсистем КФС та підвищує відтворюваність архітектур [84].

Адаптивні механізми прийняття рішень

У складних динамічних системах класичні методи управління КФС часто не здатні забезпечити необхідну адаптивність. Саме тому все більшої актуальності набувають контекстно-залежні адаптивні механізми, здатні перебудовувати логіку функціонування системи на основі аналізу змін довкілля, внутрішнього стану та історичних даних. Такі механізми реалізуються у вигляді багаторівневих агентів – рефлексивного рівня (реактивна поведінка), тактичного рівня (локальна адаптація), стратегічного рівня (довготривале планування). Усі ці рівні використовують машинне навчання, байєсівські мережі, методи зміцнювального навчання та евристичну оптимізацію [85]. У багатьох

КФС адаптивність досягається шляхом реінжинірингу функціональних маршрутів на основі аналізу індикаторів ризику, доступності ресурсів, кіберзагроз або відмов обладнання. Наприклад, в енергетичних системах споживання електроенергія перерозподіляється між джерелами відповідно до виявлених відхилень у мережевих параметрах [86].

Людино-машинні інтерфейси та візуалізація

Функціональним компонентом КФС є модуль інтерактивної взаємодії з оператором. Він реалізується зо допомогою людино-машинних інтерфейсів (HMI), що забезпечують візуалізацію поточного стану системи, сигналізацію критичних подій та інтерактивне управління параметрами. Інтерфейси нового покоління орієнтуються на використання розширеної реальності (AR) та віртуальної реальності (VR) для оперативного огляду стану КФС, виконання ремонтно-технічного обслуговування або симуляції аварійних сценаріїв [87]. Інтерфейси також адаптуються до поведінки користувача через системи розпізнавання голосу, жестикуляції та біометричної автентифікації, що особливо актуально в умовах високої відповідальності або обмеженого часу на прийняття рішень [88].

Кібербезпека функціональних підсистем

Оскільки всі компоненти КФС взаємопов’язані через відкриті або напіввідкриті мережі, безпека є невід’ємною частиною їхньої функціональності. У структурі КФС повинні бути реалізовані інтегровані засоби захисту, що охоплюють не лише прикладний рівень, але й апаратний, мережевий та логічний шари [89].

Захист функціональних підсистем реалізується за допомогою таких елементів:

ідентифікацію та автентифікацію пристроїв (на основі TPM, PKI або Zero Trust моделей);

моніторинг аномалій у трафіка або поведінці модулів на основі моделей поведінкової аналітики;

ізоляцію підсистем у сегментовані зони з обмеженим доступом;

шифрування міжкомпонентної взаємодії із застосуванням TLS 1.3, DTLS або VPN-рішень.

Нова тенденція – використання безпеково-орієнтованих мікросервісів, кожен із яких має власні політики доступу та логіку обробки ризиків [90].

Самодіагностика та обслуговування за станом

Для зниження вартості експлуатації та збільшення надійності систем у КФС широко застосовуються модулі самодіагностики, які забезпечують

виявлення порушень у функціональності елементів в режимі реального часу. Такі модулі часто побудовані на принципах убудованого моніторингу та порівняльного аналізу з цифровими еталонами, що зберігаються в системі [91]. Системи обслуговування за станом (Condition-Based Maintenance CBM) дозволяють виконувати ремонт лише в разі реального зниження ефективності чи виявлення аномалій. Для цього застосовуються:

вібраційна діагностика;

аналіз спектра живлення;

тепловізійний контроль;

моделі прогнозування відмов на основі нейромережевих предикторів.

Сучасні підходи доповнюються цифровими журналами подій (ledger-based maintenance), які фіксують усю історію втручань у систему, змін конфігурації та впровадження оновлень. Це уможливлює автоматизувати аудит та формувати верифіковану документацію.

Автономність та самоконфігурація

Останнім із важливих функціональних елементів КФС є здатність до самоконфігурації та автономного відновлення після збоїв. Цей принцип реалізується через убудовані модулі самовідновлення, які використовують резервні маршрути, адаптивне перепідключення вузлів та динамічне відновлення логіки обробки даних [92]. Прикладом є сценарії роботи роботизованих платформ, які при втраті зв’язку із центральним вузлом здатні перейти в режим локального керування з обмеженим функціоналом. Інший приклад – автономна перебудова системи маршрутизації в умовах перевантаження мережі або виявлення атаки типу DoS [93]. Для підтримки таких можливостей використовуються агентно-орієнтовані фреймворки та інфраструктура самоорганізованих мереж (ad-hoc CPS). Крім того, активне застосування віднаходять механізми довірчого обміну даними (Trusted Execution Environments) на рівні чипів і мікроконтролерів [94].

1.4. Загрози та вразливості

Загальна характеристика загроз

Кіберфізичні системи (КФС) функціонують на стику фізичного і цифрового середовища, тому піддаються одночасно як класичним інформаційним загрозам, так і фізичним впливам. Таке поєднання створює розширену площину атак, де вразливості можуть виникати не лише у програмному коді, а також у мережевій інфраструктурі, сенсорних пристроях, логіці керування або людському факторі [75].

За класифікацією [76], основні типи загроз для КФС включають:

порушення цілісності даних, що впливають на процес прийняття рішень;

несанкціоноване управління фізичними об'єктами;

відмова або зупинка критичних функцій;

викрадення конфіденційної інформації, що стосується технологічного процесу або стану системи;

підміна сигналів сенсорів та виконавчих пристроїв;

віддалене встановлення шкідливого програмного забезпечення (rootkit, logic bomb, malware).

Загрози можуть виникати внаслідок вразливостей на одному або кількох рівнях: програмному, мережевому, фізичному чи соціальному. Згідно з [77], більшість успішних атак на КФС мають багатоступеневий характер, коли зловмисник послідовно експлуатує декілька компонентів з метою досягнення контрольованої модифікації фізичного процесу.

Атаки на рівні сенсорної інфраструктури

Сенсорні компоненти, які слугують точкою входу інформації в КФС, часто залишаються найменш захищеними, особливо в умовах відкритого або агресивного довкілля. До найпоширеніших атак на сенсори належать:

підміна вхідних сигналів (spoofing), наприклад, GPS-спуфінг в автономних транспортних засобах;

ін'єкції помилкових даних (false data injection), що порушують цілісність обчислень;

електромагнітні перешкоди або спрямоване фізичне руйнування сенсорних елементів;

зворотні інженерні впливи, які дозволяють змінити калібрувальні параметри сенсора.

Як показано у [78], навіть мікросекундні затримки або спотворення сигналів із сенсорів можуть критично вплинути на керування, викликати порушення безпеки або помилки в діагностиці. При цьому виявлення подібних атак є ускладненим, оскільки більшість систем не мають убудованих механізмів перевірки достовірності сенсорних даних.

Вразливості мережевої комунікації

Мережевий рівень КФС виступає основною мішенню для атак, пов'язаних із перехопленням, підміною або блокуванням даних. Серед основних векторів атак:

Man-in-the-Middle (MitM) – перехоплення або модифікація трафіка між модулями КФС;

Denial of Service (DoS / DDoS) – виведення з ладу комунікацій шляхом перевантаження;

Replay-атаки – повторна передача автентичних, але застарілих, повідомлень;

Packet sniffing – несанкціоноване прослуховування каналів передачі.

У системах з відкритими або застарілими протоколами, такими як Modbus або DNP3, зазвичай відсутнє шифрування, що робить комунікацію прозорою для зловмисників [79]. Згідно з аналізом [80], майже 60% SCADA-систем досі не мають упроваджених TLS/SSL, що створює широке поле для атак на мережевому рівні.

Атаки на виконавчі модулі та фізичні пристрої

Виконавчі компоненти, які фізично ваємодіють із ландшафтом, є критичними точками впливу на функціонування КФС. Атаки на ці модулі можуть призвести до матеріальних втрат, аварій чи загроз життю. Найбільш небезпечні сценарії включають:

перехоплення контролю над виконавчими пристроями (наприклад, через злам PLC-контролера);

ін'єкцію команд, які порушують нормальну логіку функціонування системи;

створення небезпечних режимів експлуатації (перевантаження, перегрів, блокування рухомих частин);

логічне руйнування обладнання через модифікацію сигнального циклу.

Класичним прикладом є атака Stuxnet, яка використовувала rootkit-механізми для перепрограмування ПЛК-контролерів на іранському ядерному об'єкті [81]. Атака залишалась невиявленою упродовж тривалого часу, оскільки паралельно транслювала коректні сигнали на HMI, тоді як реальні дії відрізнялись.

Програмні та логічні вразливості

У ПЗ КФС часто використовуються вбудовані ОС (RTOS), драйвери пристроїв, контролери протоколів та користувацькі скрипти, які піддаються класичним атакам типу buffer overflow, code injection, race condition. Через обмеженість ресурсів у таких системах рідко використовуються повноцінні антивірусні або sandbox-захисти. Багато інцидентів спричиняються використанням сторонніх бібліотек без перевірки цілісності або оновлень, що відкриває шлях до supply chain attacks [82]. Також загрозою є hardcoded credentials, тобто жорстко зашиті паролі у прошивках, які можуть бути витягнуті з фізичних носіїв.

Людський фактор і соціальні загрози

Одним із найменш контрольованих, але водночас найнебезпечніших чинників загроз для КФС, залишається людський фактор. Як зазначено в [83], понад 30% успішних кібератак на промислові системи мали в своєму ланцюзі хоча б один етап соціальної інженерії – від фішингових листів до несвідомого встановлення шкідливого ПЗ через USB-носії.

Основні соціальні загрози:

фішинг та spear-phishing, орієнтовані на технічний персонал КФС;

використання соціально скомпрометованих облікових даних для входу в SCADA;

помилки операторів у стресових умовах, які призводять до порушення штатних алгоритмів керування;

інсайдерські загрози, зокрема саботаж або навмисна зміна конфігурації систем.

Вразливість людини полягає також у низькому рівні підготовки щодо кібергігієни, наприклад, використання слабких паролів, відсутність двофакторної автентифікації або ігнорування системних попереджень [84].

У багатьох випадках системи не мають належного аудиту дій користувача, що унеможливлює оперативне реагування на внутрішні інциденти.

Техногенні ризики та фізичне втручання

КФС у багатьох галузях розгортаються у важкодоступних, нестабільних або агресивних умовах (нафтовидобування, енергетика, транспорт, охорона здоров'я), що створює ризики прямого фізичного втручання. Наприклад:

підміна або пошкодження кабелів зв'язку;

знищення вузлів edge-комунікації;

порушення живлення сенсорної або виконавчої інфраструктури;

reverse engineering та витяг прошивок з флеш-пам'яті пристроїв.

У дослідженні [85] показано, що відсутність фізичного захисту у промислових розподілених системах призводить до легкого доступу до контролерів, що часто мають незахищений web-інтерфейс або відкриті порти керування. Техногенні ризики також включають вплив екстремальних температур, вібрацій, вологості або пилу, які можуть викликати спотворення сигналів, передчасний вихід із ладу сенсорів або втрату з'єднання. При цьому відсутність механізмів самодіагностики ускладнює локалізацію причин інциденту.

Типові вразливості в галузевому контексті

У різних сферах застосування КФС спостерігаються повторювані патерни вразливостей, обумовлені специфікою системи, регламентами та архітектурними обмеженнями:

Енергетика: вразливість SCADA-комунікацій, недостатній моніторинг вторгнень, використання застарілих протоколів (таких, як IEC 60870-5-104 без TLS), централізоване керування без резервних каналів [86].

Транспорт: відкритість CAN-шини в автомобілях, вразливості у GPS, LIDAR і V2X-комунікаціях, недоступність оновлень для застарілого ПЗ [87].

Охорона здоров'я: використання відкритих мереж Wi-Fi для зв'язку з медичними сенсорами, відсутність шифрування в каналах передачі з ЕКГ- або інсулінових помп, загроза зміни даних моніторингу пацієнта [88].

Виробництво: відсутність логування дій операторів, слабке розділення прав доступу, незахищені REST API у IIoT-пристроях [89].

Ці приклади свідчать про необхідність галузеспецифічної адаптації заходів безпеки, а не застосування загальних фреймворків без урахування контексту.

Профілактика: виявлення та запобігання атакам

Одним із ключових напрямів зниження ризиків у КФС є розгортання механізмів виявлення атак (Intrusion Detection Systems, IDS) із підтримкою специфіки КФС – низької затримки, фрагментованих протоколів, обмежених ресурсів тощо. Існують три основні підходи до побудови таких систем:

сигнатурний – виявлення відомих атак на основі шаблонів (не ефективний проти zero-day);

поведінковий – формування профілів нормальної роботи з виявленням аномалій [90];

гібридний – поєднання попередніх із використанням ML-алгоритмів класифікації (наприклад, SVM, decision tree, autoencoder).

Згідно з [91], сучасні IDS для КФС базуються на edge-модулях, що дозволяє виконувати виявлення загроз на рівні периферії без відправки трафіка до хмари. При цьому основна проблема таких систем – велика кількість хибнопозитивних спрацьовувань, що знижує довіру до них у комплексах оперативного виробництва.

Для запобігання атакам також використовуються:

segmentation firewall – для ізоляції сегментів КФС;

додаткові політики доступу (role-based, attribute-based);

sandboxing виконавчих процесів;

впровадження "zero trust" моделі безпеки [92].

Виведення системи з ладу через ланцюг постачання

Сучасні КФС є результатом інтеграції багатьох апаратних та програмних компонентів, що робить їх вразливими до supply chain attacks атак, здійснених через скомпрометовані оновлення, бібліотеки або вбудовані рутинги. Прикладами таких атак є:

модифікація прошивки перед встановленням;

вбудовані бекдори у драйверах пристроїв;

шкідливі зміни у відкритих бібліотеках або хмарних сервісах API [93].

Вразливість посилюється тим, що багато виробників IIoT-обладнання не мають належної процедури перевірки цілісності програмного коду,

оновлення надсилаються без підпису або завантажуються через відкриті HTTP-з'єднання.

Для зменшення ризику вразливості ланцюга постачання [94] рекомендує впровадження політик:

SBOM (Software Bill of Materials) – фіксація всіх залежностей ПЗ;

криптографічне підписання кожного релізу;

використання незалежних сканерів уразливостей (наприклад, для контейнерів Docker, бібліотек Python/Node.js);

відокремлення систем розробки, тестування та продакшну.

Комплексний аналіз загроз, вразливостей та механізмів реагування у кіберфізичних системах

В умовах швидкої еволюції кіберфізичних систем одним із найважливіших напрямів досліджень і практичного забезпечення надійності залишається аналіз загроз та вразливостей, які притаманні як апаратній, так і програмній інфраструктурі. КФС, що охоплюють елементи фізичного світу та цифрових технологій, водночас стають об'єктами дії багатовекторних ризиків, які посилюються через складність їхньої побудови, гетерогенність компонентів і високий ступінь інтеграції з відкритими мережами. На відміну від класичних IT-систем, де атака часто спрямована на інформацію, у випадку КФС метою зловмисника можуть бути не лише дані, але й фізична компонента, об'єкти критичної інфраструктури, або ж життя та здоров'я людей, що значно ускладнює як захист, так і аналіз ризиків [75], [77].

Вразливості КФС мають системний характер і виявляються на всіх рівнях: від периферійних сенсорів і пристроїв введення до хмарних сервісів обробки та стратегічного управління. Згідно з сучасною практикою, більшість успішних атак на кіберфізичні системи є складносконфігурованими та включають послідовне використання кількох каналів впливу: соціального (через людський фактор), мережевого (через вразливості комунікації), прикладного (шляхом маніпуляцій з логікою ПЗ) та фізичного (наприклад, через безпосередній доступ до вузлів керування) [76], [78]. На особливу увагу заслуговують атаки, які націлені на приховане маніпулювання виконавчими пристроями. Це явище набуло критичного значення після кейсу Stuxnet, коли через програмне втручання в логіку ПЛК-контролера вдалося вплинути на швидкість обертання центрифуг при повному збереженні ілюзії штатної роботи системи [81]. Аналогічні приклади спостерігалися і в інших секторах: в автомобілебудуванні (дистанційне втручання в CAN-шину), енергетиці (переналаштування логіки перемикання навантаження), медицині (спотворення дозування на інсулінових помпах) та логістиці (маніпуляції даними GPS-навігації для безпілотних засобів доставки) [83], [84]. Невід'ємною складовою вразливості КФС є комунікаційна інфраструктура.

Велика кількість об’єктів використовує відкриті або застарілі протоколи без криптографічного захисту, зокрема Modbus, DNP3, OPC-UA без TLS. Внаслідок цього мережі стають вразливими до таких атак, як перехоплення та підміна пакетів, повторна передача старих даних, сканування топології, а також ін’єкція шкідливих команд [80], [86]. Зафіксовано численні випадки, коли втручання в мережевий трафік дозволяло повністю взяти під контроль виробничий процес, змінити параметри подачі енергії або викликати аварійне зупинення обладнання [87]. Ще одним проблемним аспектом залишається питання оновлення програмного забезпечення в пристроях, що становлять КФС. Значна частина вразливостей виникає через відсутність захищених каналів оновлення, застосування сторонніх компонентів із відкритим кодом без перевірки цілісності, або використання бібліотек зі встановленими бекдорами. Supply chain-атаки стають особливо небезпечними в контексті автоматизованих систем, де один скомпрометований компонент потенційно здатен уразити всю інфраструктуру [90], [93]. При цьому часто спостерігається відсутність політик і процедур криптографічного підпису, багаторівневої перевірки залежностей або реєстрів використаного ПЗ, що значно ускладнює верифікацію ланцюгів постачання [94]. На додаток до технічних уразливостей критичну роль відіграє людський фактор.

Більшість операторів систем не проходять спеціальної підготовки з кібергігієни, а також схильні використовувати прості або повторювані паролі, зберігати доступи у відкритому вигляді або несвідомо запускати шкідливі скрипти [82], [85]. Навіть за наявності засобів аутентифікації та журналювання дій операторів часто відсутній моніторинг або аналітика, що перешкоджає виявленню нетипових шаблонів поведінки. З іншого боку, зловмисники активно використовують соціальну інженерію – зокрема, spear-phishing або підроблені оновлення, як стартову точку атаки на КФС [83]. У контексті реагування на загрози ключове значення має використання систем виявлення атак, здатних працювати в умовах обмеженої обчислювальної потужності та низької латентності. Сучасні системи IDS/IPS, адаптовані до вимог КФС, базуються на гібридному підході: сигнатурні методи поєднуються з алгоритмами поведінкової аналітики та глибинного навчання [88], [89]. Такий підхід дозволяє не лише виявляти відомі атаки, але й формувати профілі нормальної поведінки системи, щоб оперативно фіксувати аномальні зміни. Проте застосування таких рішень у виробничому техногенному ландшафті стикається з проблемами, зокрема з високим рівнем хибнопозитивних спрацьовувань, складністю налаштування параметрів у реальному часі та обмеженнями щодо обсягів трафіка, який можна аналізувати на периферійному рівні [91]. Важливо підкреслити, що моделі ризику для КФС не можуть базуватися виключно на класичних підходах до оцінки інформаційної безпеки. Оскільки йдеться про системи, які фізично взаємодіють із об’єктами навколишнього середовища, необхідно включати

до аналізу параметри, пов’язані з динамікою фізичних процесів, імовірністю виникнення аварій, а також соціальними наслідками вторгнень [79], [84]. Наприклад, у промисловому середовищі порушення ритму роботи станка через мережеву аномалію може призвести до пошкодження сировини, зниження якості продукції або навіть техногенних катастроф. Саме тому провідні дослідники пропонують використовувати розширені методи моделювання, що поєднують оцінку ймовірностей атак,

Показовими в цьому плані є підходи до оцінки ризиків у системах типу smart grid, де враховується не лише ймовірність атаки на конкретний вузол, а також швидкість її поширення, масштаб впливу на баланс мережі, можливість виникнення каскадних відмов та коефіцієнт енергонезалежності регіону [86], [87]. Аналогічно, в транспортних системах важливим є не лише запобігання перехопленню керування безпілотними платформами, але й прогнозування сценаріїв, коли компрометація одного елемента викликає доміно-ефект, що впливає на загальну логістику або безпеку пасажирів [88]. Отже, аналіз загроз і вразливостей КФС потребує міждисциплінарного підходу, що охоплює кібербезпеку, інженерію систем керування, фізичне моделювання та поведінкову аналітику. Структурована класифікація загроз має враховувати рівень уразливості, характер впливу (цифровий або фізичний), джерело атаки (внутрішнє, зовнішнє, постачальницьке), а також тип використовуваної інфраструктури. Така типологізація дає змогу формувати ефективні стратегії виявлення, превенції та реагування, адаптовані до контексту конкретної галузі [93], [94].

1.5. Методи управління ризиками

У сучасних кіберфізичних системах (КФС) управління ризиками є критичним компонентом, що забезпечує надійність, безперервність і безпеку функціонування інтегрованих фізико-цифрових процесів. Особливістю ризик-менеджменту в КФС є його міждисциплінарний характер: процеси ідентифікації, оцінки, мінімізації та моніторингу ризиків мають враховувати не лише IT-рівень, але й специфіку фізичних об’єктів, мережевої інфраструктури, часових обмежень та людського чинника [39], [40].

Концептуальні основи управління ризиками в КФС

Управління ризиками в КФС ґрунтується на класичних принципах теорії ймовірностей, а також методах безперервного оцінювання стану системи в умовах невизначеності. Під ризиком у даному контексті розуміють функцію від імовірності настання небажаної події та тяжкості її наслідків [41]. Ця модель є особливо актуальною для критичних секторів – енергетики, транспорту, медицини, промисловості – де помилка в керуванні може призвести не лише до економічних збитків, але й до людських втрат [42].

Ризик-орієнтоване управління передбачає побудову матриць ризику, у яких оцінюється взаємозв'язок між вразливістю системи, потенційними загрозами та рівнем впливу на компоненти CPS [43]. Така модель дозволяє не тільки пріоритезувати загрози, а також розробити стратегії реагування з урахуванням вартості реалізації контрзаходів.

Методології ідентифікації та оцінки ризиків

Основні підходи до ідентифікації ризиків у КФС можна поділити на:
експертно-аналітичні (наприклад, метод Delphi);
статистичні (аналіз логів інцидентів, частотних розподілів);
моделювальні (симуляції на основі сценаріїв, Petri Net, системної динаміки);
інтелектуальні (з використанням машинного навчання для виявлення аномалій) [44], [45].

Зокрема, для реальних систем із великою кількістю вхідних параметрів актуальним є використання методів нечіткої логіки (Fuzzy Logic), які дозволяють формалізувати лінгвістичні оцінки, наприклад: "низький ризик втрати керування" або "висока ймовірність збоїв сенсорної мережі" [46].

Архітектурні моделі ризик-менеджменту

Ефективне управління ризиками потребує чіткого архітектурного поділу відповідальностей. У типовій структурі КФС можна виокремити:
рівень сенсорної діагностики – збір і первинна обробка даних про стан об'єктів;
рівень локального контролю – виявлення відхилень на місцевих вузлах;
рівень глобального аналізу – об'єднання даних, формування ризикових патернів, моделювання сценаріїв;
рівень ухвалення рішень – розробка та реалізація відповідних заходів [47], [48].

У новітніх реалізаціях додається рівень цифрового двійника, який уможливлює прогнозувати ризики шляхом імітації роботи системи віртуально до втручання в реальний об'єкт [49].

Методи обробки ризиків: запобігання, зменшення, передача, прийняття

1. Запобігання ризику реалізується через архітектурне резервування, подвійне сенсорне дублювання, багатофакторну автентифікацію та ізоляцію критичних компонентів.

2. Зменшення ризику – впровадження IDS/IPS, алгоритмів самовідновлення, інструментів виявлення аномалій на основі машинного навчання [50], [51].

3. Передача ризику – укладання контрактів із постачальниками послуг кіберзахисту або страхування відповідальності за збої у роботі.

4. Прийняття ризику – допустиме в окремих випадках для некритичних компонентів із низьким впливом на загальну систему [52].

Інструменти ризик-аналізу

Для моделювання сценаріїв ризиків в КФС активно використовуються такі інструменти:

FTA (Fault Tree Analysis);

FMEA (Failure Mode and Effects Analysis);

HAZOP (Hazard and Operability Study);

BOW-TIE-моделі, що поєднують причинно-наслідковий та бар'єрний аналіз [53], [54].

Упровадження подібних підходів особливо актуальне для підприємств, що сертифікуються за стандартами ISA/IEC 62443 та ISO 27005 [55], [56].

Стандарти та фреймворки ризик-менеджменту

Ключові стандарти, що регламентують управління ризиками в КФС, включають:

IEC 62443 – багаторівневий підхід до безпеки промислових систем;

NIST SP 800-30 – методологія оцінки ризиків у кіберфізичних ландшафтах;

ISO/IEC 27005 – загальна методологія інформаційного ризик-менеджменту;

ISA 95 – інтеграція ризик-менеджменту у виробничі процеси [57], [58].

Інтелектуальні методи адаптивного управління ризиками

Інтеграція інтелектуальних технологій у процеси управління ризиками кіберфізичних систем (КФС) відкриває нові горизонти в напрямі динамічної адаптації до змінних умов експлуатації. Традиційні підходи, що базуються на фіксованих шаблонах загроз або статичних правилах, не здатні забезпечити належну гнучкість у середовищах із високим ступенем динамізму та багатовимірності. З цією метою застосовуються нейронні мережі, генетичні алгоритми, методи підкріплювального навчання та нечіткі експертні системи, які дозволяють не лише розпізнавати ризикові патерни, але й проактивно формувати рекомендації з реагування [59]. Адаптивні системи аналізу ризику можуть працювати в режимі реального

часу, використовуючи розподілену інфраструктуру (edge/fog/cloud), при цьому забезпечуючи масштабовану обробку великих потоків телеметричних даних. Моделі глибокого навчання (Deep Learning) дозволяють виявляти складні кореляції між станами сенсорів, затримками в мережі, поведінкою користувачів і станом обладнання. Особливу ефективність демонструють згорткові нейромережі (CNN) для аналізу просторових шаблонів та рекурентні мережі (LSTM, GRU) для виявлення часових аномалій у телеметрії [60].

Контекстно-орієнтовані моделі ризику

Високий рівень гетерогенності КФС потребує не лише структурного аналізу, але й глибокого контекстуального розуміння ситуації. Для цього застосовуються контекстно-залежні моделі ризику, в яких враховуються чинники довкілля, тип об'єкта, час доби, мережеве навантаження, енергетичний баланс, а також поведінкові патерни операторів і користувачів [61]. Подібні моделі реалізуються за допомогою онтологій, що описують типи загроз, структури взаємодії, логіку процесів та семантику даних. Завдяки семантичному аналізу стає можливим не лише виявлення нетипових сценаріїв, але й перевірка узгодженості з нормативними вимогами та політиками безпеки. Наприклад, у енергосистемах використання онтологічного опису дозволяє зіставляти стан мережі з допустимими параметрами нормативу IEC 61850, що критично важливо для запобігання аваріям [62].

Байєсівські та гібридні підходи

Одним із поширених інструментів для оцінювання ризиків у КФС є байєсівські мережі – імовірнісні графові моделі, що дозволяють враховувати як відомі причинно-наслідкові зв'язки, так і невизначені або частково доступні параметри. Ці моделі здатні оновлюватися в режимі реального часу в міру надходження нової інформації, що забезпечує динамічну адаптацію до змін стану системи [63]. Гібридні методи, які поєднують логіку на основі правил, стохастичне моделювання та елементи машинного навчання, демонструють високу стійкість у складних сценаріях. У роботі таких систем може брати участь ансамбль моделей, кожна з яких відповідає за конкретну підсистему (наприклад, обробку сенсорних даних, аналіз енергоспоживання або детекцію вторгнень), а фінальне рішення приймається через механізм агрегування ризикових оцінок [64].

Цифрові двійники в управлінні ризикам

Однією з найефективніших концепцій сучасного ризик-менеджменту є впровадження цифрових двійників (Digital Twins, DT), які забезпечують моделювання поведінки реальних об'єктів у віртуальному середовищі з

високою точністю. Такі моделі дозволяють симулювати атаки, зміну конфігурацій, збої у комунікації та інші несприятливі чинники без ризику пошкодження реального обладнання [65]. Цифрові двійники використовуються не лише для виявлення потенційних точок відмов, а й для побудови сценаріїв відновлення, оптимізації технічного обслуговування та формування рекомендацій щодо резервування критичних елементів. У поєднанні з аналітикою Big Data та хмарними платформами, DT стали основою для створення стратегічних систем управління ризиками у промисловості 4.0 [66].

Галузеві приклади реалізації

В енергетичних КФС широко застосовуються нейромережеві оцінки стійкості мережі до аварій (перенавантаження, втрати синхронізації), які дозволяють у реальному часі перерозподіляти потоки енергії, уникати каскадних збоїв та прогнозувати баланс попиту і пропозиції [67]. У транспортному секторі, зокрема в автономних логістичних системах, ризик-аналіз охоплює оцінку відмов GPS, збоїв у V2X-комунікаціях, перешкод на маршруті та втрати зв’язку з оператором. Для цього застосовуються гібридні моделі, які поєднують дані з датчиків, картографії та попередніх маршрутів [68]. У промисловості впроваджуються системи FMEA-аналізу, інтегровані з цифровими платформами моніторингу. Вони дозволяють ранжувати вузли за ймовірністю збоїв, формувати пріоритети технічного обслуговування та оновлювати політики безпеки на основі реального досвіду експлуатації [69].

Прогностичні інструменти та інтеграція з DevSecOps

Актуальним напрямом розвитку є інтеграція ризик-менеджменту в цикли DevSecOps, що дозволяє реалізовувати безпеку «за замовчуванням» ще на етапі проєктування системи. Прогностичні інструменти, що базуються на історичних даних і трендовому аналізі, інтегруються безпосередньо в CI/CD-процеси, автоматично оцінюючи ризики при зміні конфігурацій або впровадженні оновлень [70]. Особливої популярності набувають платформи, що поєднують безпеку, продуктивність і ризик-аналіз у межах єдиного контролера. Такі архітектури дозволяють не лише виявляти нові загрози, але й оцінювати їхній вплив на продуктивність та вартість підтримки [71].

Автоматизація оцінки ризиків у режимі реального часу

У системах із високими вимогами до режиму часу реагування критично важливою є здатність автоматично оцінювати ризики без участі людини. Для цього впроваджуються механізми обчислення так званих адаптивних порогів впливу (Adaptive Impact Thresholds), що враховують

динаміку змін параметрів та дозволяють підлаштовувати реакцію системи до контексту [72]. Наприклад, у КФС для хімічного виробництва зміна температури або тиску може бути нормальною у штатному режимі, але критичною у разі супутнього спаду енергоживлення. У цьому випадку автоматизовані модулі ризик-оцінки повинні зіставляти кілька параметрів одночасно й оцінювати їх кумулятивний вплив [73].

Експертні системи та інтерпретованість рішень

Хоча нейромережі демонструють високу точність прогнозів, одним із ключових викликів залишається забезпечення інтерпретованості прийнятих рішень. Тому у практиці ризик-менеджменту активно застосовуються експертні системи на основі продукційних правил, які дозволяють пояснити, чому певна подія класифікується як критична або безпечна [74]. Такі системи особливо важливі у регульованих галузях (наприклад, фармацевтиці або аерокосмічній інженерії), де кожне рішення повинно мати документоване обґрунтування і проходити аудит. Крім того, модулі explainable AI (XAI) дозволяють вбудовувати інтерпретовані моделі у глибокі мережі, зберігаючи баланс між точністю та прозорістю [75], [76].

Моделі реагування на інциденти в кіберфізичних системах

Управління ризиками в КФС не обмежується етапом ідентифікації та оцінки потенційних загроз – не менш важливою є здатність до ефективного реагування на інциденти, з урахуванням режиму реального часу, пріоритетів і впливу на фізичні об’єкти. Типова модель реагування включає фазу детекції, верифікації, локалізації впливу, вибору сценарію відповіді, реалізації протидії та оцінювання результатів. Вона може бути реалізована як централізовано (через SOC або центральні вузли керування), так і децентралізовано (через локальні інтелектуальні агенти) [77]. Особливої уваги потребує автоматизоване коригування політик безпеки та контроль впливу рішень на технічні процеси. У виробничих КФС, наприклад, невдале реагування може призвести до аварійної зупинки конвеєра, зриву графіка чи руйнування обладнання. Тому в сучасних системах застосовуються предиктивні сценарні моделі реагування, що ґрунтуються на історичних шаблонах, оперативному контексті та поточному ризиковому профілі системи [78].

Інтеграція механізмів самовідновлення

Ключовим аспектом стійкості КФС до ризиків є здатність до самовідновлення після інцидентів. Це досягається за рахунок побудови резервних шляхів обміну даними, дублювання керувальних модулів, використання альтернативних джерел енергії та алгоритмів повторного конфігурування. У гетерогенних середовищах активно застосовуються механізми agent-based recovery – розподілені агенти, здатні автономно

ініціювати перезавантаження модулів, перенаправлення трафіка чи перепризначення задач [79]. Підхід до самовідновлення нерозривно пов'язаний із застосуванням цифрових близнюків, які дозволяють змоделювати динаміку аварійного розвитку ситуації та оцінити ефективність альтернативних стратегій відновлення. У транспортних КФС такі стратегії реалізуються через резервні маршрути автономного пересування, у енергетиці – через миттєве перемикання навантаження на інші лінії живлення [80].

Роль мультиагентних систем в управлінні ризиками

Мультиагентні системи (MAS) все активніше впроваджуються у КФС як засіб реалізації децентралізованого та адаптивного управління ризиками. Вони базуються на взаємодії незалежних, але координованих агентів, кожен з яких виконує окремі функції – моніторинг, оцінку загроз, прогнозування наслідків, прийняття локальних рішень [81]. Перевага таких систем полягає у їхній здатності до колективного реагування в умовах часткової доступності інформації або недоступності центрального контролера. MAS використовуються, наприклад, у смарт-мережах, де енергогенератори, накопичувачі та споживачі координують баланс навантаження з урахуванням локальних ризиків (перевантаження, атаки на IoT-шлюзи, коливання частоти тощо) [82].

Концепція операційної стійкості (cyber resilience)

Поняття стійкості (resilience) все частіше виступає центральним у фреймворках ризик-менеджменту для КФС. На відміну від традиційної моделі безпеки, resilience фокусується не лише на запобіганні інцидентам, але й на здатності системи зберігати функціональність, адаптуватися до змін, відновлюватися після порушень і навіть навчатися на помилках [83].

Операційна стійкість включає чотири компоненти:

anticipate – прогнозування загроз;

withstand – здатність функціонувати попри інцидент;

recover – оперативне відновлення;

evolve – адаптація стратегії захисту.

Цей підхід є критично важливим у середовищах із високим рівнем динаміки та взаємозалежностей (наприклад, «розумні» міста, розподілене виробництво, медичні IoT-системи) [84].

Динамічне оцінювання ризиків у режимі реального часу

Одним із викликів в управлінні ризиками КФС є необхідність безперервного оновлення ризикових профілів у режимі реального часу. Для цього використовуються потоки телеметричних даних, агреговані з edge-

вузлів, у поєднанні з історичними шаблонами інцидентів та онтологічною інтерпретацією контексту. Інструментом є, наприклад, потокова обробка подій (complex event processing, CEP), яка дозволяє виявляти складні сценарії ризиків на основі взаємозалежних подій [85]. Крім того, формуються індекси агрегованого ризику (Risk Index Aggregator), які відображають загальний стан системи і використовуються для визначення рівня допуску або заборони на певні дії, наприклад, активацію нового виробничого модуля чи розгортання оновлення в хмарі [86].

Критерії ефективності стратегій ризик-менеджменту

Оцінка ефективності реалізованих методів управління ризиками передбачає аналіз не лише технічних метрик, а й економічних, експлуатаційних та регуляторних показників. До ключових належать:

MTTR (mean time to recovery) – середній час відновлення функціональності після інциденту;

RTO/RPO – допустимий час простою та втрати даних;

кількість виявлених інцидентів на одиницю часу;

коефіцієнт хибнопозитивних спрацювань [87].

У галузях, де управління ризиками має юридичні наслідки (наприклад, фармацевтика, авіація, оборона), додатково враховуються вимоги сертифікації, стандарти верифікації модулів та критерії аудиторської відстежуваності [88].

Використання сценарних симуляторів ризиків

Для тестування реакцій системи на екстремальні сценарії широко застосовуються програмні симулятори ризиків, які дозволяють у безпечному довкіллі моделювати атаки, аварії, втрати зв’язку або некоректну поведінку компонентів. Ці симуляції дають змогу виявити слабкі місця, які не фіксуються при стандартному тестуванні [89]. Симулятори використовують підходи агентного моделювання, системної динаміки, логіки подій, дискретних автоматів. Наприклад, у роботизованих системах імітується спотворення даних із сенсорів позиціонування або атакуюча команда на виконавчий модуль. Це дозволяє протестувати сценарії без фізичного втручання [90].

Тенденції подальшого розвитку

Сучасний стан управління ризиками в КФС демонструє перехід від статичних політик до адаптивних, контекстно-чутливих і самооптимізованих систем. У перспективі основними тенденціями є:

застосування quantum-safe алгоритмів у захисті моделей;

поєднання цифрових двійників і предиктивного моделювання на базі AI;

впровадження кіберстрахування як частини управління ризиками;

розвиток інтероперабельних фреймворків оцінювання у мультиагентних екосистемах [91], [92].

Окремим напрямом є використання нейроінтерпретованих стратегій ризик-менеджменту, де архітектура контролера змінюється залежно від метарівня аналізу, тобто не лише оцінка ризику, а також оцінка достовірності самого оцінювання [93]. Інтеграція всіх елементів у єдине стратегічне бачення, де ризик оцінюється як динамічна характеристика системи, є ключем до створення стійких, самонавчальних, інтероперабельних кіберфізичних систем нового покоління [94].

Висновки до розділу

Розділ дає системне уявлення про теоретичні засади функціонування, розвитку та структури кіберфізичних систем (КФС) як міждисциплінарного феномену, що знаходиться на стику кібернетики, автоматизації, інформаційних технологій та інженерії. На основі аналізу історії становлення КФС, основних архітектурних моделей, типових вразливостей і сучасних підходів до управління ризиками, сформовано узагальнене бачення ключових напрямів теоретичного осмислення та практичної реалізації таких систем.

Перш за все, встановлено, що КФС пройшли складну еволюцію від перших елементів автоматизованого контролю до складних багаторівневих інтелектуальних екосистем, які забезпечують злагоджену взаємодію фізичного середовища, програмного керування та комунікаційних технологій. Цей розвиток супроводжувався впровадженням SCADA-систем, мережевих протоколів, мікропроцесорної бази та, згодом, технологій штучного інтелекту, IoT, Big Data, цифрових близнюків і хмарної обробки. У межах визначення і структури КФС встановлено, що вони складаються з фізичних пристроїв, мережі зв’язку, обчислювального довкілля, когнітивного рівня аналізу й управління, та підсистеми безпеки. У кожній з цих складових переважають певні виклики: на фізичному рівні – надійність і чутливість сенсорів; на мережевому – забезпечення синхронізації і захисту; на обчислювальному – продуктивність і адаптивність; на когнітивному – інтерпретованість рішень; у безпеці – інтегрованість захисту на всіх шарах системи. Окрему увагу приділено проблематиці функціональних компонентів КФС. Встановлено, що сенсорні, керуючі, мережеві та виконавчі рівні утворюють замкнений контур зворотного зв’язку, який в умовах швидкозмінного ландшафту повинен діяти із мінімальною затримкою.

Використання периферійних і хмарних обчислень, цифрових близнюків та інтелектуальних механізмів прийняття рішень є необхідною умовою підвищення оперативності та автономності таких систем. Розглянута структура загроз та вразливостей КФС свідчить про багаторівневу й багатовимірну природу ризиків, що охоплюють як

класичні інформаційні загрози, так і фізичні, соціальні, техногенні та програмні аспекти. Зокрема, критичними виявляються вразливості на рівні сенсорної інфраструктури, відкритих мережевих протоколів, відсутність сегментації мереж, використання небезпечних оновлень ПЗ, людський фактор, та несанкціоноване втручання у ланцюг постачання. У реальних прикладах – від вірусу Stuxnet до інцидентів у транспорті та охороні здоров'я – продемонстровано потенційну небезпечність експлуатації КФС без належного ризик-менеджменту.

Розглянуті методи управління ризиками підтверджують доцільність переходу від традиційної безпекової моделі до концепції адаптивного багатошарового управління з урахуванням імовірнісних, поведінкових, сценарних і когнітивних оцінок. КФС потребують безперервного, автоматизованого оцінювання ризиків у режимі реального часу з інтеграцією результатів в архітектуру системи керування. Основні стратегії управління включають запобігання, зменшення, передачу та прийняття ризику. Визначальними є здатність до самовідновлення, децентралізованого реагування та гнучкого масштабування захисту. Інтеграція моделей глибокого навчання, байєсівських мереж, нейрофаззі-контролерів, цифрових двійників та DevSecOps-підходів забезпечує сучасні підходи до детекції, прогнозування і реагування на загрози. При цьому особливе значення мають Explainable AI-рішення, які дозволяють формувати інтерпретовані, аудитоздатні та нормативно сумісні системи керування ризиками. Автоматизоване симулювання інцидентів, розгортання оновлень за CI/CD-підходом, а також створення адаптивних маршрутів ухвалення рішень формують передумови досягнення повноцінної кіберстійкості.

Таким чином, результати проведеного аналізу дозволяють сформулювати такі ключові висновки:

Кіберфізичні системи стали фундаментом інфраструктури індустріального, енергетичного, транспортного та медичного секторів, що вимагає переосмислення безпеки як складової технічної архітектури.

Ризики в КФС мають складну природу – вони виникають у цифровій, фізичній, логічній та соціальній площинах, отже, потребують комплексного підходу до виявлення та нейтралізації.

Ефективне управління ризиками потребує побудови багатоагентних моделей контролю, використання цифрових близнюків, прогнозно-адаптивного аналізу, машинного навчання та динамічного конфігурування захисних механізмів.

Основу майбутніх КФС становитиме інтелектуальний, інтероперабельний і стійкий до збоїв фреймворк з можливістю самонавчання, аудиту, локального реагування і автономного масштабування.

Виклики цифровізації, кіберзагроз і фізичного довкілля повинні бути вирішені в рамках об'єднаної наукової парадигми кібербезпеки, інженерії керування, когнітивної аналітики та регуляторної відповідності.

Ураховуючи вищенаведене, теоретичні основи кіберфізичних систем стають надійною базою для формування нових поколінь захищених, масштабованих і контекстно-адаптивних технологій, здатних до стійкого функціонування в умовах невизначеності, високої складності та постійної загрози зовнішнього впливу.

Джерела:

39. Wiener, N. Cybernetics: Or Control and Communication in the Animal and the Machine. Paris: Hermann & MIT Press, 1948. DOI: відсутній.

40. Anton, S. D., Fraunholz, D., Lipps, C., Pohl, F., Zimmermann, M., & Schotten, H. D. Two decades of SCADA exploitation: A brief history. IEEE Conference on Artificial Intelligence and Network Security, 2019. DOI: https://doi.org/10.1109/AINS.2017.8270432.

41. Tamy, S., Belhadaoui, H., Rabbah, M. A., Rabbah, N., & Rifi, M. SCADA communication real time protocols. Indian Journal of Science and Technology, 2019, vol. 12(34). DOI: https://doi.org/10.17485/ IJST/2019/ 12I34/147550.

42. Rajkumar, R., Lee, I., Sha, L., & Stankovic, J. Cyber-physical systems: The next computing revolution. Design Automation Conference (DAC), 2010. DOI: https://doi.org/10.1145/1837274.1837461.

43. Thramboulidis, K. A cyber-physical system-based approach for industrial automation systems. Computers in Industry, 2015, vol. 74, pp. 11–25. DOI: https://doi.org/10.1016/j.compind.2015.04.006.

44. Sehr, M. A., Lohstroh, M., Weber, M., et al. Programmable logic controllers in the context of Industry 4.0. IEEE Transactions on Industrial Informatics, 2021. DOI: https://doi.org/10.1109/TII.2020.3007764.

45. Bangemann, T., Riedl, M., Thron, M., & Diedrich, C. Integration of classical components into industrial cyber–physical systems. Proceedings of the IEEE, 2016, vol. 104(5), pp. 947–960. DOI: https://doi.org/10.1109/ JPROC.2015.2510981.

46. Rajabpour, N., & Sedaghat, Y. A hybrid-based error detection technique for PLC-based industrial control systems. Emerging Technologies and Factory Automation (ETFA), 2015. DOI: https://doi.org/10.1109/ ETFA.2015.7301525.

47. Yadav, G., & Paul, K. Architecture and security of SCADA systems: A review. International Journal of Critical Infrastructure Protection, 2021, vol. 34, 100433. DOI: https://doi.org/10.1016/j.ijcip.2021.100433.

48. Pivoto, D. G. S., Almeida, L. F. F. de, Righi, R. da R., Rodrigues, J. J. P. C., Lugli, A. B., & Alberti, A. M. Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0. Journal of

Manufacturing Systems, 2021, vol. 58, pp. 176–192. DOI: https://doi.org/10.1016/j.jmsy.2020.11.017.

49. Oks, S. J., Jalowski, M., Lechner, M., et al. Cyber-physical systems in the context of Industry 4.0: A review, categorization and outlook. Information Systems Frontiers, 2022. DOI: https://doi.org/10.1007/s10796-022-10252-x.

50. Su, W., Machica, I. K., Xu, G., He, Z., & Kong, Y. Industrial cyber intelligent control operating system hybrid with IEC 61499 and big data. IEEE Conf. on Artificial Intelligence and Big Data, 2022. DOI: https://doi.org/10.1109/icaibd55127.2022.9820278.

51. Mutua, E. Cyber-physical systems and their role in Industry 4.0. Journal of Technology and Systems, 2024. DOI: https://doi.org/10.47941/jts.2149.

52. Singh, K. Cyber-physical systems and Industry 4.0. In: Cybersecurity and Secure Information Systems, 2022, pp. 113–130. DOI: https://doi.org/10.1007/978-3-031-18239-6_5.

53. Wang, L., & Wang, G. Big data in cyber-physical systems, digital manufacturing and Industry 4.0. International Journal of Engineering and Manufacturing, 2016, vol. 6(4), pp. 1–8. DOI: https://doi.org/10.5815/IJEM.2016.04.01.

54. Thakur, P., & Sehgal, V. K. Emerging architecture for heterogeneous smart cyber-physical systems. Computers & Industrial Engineering, 2021, vol. 162, 107750. DOI: https://doi.org/10.1016/j.cie.2021.107750.

55. Pivoto, D. G. S., et al. Cyber-physical systems architectures for industrial internet of things applications. Journal of Manufacturing Systems, 2021, vol. 58, pp. 176–192. DOI: https://doi.org/10.1016/j.jmsy.2020.11.017.

56. Ramanathan, L., & Nandhini, R. S. Cyber-physical system—An architectural review. In: Cyber Physical Systems: A Computational Perspective. Singapore: Springer, 2021. DOI: https://doi.org/10.1007/978-981-16-0739-4_13.

57. Plakhotnikov, D. P., & Kotova, E. E. Design and analysis of cyber-physical systems. IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, 2021. DOI: https://doi.org/10.1109/ELCONRUS51938.2021.9396364.

58. Mishra, A., & Ray, A. K. A novel layered architecture and modular design framework for next-gen cyber physical system. IEEE Int. Conf. on Computing, Communication and Intelligent Systems (ICCCI), 2022. DOI: https://doi.org/10.1109/iccci54379.2022.9740757.

59. Lankhorst, T. J. W. An architectural approach to cyber-physical system design. In: Enterprise Architecture at Work. Heidelberg: Springer, 2021. DOI: відсутній.

60. Pickles, D. Formalization of cyber-physical system interface using discrete event system specifications. PhD Dissertation. Carleton University, 2022. DOI: https://doi.org/10.22215/etd/2022-14851.

61. Machine learning based IoT models and analysis. In: Machine Learning for Cyber Physical Systems. Boca Raton: CRC Press, 2021, ch. 21. DOI: https://doi.org/10.1201/9781003156406-21.

62. Karthick, G. S., & Sumathi, V. Cyber-physical systems. In: Handbook of Research on Innovations and Applications of AI, IoT, and Cognitive Technologies. IGI Global, 2022, ch. 1. DOI: https://doi.org/10.4018/978-1-6684-7879-0.ch001.

63. Zhang, W., & Zhang, L. Designing and modeling cyber physical systems by a service-based approach. IEEE Int. Conf. on System Science and Engineering (ICSESS), 2015. DOI: https://doi.org/10.1109/ ICSESS. 2015.7339146.

64. Kovacshazy, T. Distributed architecture for real-time cyber-physical system, time-sensitive networks. IEEE Carpathian Communication Conference, 2018. DOI: https://doi.org/10.1109/CARPATHIANCC.2018.8399588.

65. Rehman, S. U., Iannella, A., & Gruhn, V. A security based reference architecture for cyber-physical systems. In: Critical Infrastructure Protection XII. Springer, 2019. DOI: https://doi.org/10.1007/978-3-030-01535-0_12.

66. Lin, J. Cyber physical systems. In: Handbook of Research on Cloud Computing and Big Data Applications in IoT. IGI Global, 2019. DOI: відсутній.

67. Malhotra, J., Iqbal, F., Sahu, A. K., & Jha, S. A cyber-physical system architecture for smart manufacturing. In: Handbook of Industry 4.0 and Smart Systems. Springer, 2020. DOI: https://doi.org/10.1007/978-981-32-9417-2_53.

68. Rajkumar, R., de Niz, D., & Klein, M. Cyber-Physical Systems. Monograph. Addison-Wesley, 2017. DOI: відсутній.

69. Lee, J., Bagheri, B., & Kao, H.-A. A cyber-physical systems architecture for Industry 4.0-based manufacturing systems. Manufacturing Letters, 2015, vol. 3, pp. 18–23. DOI: https://doi.org/ 10.1016/ j.mfglet.2014.12.001.

70. Givehchi, O., Landsdorf, K., Simoens, P., & Colombo, A. W. Interoperability for industrial cyber-physical systems: An approach for legacy systems. IEEE Trans. on Industrial Informatics, 2017, vol. 13(6), pp. 3370–3378. DOI: https: //doi.org/10.1109/TII.2017.2740434.

71. Sharevski, F., & Oteafy, S. Security for cyber-physical systems: Leveraging cellular networks and fog computing. arXiv preprint, 2018. DOI: https://doi.org/10.48550/arXiv.1806.11053.

72. Hofer, F. Architecture, technologies and challenges for cyber-physical systems in Industry 4.0: A systematic mapping study. In: ACM/IEEE Int. Symposium on Empirical Software Engineering and Measurement (ESEM), 2018. DOI: https://doi.org/10.1145/3239235.3239242.

73. Tao, F., Qi, Q., Wang, L., & Nee, A. Digital twins and cyber–physical systems toward smart manufacturing and Industry 4.0: Correlation and comparison. Engineering, 2019, vol. 5(4), pp. 653–661. DOI: https://doi.org/10.1016/j.eng.2019.01.014.

74. Humayed, A., Lin, J., Li, F., & Luo, B. Cyber-physical systems security — A survey. IEEE Systems Journal, 2017, vol. 11(4), pp. 3847–3871. DOI: https://doi.org/10.1109/JSYST.2016.2622260.

75. Khaitan, S. K., & McCalley, J. D. Design techniques and applications of cyberphysical systems: A survey. IEEE Systems Journal, 2015, vol. 9(2), pp. 350–365. DOI: https://doi.org/10.1109/JSYST.2014.2322503.

76. Jirkovsky, V., Obitko, M., & Marik, V. Understanding data heterogeneity in the context of cyber-physical systems integration. IEEE Trans. on Industrial Informatics, 2017, vol. 13(2), pp. 660–668. DOI: https://doi.org/10.1109/TII.2016.2596101.

77. Lu, C., Saifullah, A., Li, B., Sha, M., Gonzalez, H., Gunatilaka, D., & Chen, Y. Real-time wireless sensor-actuator networks for industrial cyber-physical systems. Proceedings of the IEEE, 2016, vol. 104(5), pp. 1013–1024. DOI: https://doi.org/10.1109/JPROC.2015.2497161.

78. Jiang, Y., Yin, S., & Kaynak, O. Data-driven monitoring and safety control of industrial cyber-physical systems: Basics and beyond. IEEE Access, 2018, vol. 6, pp. 47374–47391. DOI: https://doi.org/10.1109/ ACCESS. 2018.2866403.

79. Fatima, I., Malik, S. U. R., Anjum, A., & Ahmad, N. Cyber physical systems and IoT: Architectural practices, interoperability, and transformation. IT Professional, 2020, vol. 22(3), pp. 46–54. DOI: https://doi.org/ 10.1109/MITP.2019.2912604.

80. Oks, S. J., Jalowski, M., Vogel-Heuser, B., et al. Cyber-physical systems in the context of Industry 4.0: A review, categorization and outlook. Information Systems Frontiers, 2022. DOI: https://doi.org/10.1007/s10796-022-10252-x.

81. Dobaj, J., Riel, A., Krug, T., et al. Towards digital twin-enabled DevOps for CPS providing architecture-based service adaptation & verification at runtime. In: Proc. of the 17th Int. Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS ’22), 2022. DOI: https://doi.org/10.1145/3524844.3528057.

82. Fritzsch, J., Bogner, J., Haug, M., et al. Adopting microservices and DevOps in the cyber-physical systems domain: A rapid review and case study. arXiv preprint, 2022. DOI: https://doi.org/10.48550/arXiv.2210.06858.

83. Chaâri, T., Koubâa, A., Youssef, H., Abid, M., & Toumi, S. (2016). Real-time wireless sensor-actuator networks for industrial cyber-physical systems. IEEE Proceedings, DOI: https://doi.org/10.1109/ JPROC. 2015.2497161.

84. Jamthe, A., & Poellabauer, C. (2019). Reliability in cyber-physical systems using a co-simulation framework. Remote Sensing, 11(19), 2252. DOI: https://doi.org/10.3390/rs11192252.

85. Zhou, Y., Yu, R., & Xie, S. (2021). A secure control learning framework for cyber-physical systems under sensor and actuator attacks. IEEE

Transactions on Cybernetics. DOI: https://doi.org/10.1109/ TCYB. 2020.3006871.

86. Hribernik, K. A., Wuest, T., & Thoben, K.-D. (2021). Autonomous, context-aware adaptive digital twins. Computers in Industry, 130, 103508. DOI: https://doi.org/10.1016/j.compind.2021.103508.

87. Lu, Y., Morris, K. C., & Frechette, S. (2016). Current standards landscape for smart manufacturing systems. NIST Special Publication. DOI: https://doi.org/10.6028/NIST.SP.1500-201.

88. Lee, J., Bagheri, B., & Kao, H. A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. Manufacturing Letters, 3, 18–23. DOI: https://doi.org/10.1016/j.mfglet.2014.12.001.

89. Chaâri, T. et al. (2016). Real-time wireless sensor-actuator networks for industrial cyber-physical systems. IEEE Proceedings. DOI: https://doi.org/10.1109/JPROC.2015.2497161.

90. Tao, F., Zhang, M., Liu, Y., & Nee, A. Y. C. (2020). Digital twin driven smart manufacturing: Connotation, reference model, applications and research issues. IEEE Transactions on Industrial Informatics, 15(4), 2405–2415. DOI: https://doi.org/10.1109/TII.2019.2950192.

91. Eckhart, M., & Ekelhart, A. (2019). Digital twins for cyber-physical systems security: State of the art and outlook. In: Security and Quality in Cyber-Physical Systems Engineering. Springer. DOI: https://doi.org/10.1007/978-3-030-25312-7_14.

92. Tao, F., Sui, F., Liu, A., Qi, Q., & Zhang, M. (2019). Digital twin-driven product design framework. Computers in Industry, 111, 103–115. DOI: https://doi.org/10.1016/j.compind.2019.07.004.

93. Eckhart, M., Winkler, C., & Ekelhart, A. (2023). A cyber digital twin framework to support cyber-physical systems security. In: IEEE SWC 2023 Proceedings. DOI: https://doi.org/10.1109/SWC57546.2023.10449161.

94. Eckhart, M., Winkler, C., & Ekelhart, A. (2024). Role of AI in digital twin cybersecurity: From anomaly detection to incident response. Artificial Intelligence Review. DOI: https://doi.org/10.1007/s10462-024-10805-3.

РОЗДІЛ 2
НЕЙРОННІ МЕРЕЖІ В УПРАВЛІННІ РИЗИКАМИ КІБЕРФІЗИЧНИХ СИСТЕМ

Кіберфізичні системи дедалі частіше виступають ядром критичної та промислової інфраструктури, де вразливості інформаційної підсистеми миттєво відбиваються на фізичних процесах. У першому розділі було окреслено понятійний апарат, класи активів і загроз, а також загальні підходи до оцінювання ризиків у КФС. Цей розділ розвиває логіку від загальної risk-моделі до інтелектуальних методів її реалізації: показує, як штучні нейронні мережі, завдяки здатності моделювати нелінійні залежності, працювати з великомасштабними, різнорідними та нестаціонарними даними стають практичною основою для виявлення, оцінювання та пом'якшення ризиків у режимі реального часу. Порівняно з традиційними алгоритмами, нейромережеві підходи забезпечують вищу чутливість до слабких сигналів, кращу узагальнювальну здатність і можливість виявляти невідомі або еволюційні сценарії атак, що особливо важливо для Zero-Day та «повзучих» вторгнень [95–101]. Предметом подальшого аналізу є як фундаментальні елементи нейромережевих моделей, так і їх прикладне розгортання в підсистемах безпеки КФС. Розділ узагальнює математичні основи штучного нейрона та роль активаційних функцій у формуванні нелінійності, що безпосередньо впливає на збіжність навчання й стійкість до шуму. Далі систематизуються архітектури, релевантні задачам ризик-менеджменту: від класичних MLP для класифікації агрегованих ознак до RNN/LSTM/GRU для часових рядів, CNN для просторових патернів у зображеннях та сигналах, автоенкодерів для безучительського виявлення аномалій і GAN для синтетичного розширення рідкісних або нових класів загроз. Акцент зроблено на гібридних поєднаннях на кшталт CNN+LSTM, що об'єднують просторову та часову обробку, а також на варіаційних і змагальних модифікаціях автоенкодерів, придатних для робастного контролю якості даних і підвищення чутливості детекторів [95–106]. Окрему увагу приділено методології навчання та експлуатації моделей у виробничих середовищах. Розглядаються контрольовані, напівконтрольовані та безконтрольні схеми, а також навчання з підкріпленням для задач адаптивного керування. Показано, як у КФС компенсувати дефіцит розмічених даних і дисбаланс класів через використання автоенкодерів, псевдорозмітки, активного навчання та генерації синтетичних прикладів. Сформульовано практичні рекомендації з регуляризації, валідації та калібрування порогів, що мінімізують перенавчання й керують компромісом між чутливістю та специфічністю в умовах різкого зміщення розподілів. У контексті життєвого циклу моделі проаналізовано континуальне та федеративне навчання, необхідні для підтримки актуальності детекторів за збереження приватності та відповідності нормативним вимогам [102–106]. Прикладний

блок розділу демонструє інтеграцію нейромереж у ключові контури безпеки КФС: інтелектуальні IDS із поєднанням сигнатурних правил і глибинних моделей; аналіз шкідливого ПЗ у статичній і динамічній постановках; поведінкове профілювання суб'єктів доступу в парадигмі Zero Trust; моніторинг IoT і SCADA з виявленням повільних відхилень телеметрії. Наведені кейси ілюструють, як гібридні архітектури підвищують повноту виявлення за прийнятного рівня помилкових спрацювань, скорочуючи час реагування та закриваючи прогалини традиційних підходів [100–106]. З огляду на вимоги до прозорості та аудиту в критичних доменах, у розділі також узагальнено підходи Explainable AI для інтерпретації рішень нейромережевих детекторів та їх узгодження з політиками безпеки підприємства. Обґрунтовано місце пояснюваності у процесах оцінювання ризиків, розподілу відповідальності та прийняття рішень оператором, а також взаємозв'язок із галузевими стандартами управління інформаційною безпекою. Це доповнюється оглядом етичних і правових обмежень застосування штучних нейронних мереж (ШНМ) у КФС, що визначають вимоги до даних, їх обробки та контролю якості.

Структурно матеріал подано від загальних принципів до конкретних реалізацій: спершу розкрито базові поняття та архітектури, далі методи навчання і валідації в умовах реальної експлуатації, після чого наведено приклади застосувань і підсумовано сильні сторони та обмеження нейромереж у контексті ризик-менеджменту. Така логіка уможливлює поєднати термінологічно узгодженість із практичною придатністю, забезпечуючи читачеві цілісну дорожню карту впровадження нейромережевих технологій безпеки на різних рівнях кіберфізичних систем. У підсумку розділ формує методичний каркас для інженерів і дослідників: від вибору архітектури з урахуванням типу даних і цільової метрики до побудови відтворюваних пайплайнів виявлення загроз, здатних до адаптації, масштабування та пояснення результатів у регульованих середовищах. Це створює підґрунтя для подальшої уніфікації термінів та інтеграції нейромережевих підходів у системи управління ризиками, визначені у першому розділі, та прокладає місток до практичних кейсів і шаблонів розгортання, розглянутих далі [95–106].

2.1. Загальні принципи застосування штучних нейронних мереж

Теоретичне обґрунтування

Штучні нейронні мережі (ШНМ) є обчислювальними системами, побудованими на принципах функціонування біологічних нейронів людського мозку. Їхня здатність до навчання, узагальнення та адаптації робить їх надзвичайно ефективними в задачах виявлення й управління ризиками в кіберфізичних системах (КФС) [95]. У нейронних мережах

кожен нейрон є спрощеною моделлю біологічного нейрона і складається з набору входів, які множаться на відповідні вагові коефіцієнти, сума яких обробляється за допомогою нелінійної активаційної функції [96]. Теоретична база застосування ШНМ у контексті управління ризиками базується на принципах машинного навчання, що передбачає автоматичне налаштування параметрів моделі на основі наявних даних [97]. Це дозволяє мережам «вчитися» ідентифікувати складні закономірності, які складно або неможливо описати традиційними алгоритмічними методами. Можливість навчання на великій кількості даних і здатність узагальнювати отриманий досвід на нових прикладах забезпечує ефективність ШНМ у задачах прогнозування та виявлення аномалій [98].

Універсальність моделей у задачах виявлення загроз

Однією з ключових переваг ШНМ є їхня універсальність, що дозволяє успішно застосовувати ці моделі в широкому спектрі задач, зокрема й у виявленні загроз КФС [99]. ШНМ можуть ефективно розв'язувати задачі класифікації, регресії, прогнозування, а також задачі виявлення аномалій. Останні особливо важливі в контексті забезпечення безпеки КФС, де необхідно швидко й точно ідентифікувати потенційно небезпечні події або поведінку системи [100]. Універсальність нейронних мереж забезпечується різноманітністю архітектур і гнучкістю підходів до навчання [101].

Наприклад, багатошарові персептрони (MLP) застосовуються для розпізнавання складних нелінійних взаємозв'язків, рекурентні мережі (RNN) – для роботи з послідовними даними, а згорткові нейронні мережі (CNN) є ефективними у задачах аналізу просторових структур, таких як зображення або сигнали [102]. Всі ці типи мереж мають загальні принципи побудови та функціонування, що дозволяє інтегрувати їх у комплексні системи управління ризиками КФС.

Математична модель нейрона

Основою будь-якої штучної нейронної мережі є математична модель нейрона. Класична модель штучного нейрона являє собою формалізоване представлення біологічного нейрона, що складається з набору вхідних сигналів $x_1, x_2, \dots, x_n$, кожен із яких відповідає сигналам, що отримує нейрон від інших нейронів або зовнішніх джерел інформації. Ці сигнали множаться на відповідні вагові коефіцієнти $w_1, w_2, \dots, w_n$, що відображають силу та важливість кожного вхідного сигналу:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \mathbf{w} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}, w_i, x_i \in \mathbb{R} \#(1)$$

Сумарний вхідний сигнал нейрона розраховується як скалярний добуток векторів вхідних сигналів та вагових коефіцієнтів із додаванням зміщення (bias):

$$z = \sum_{i=1}^{n} w_i x_i + b = \mathbf{w}^T \mathbf{x} + b, b \in \mathbb{R} \#(2),$$

де b – це параметр зміщення, що забезпечує можливість переміщення порогового значення активації нейрона вздовж осі абсцис. Зміщення дозволяє моделі ефективніше адаптуватися до різноманітних наборів даних та краще налаштовуватись під конкретні задачі.

Результат цієї суми проходить крізь активаційну функцію $f(z)$, яка є ключовим компонентом для забезпечення нелінійності вихідних сигналів нейрона. Загальна формула нейрона з активаційною функцією записується так:

$$y = f(z) = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \#(3)$$

Активаційна функція має важливе значення, оскільки вона дозволяє нейронній мережі апроксимувати складні нелінійні взаємозв'язки між вхідними та вихідними даними, що неможливо досягти звичайними лінійними методами. Наприклад, широко використовуються такі активаційні функції:

Сигмоїдна функція (sigmoid):

$$f(z) = \sigma(z) = \frac{1}{1 + e^{-z}} \#(4)$$

Гіперболічний тангенс (tanh):

$$f(z) = \tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \#(5)$$

Випрямлена лінійна одиниця (ReLU):

$$f(z) = \mathrm{ReLU}(z) = \max(0, z) \#(6)$$

Кожна з цих функцій має власні переваги й особливості використання. Наприклад, функція ReLU є простою для реалізації і дозволяє уникнути проблеми зникання градієнтів, тоді як sigmoid та tanh добре підходять для задач бінарної класифікації та інших специфічних сценаріїв. Таким чином, математична модель штучного нейрона є складним та гнучким інструментом, що дозволяє ефективно розв'язувати широкий спектр задач завдяки нелінійному перетворенню вхідних сигналів у вихідні результати.

Активаційні функції

Активаційні функції є критично важливим елементом у побудові та роботі штучних нейронних мереж, адже саме завдяки їм модель отримує здатність апроксимувати складні нелінійні взаємозв'язки між вхідними та вихідними даними. Вибір активаційної функції впливає на швидкість навчання, стійкість та ефективність роботи нейронних мереж у різних задачах, включаючи виявлення загроз у кіберфізичних системах (КФС).

Серед найпоширеніших активаційних функцій є sigmoid, tanh та ReLU, кожна з яких має свої математичні особливості, переваги та недоліки.

Sigmoid-функція

Sigmoid-функція визначається за формулою:

$$\sigma(z) = \frac{1}{1 + e^{-z}}, z \in \mathbb{R}. \#(1)$$

Sigmoid відображає вхідний сигнал на інтервал (0,1), що робить її дуже зручною для задач бінарної класифікації, оскільки вихід можна трактувати як імовірність належності до певного класу. Проте ця функція має суттєві недоліки: основним із них є проблема насичення (saturation) функції, коли похідна стає близькою до нуля на її крайніх значеннях. Це ускладнює навчання мережі, спричиняючи так звану проблему зникання градієнтів (vanishing gradients):

$$\sigma'(z) = \sigma(z)\big(1 - \sigma(z)\big). \#(2)$$

Похідна sigmoid завжди є додатною, але дуже малою на крайніх ділянках функції, що значно уповільнює процес навчання глибоких нейронних мереж.

Гіперболічний тангенс (tanh)

Функція гіперболічного тангенса описується виразом:

$$\tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}, z \in \mathbb{R}. \#(3)$$

Функція tanh відображає вхідні значення на інтервал ($-1,1$), що забезпечує середнє значення виходів нейронів близько нуля та сприяє більш ефективному навчанню мережі. Похідна $\tanh$ має вигляд:

$$\tanh'(z) = 1 - \tanh^2(z) \#(4)$$

Завдяки більш крутому нахилу, порівняно з sigmoid, tanh часто забезпечує швидше навчання. Проте функція tanh, як і sigmoid, також може страждати від проблеми зникання градієнтів у глибоких мережах.

Rectified Linear Unit (ReLU)

Функція ReLU, або випрямлена лінійна одиниця, визначається за простою формулою:

$$\mathrm{ReLU}(z) = \max(0, z), z \in \mathbb{R}. \#(5)$$

Головною перевагою ReLU є її простота та ефективність у тренуванні глибоких нейронних мереж. Похідна функції ReLU має вигляд:

$$\mathrm{ReLU}'(z) = \begin{cases} 1, & z > 0 \\ 0, & z \leq 0 \end{cases} \#(6)$$

ReLU розв'язує проблему зникання градієнтів для позитивних значень входу. Однак вона може стикатися з проблемою «мертвих» нейронів ($dyi\ ng$ReLU problem), коли нейрони перестають активуватись та оновлювати свої вагові коефіцієнти, якщо вихід нейрона стабільно стає меншим або рівним нулю.

Порівняння та рекомендації щодо вибору

Вибір активаційної функції залежить від особливостей задачі, типу нейронної мережі та особливостей даних. Загальні рекомендації:

Sigmoid – для бінарних класифікаційних задач або задач, де потрібна ймовірнісна інтерпретація виходу нейронної мережі.

Tanh – для задач, які потребують симетричного інтервалу вихідних значень та стабільного середнього значення, близького до нуля.

ReLU – рекомендовано для глибоких нейронних мереж та більшості задач через її швидкість та ефективність у процесі навчання.

Таким чином, правильний вибір активаційної функції є критично важливим етапом проєктування нейронних мереж для забезпечення оптимальної продуктивності та ефективності їх застосування в кіберфізичних системах.

Універсальність моделей штучних нейронних мереж (ШНМ) дозволяє ефективно використовувати їх у широкому спектрі задач, зокрема у сфері кіберфізичних систем (КФС), для ідентифікації потенційних загроз і ризиків. Здатність нейронних мереж до виконання різноманітних завдань, включаючи класифікацію, регресію, виявлення аномалій та прогнозування, робить їх потужним інструментом у забезпеченні стабільності та безпеки

КФС [95]. Завдання класифікації є одними з найпоширеніших у практиці застосування нейромереж, оскільки вони полягають у зарахуванні аналізованих об’єктів до певних класів на основі їх характеристик.

Багатошарові персептрони (MLP) є класичним прикладом архітектур, що демонструють високу точність та стабільність у класифікаційних задачах завдяки можливості виявляти та узагальнювати нелінійні закономірності [96]. Регресійні завдання також активно виконуються за допомогою нейронних мереж, адже ШНМ дозволяють моделювати складні нелінійні взаємозв’язки між вхідними параметрами та прогнозованими змінними [97]. Особливе місце серед завдань, що виконуються нейромережами, належить виявленню аномалій, що є ключовим компонентом систем управління безпекою КФС. Нейронні архітектури, так як автокодери, ефективно використовуються для виявлення відхилень від нормальної поведінки системи. Це реалізується за допомогою навчання мережі на прикладах нормальної роботи та виявлення суттєвих відмінностей між реальними й реконструйованими сигналами, що свідчать про наявність аномалій [98]. Прогнозування є ще одним критично важливим напрямом застосування ШНМ в управлінні ризиками КФС. Зокрема, рекурентні нейронні мережі (RNN), такі як мережі довгої короткострокової пам’яті (LSTM) та керовані рекурентні блоки (GRU), є ефективними завдяки своїй здатності враховувати часові залежності та довгострокові закономірності в даних [99]. Це дозволяє вчасно виявляти і попереджувати можливі несправності та загрози, забезпечуючи безперервність і надійність роботи КФС. Згорткові нейронні мережі (CNN) стали ще одним потужним засобом аналізу просторово-часових даних, зокрема в задачах, пов’язаних з аналізом зображень та сигналів у КФС [100]. Вони здатні автоматично виділяти найбільш значущі ознаки завдяки спеціалізованим шарам згортки та пулінгу, що робить їх ефективними в моніторингу та ідентифікації аномалій у реальному часі. Використання ШНМ має низку переваг, порівняно з традиційними підходами, такими як можливість глибокого навчання, здатність виявляти складні нелінійні залежності та висока точність прогнозування [101]. Однак існують і деякі обмеження, включаючи необхідність у великому обсязі навчальних даних, складність інтерпретації результатів роботи мережі, а також ризик перенавчання (overfitting) [102]. Важливість вибору правильної архітектури нейронної мережі та її оптимального налаштування для конкретних задач не може бути переоцінена. Вибір підходящої архітектури та методу навчання залежить від специфіки даних, наявних ресурсів та конкретних цілей завдання [103]. Застосування сучасних методів регуляризації, таких як dropout та batch normalization, допомагає уникнути перенавчання та поліпшити узагальнювальну здатність моделей [104].

Таким чином, завдяки універсальності та гнучкості нейронні мережі стають незамінними інструментами у вирішенні складних завдань

виявлення та управління ризиками в кіберфізичних системах, що підтверджується численними дослідженнями та практичними реалізаціями у цій галузі [105, 106].

Архітектури штучних нейронних мереж (ШНМ) мають суттєве значення в контексті управління ризиками кіберфізичних систем (КФС), оскільки правильний вибір архітектури нейронної мережі впливає на ефективність і точність вирішення конкретних завдань. У контексті КФС поширеними архітектурами є багатошарові персептрони (MLP), рекурентні нейронні мережі (RNN), мережі довгої короткострокової пам'яті (LSTM), керовані рекурентні блоки (GRU), згорткові нейронні мережі (CNN) та автокодери (autoencoders) [95]. Багатошаровий персептрон (MLP) є найпростішою та найпоширенішою архітектурою ШНМ, яка використовується для розв'язання задач класифікації та регресії. Він складається з одного або кількох прихованих шарів, що з'єднують вхідний і вихідний шари за допомогою повнозв'язних зв'язків. Завдяки глибоким архітектурам MLP здатні моделювати складні нелінійні взаємозв'язки між змінними, що є важливим для точного прогнозування станів і поведінки КФС [96]. Рекурентні нейронні мережі (RNN) вирізняються своєю здатністю працювати з послідовними даними та враховувати часові залежності. RNN використовують зворотні зв'язки, що дозволяє їм зберігати інформацію про попередні стани та ефективно застосовувати ці знання для аналізу та прогнозування часових рядів. Особливо ефективними в цьому контексті є архітектури LSTM та GRU, які розв'язують проблему "зникання градієнтів" завдяки спеціалізованим механізмам пам'яті [97]. Мережі довгої короткострокової пам'яті (LSTM) є різновидом рекурентних мереж, спеціально розроблених для запам'ятовування інформації на тривалі періоди часу. Кожен блок LSTM складається з декількох вентилів (воріт): вхідних, вихідних та вентилів забування. Ці вентилі контролюють потік інформації, дозволяючи мережі вибірково зберігати або забувати інформацію, що робить їх ідеальними для довгострокового прогнозування та виявлення складних залежностей у послідовностях даних КФС [98].

Керовані рекурентні блоки (GRU) є спрощеною версією LSTM, які мають лише два типи вентилів: вентиль оновлення і вентиль скидання. Це робить GRU менш обчислювально затратними і швидшими в навчанні порівняно з LSTM, зберігаючи при цьому здатність ефективно моделювати довгострокові залежності в даних. Завдяки цим перевагам GRU також широко використовуються в задачах прогнозування і моніторингу КФС [99].

Згорткові нейронні мережі (CNN) мають особливе значення у сфері КФС завдяки своїй здатності до ефективного аналізу просторових структур, таких як зображення та багатовимірні сигнали. CNN використовують шари згортки та пулінгу, які дозволяють автоматично виявляти ключові ознаки у вхідних даних, що є критично важливим у

задачах моніторингу, виявлення аномалій та ідентифікації загроз у режимі реального часу. CNN показують високу продуктивність у задачах аналізу відеопотоків та сигналів датчиків у КФС [100].

Автокодери є спеціалізованими нейронними мережами, призначеними для навчання ефективних представлень даних шляхом зниження їх розмірності. Автокодер складається з кодуючої частини, яка стискає інформацію, і декодуючої частини, яка намагається відновити початковий сигнал. В контексті КФС автокодери широко застосовуються для виявлення аномалій, оскільки значні відхилення між вихідним і реконструйованим сигналом вказують на потенційні проблеми або загрози [101]. Правильний вибір архітектури ШНМ має вирішальне значення для успішного застосування моделей у КФС. При виборі архітектури необхідно враховувати тип вхідних даних, характер задачі та обчислювальні ресурси, доступні для навчання і використання моделей. Ефективне застосування нейронних мереж також потребує використання сучасних методик оптимізації, таких як стохастичний градієнтний спуск, а також регуляризаційних методів, що дозволяють уникнути перенавчання моделей і підвищити їх узагальнювальну здатність [102–106]

Візуалізації:

Штучний нейрон – це базовий структурний елемент нейронної мережі, що імітує функціонування біологічного нейрона. Він приймає кілька вхідних сигналів, обробляє їх за допомогою математичних операцій формує один вихід. Саме завдяки цій здатності штучні нейрони можуть виявляти закономірності в даних і приймати рішення на їх основі. Подана нижче схема ілюструє основні етапи обробки сигналу всередині нейрона.

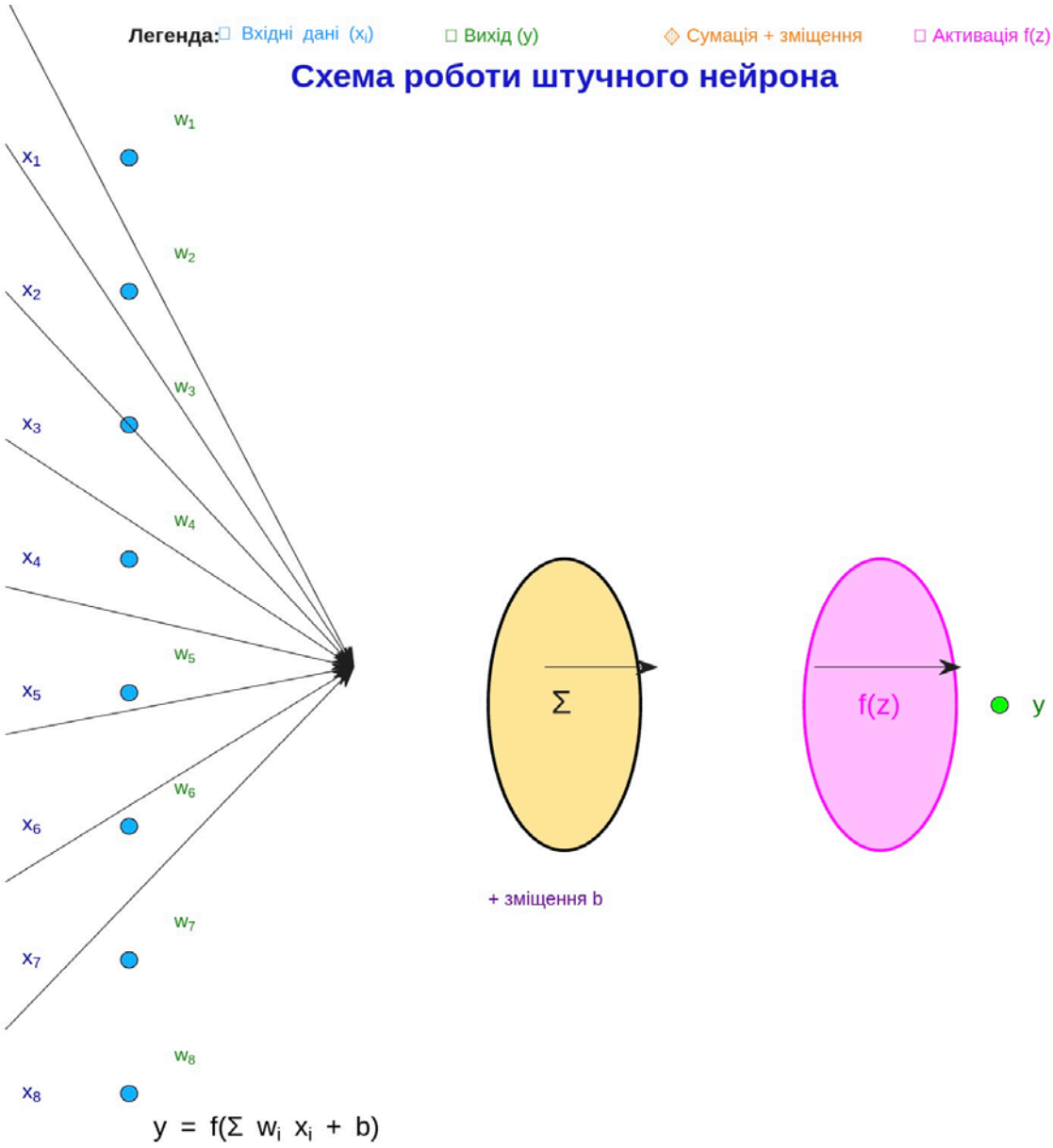


Рис. 1. Опис елементів схеми

Вхідні дані – це набір чисел, які надходять до нейрона ззовні або з попереднього шару мережі. Кожен із них подає певну характеристику чи події.

Вагові коефіцієнти – кожному вхідному сигналу відповідає власна вага, яка вказує на його важливість. Чим більша вага, тим більший вплив має цей сигнал на прийняте рішення.

Сумація сигналів – усі зважені вхідні значення підсумовуються. До результату також додається спеціальний параметр зміщення, який допомагає налаштувати гнучкість моделі.

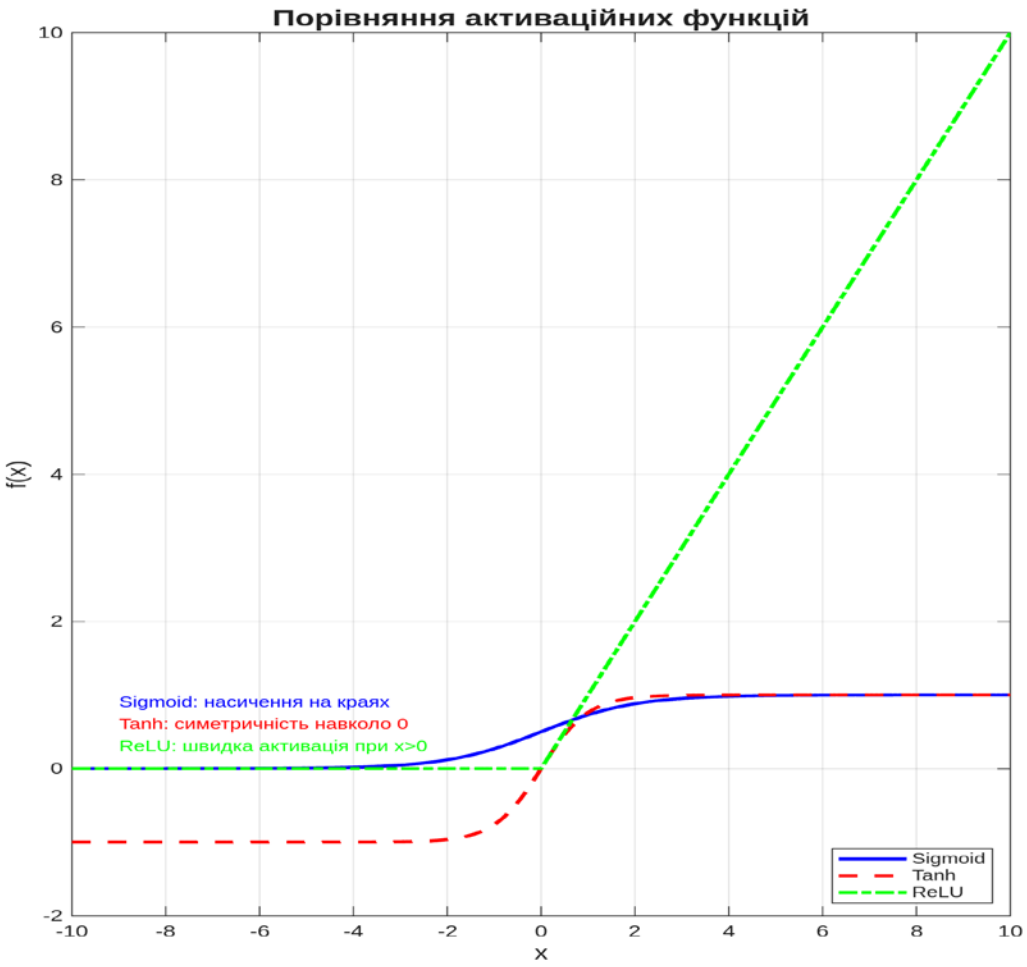


Рис. 2. Опис візуалізації

Активаційна функція – це спеціальний математичний механізм, який обробляє сумарний сигнал і вирішує, чи має нейрон “активуватись” і передати сигнал далі. Саме ця функція дозволяє моделі працювати з нелінійними залежностями.

Вихід нейрона – фінальний результат обробки. Він або передається далі в мережу, або використовується для прийняття рішення.

Активаційні функції

– ключовий компонент штучних нейронних мереж, що визначає, як саме нейрон реагує на вхідний сигнал. Саме завдяки цьому модель отримує здатність відображати складні, нелінійні залежності між даними. У цій візуалізації представлено графіки трьох найпоширеніших активаційних функцій: Sigmoid, Tanh та ReLU. Порівняння їхньої форми дозволяє

зрозуміти, як саме кожна функція впливає на процес навчання та ефективність моделі.

Sigmoid-функція (синя лінія) – м'яко "згладжує" вихід у діапазоні від 0 до 1. Підходить для задач, де потрібна ймовірнісна інтерпретація. Проте на великих або малих значеннях вхідного сигналу має насичення, тобто її вихід майже не змінюється, що ускладнює навчання (втрачається градієнт).

Гіперболічний тангенс – tanh (червона пунктирна лінія) – працює схоже до sigmoid, але має симетричний діапазон від –1 до 1. Це забезпечує "нульове центроване" навчання, що часто покращує швидкість сходження.

ReLU – випрямлена лінійна одиниця (зелена штрихова лінія) – найпростіша та одна з найефективніших функцій. Вона пропускає лише позитивні значення, відкидаючи всі негативні. Це дозволяє уникнути проблеми насичення та прискорює навчання, хоча може призводити до явища "мертвих нейронів", які не активуються.

Цінність порівняння

Ця візуалізація допомагає:

- *інтуїтивно зрозуміти, як функція трансформує вхідні сигнали;*
- *обґрунтовано обирати функцію для конкретної архітектури нейронної мережі;*
- *оцінити ризики та переваги кожної функції щодо стабільності градієнтів і швидкості навчання.*

Таке порівняння особливо важливе при розробці моделей для виявлення загроз у кіберфізичних системах, де точність і ефективність мають критичне значення.

Інформаційне порівняння

Різні архітектури штучних нейронних мереж (ШНМ) мають свої унікальні властивості, які визначають ефективність їх використання у конкретних типах задач. У контексті кіберфізичних систем (КФС) правильний вибір архітектури є критично важливим для досягнення точності, швидкодії та адаптивності моделей. Наведена нижче таблиця узагальнює ключові характеристики найпоширеніших типів нейронних мереж: багатошарових персептронів (MLP), рекурентних мереж (RNN) та згорткових мереж (CNN).

Таблиця 1

Порівняння архітектур нейронних мереж

Тип	Основна задача	Вхідні дані	Переваги	Приклади застосування
MLP	Класифікація / регресія	Статичні	Прості в реалізації	КФС стан / подія
RNN	Прогнозування	Послідовності	Часові залежності	Моніторинг
CNN	Виявлення ознак	Зображення / сигнали	Просторові патерни	Аналіз відео / датчиків

Таблиця 1

Тип – назва архітектури нейронної мережі.

Основна задача – тип задач, які найкраще вирішуються за допомогою відповідної архітектури (класифікація, прогнозування тощо).

Вхідні дані – формат даних, які модель може ефективно обробляти (статичні вектори, часові послідовності, зображення або сигнали).

Переваги – головні технічні або практичні переваги архітектури.

Приклади застосування – типові сценарії використання в межах кіберфізичних систем.

Це порівняння дозволяє швидко зорієнтуватися, яку архітектуру доцільно використовувати залежно від природи вхідних даних та цілей моделі – наприклад, стан моніторингу, виявлення аномалій або класифікація подій у системах промислової безпеки. За потреби таблицю можна розширити іншими архітектурами (LSTM, GRU, Autoencoder) для більш повного охоплення.

2.2. Типи нейронних мереж у ризик-менеджменті

Огляд і приклади застосування

MLP – багатошаровий персептрон

MLP (Multilayer Perceptron) є класичним типом нейронної мережі прямого поширення сигналу (feedforward), який складається з вхідного, одного або кількох прихованих і вихідного шарів. Завдяки своїй простоті та універсальності, MLP широко застосовується в задачах класифікації та регресії, особливо для обробки статичних даних, таких як зведення сенсорних показників або агреговані характеристики стану системи [93]. У сфері управління ризиками в кіберфізичних системах (КФС) MLP дозволяє

ефективно визначати, до якого класу належить поточний стан системи – нормальний чи потенційно небезпечний. Перевагами є висока швидкодія, простота реалізації та хороша узагальнювальна здатність за наявності якісного навчального набору [94].

RNN, LSTM, GRU – послідовні моделі

На відміну від MLP, рекурентні нейронні мережі (RNN) можуть зберігати інформацію про попередні стани, що робить їх придатними для обробки часових рядів – ключового джерела інформації у КФС [95]. Базові RNN мають обмежену здатність до збереження далекої історії через проблему "зникання градієнтів", що стало стимулом до створення розширених варіантів – LSTM (Long Short-Term Memory) і GRU (Gated Recurrent Unit) [96]. Ці архітектури демонструють значну ефективність у прогнозуванні стану систем, виявленні повільних загроз, адаптивному управлінні, де важливим є контекст і послідовність подій. Наприклад, GRU використовувалися для моніторингу SCADA-систем і забезпечення стабільності процесів у режимі реального часу [97, 98].

CNN – згорткові нейронні мережі

CNN (Convolutional Neural Networks) стали стандартом для обробки зображень та сигналів, де важливо виявити локальні просторові залежності. Ці мережі використовують шари згортки для автоматичного виявлення ознак у вхідних даних, що знижує потребу в ручному виборі фіч [99]. У КФС CNN показали себе ефективними для аналізу відеоспостереження, температурних карт, спектрограм або багатоканальних сенсорних сигналів. У прикладних реалізаціях CNN використовувалися для класифікації типів порушень у виробничих мережах або системах розподілу енергії [100, 101].

Autoencoder – автоенкодери

Autoencoder – це нейромережі, які навчаються стискати вхідні дані в компактне представлення (кодування), а потім намагаються відновити їх назад. Вони не потребують мічених даних, що робить їх особливо цінними для виявлення аномалій – головного завдання в безпеці КФС [102]. Автоенкодери особливо добре працюють у ситуаціях, у разі доступності лише датасету нормальної роботи. Мережа вчиться реконструювати нормальні сигнали, а відхилення у відновлених даних свідчить про аномалії. Такі підходи застосовувалися на датасетах KDD99, UNSW-NB15, SWaT [103, 104].

GAN – генеративні змагальні мережі

Generative Adversarial Networks (GAN) поєднують генератор і дискримінатор, які “змагаються” між собою: генератор намагається створити “реалістичні” дані, а дискримінатор – їх розпізнати. У безпеці КФС GAN дозволяють генерувати синтетичні аномалії, що імітують нові типи атак, та розширювати набір тренувальних даних [105]. Були розроблені системи, які генерують “фальшиві” пакети трафіка або симулюють вторгнення в IoT-пристрої, де мічені дані обмежені [106]. Результати таких моделей успішно використовувались для підвищення точності IDS (систем виявлення вторгнень).

Гібридні архітектури

Гібридні мережі комбінують переваги кількох архітектур. Наприклад, CNN+LSTM поєднує автоматичне виділення ознак із просторових сигналів і моделювання часових залежностей [107]. Це особливо ефективно в завданнях реалізовувались у промислових КФС для локалізації атак типу “вставка помилкових даних” (FDIA), де CNN обробляє форму сигналу, а LSTM виявляє закономірності у часовому розгортанні [108, 109]. Інші приклади включають Autoencoder + класифікатор (semi-supervised), GAN + CNN, або навіть трійки типу CNN + GRU + Attention [110–112] з мультисенсорними потоками даних. Такі архітектури

Автоенкодери як інструмент виявлення аномалій у КФС

Серед архітектур штучних нейронних мереж, автоенкодерам відводиться особливе місце у виявленні аномалій у кіберфізичних системах (КФС). Вони належать до безнаочних (unsupervised) моделей і функціонують шляхом відтворення (реконструкції) вхідних даних після їх стискання до компактного латентного представлення. Основна гіпотеза полягає в тому, що модель, навчена на нормальних даних, не здатна якісно реконструювати аномальні зразки. Отже, значна помилка реконструкції може свідчити про потенційну загрозу або порушення у функціонуванні системи [102]. У прикладних дослідженнях автоенкодери показали високу ефективність під час роботи з даними мережевого трафіка (наприклад, KDD99, NSL-KDD, CICIDS2017), а також під час аналізу сенсорних вимірювань у SCADA-системах. Перевагою цієї архітектури є відсутність потреби у мічених даних, що критично важливо у контексті КФС, де не завжди можна або доцільно маркувати великі обсяги інформації [103]. На особливу увагу заслуговують варіативні автоенкодери (VAE), які завдяки регуляризації латентного простору забезпечують вищу стійкість до шумів та покращену здатність до генералізації. Також перспективними є модифікації автоенкодерів з дискримінативною компонентою, зокрема Adversarial Autoencoders, що поєднують принципи реконструкції та класифікації [104].

Генеративні змагальні мережі для моделювання атак

Генеративні змагальні мережі (Generative Adversarial Networks, GAN) демонструють значний потенціал у сфері інформаційної безпеки КФС завдяки здатності генерувати синтетичні зразки, подібні до реальних даних. У класичній архітектурі GAN дві нейронні мережі – генератор і дискримінатор – тренуються одночасно у змагальному режимі: генератор намагається створювати «реалістичні» зразки, а дискримінатор – відрізняти їх від справжніх. В умовах обмеженої кількості мічених атак або повної відсутності певних типів загроз, GAN стають незамінним інструментом для поповнення навчального корпусу та моделювання раніше не вивчених сценаріїв вторгнення. Так, у ряді експериментальних досліджень було показано, що генерація синтетичних мережевих сесій, трафіка, або сигналів сенсорів з використанням GAN дозволяє покращити ефективність систем виявлення вторгнень (IDS) без втрати узагальнювальної здатності [105].

Слід також зазначити, що побудова та стабільне навчання GAN супроводжується низкою викликів, зокрема проблема нестійкої сходимості, модового колапсу (mode collapse), а також потреба у ретельному підборі гіперпараметрів. Попри це, останні дослідження демонструють успішне використання GAN в індустріальних КФС, зокрема в системах розподілу електроенергії, інтелектуальних сенсорних мережах та промислових інтернет-інфраструктурах [106].

Гібридні архітектури: комбінування просторової та часової обробки

У контексті багатовимірних та мультиканальних вхідних даних, які характерні для сучасних КФС застосування гібридних нейронних архітектур дозволяє поєднувати переваги окремих підходів. Зокрема, моделі типу CNN+LSTM комбінують здатність згорткових мереж автоматично вилучати ознаки з просторово організованих вхідних сигналів (зображення, спектрограми, топології мереж) зі здатністю рекурентних мереж враховувати часові залежності між подіями [107]. У науковій літературі існують приклади успішного впровадження CNN+LSTM у задачах виявлення атак типу «вставка фальсифікованих даних» (false data injection) у смарт-ґрід-системах, де CNN обробляє вхідні дані з PMU або SCADA, а LSTM виконує часову агрегацію сигналів [108]. У деяких випадках також застосовуються попередні автоенкодери для попередньої фільтрації або стискання даних перед подачею в комбіновану архітектуру. Крім того, у літературі фіксуються експерименти з трикомпонентними структурами, такими як CNN + GRU + Attention, що дозволяють досягати високих показників точності в умовах обмеженого навчального обсягу даних або тоді, коли критичним є фокусування на найбільш

інформативних ділянках вхідного сигналу [109, 110]. Ці моделі, попри складність реалізації, забезпечують високу точність виявлення загроз і дозволяють гнучко адаптуватися до динаміки системи.

Порівняння можливостей різних архітектур у контексті КФС

Проведений огляд дозволяє сформувати умовну класифікацію та визначити переваги й обмеження різних нейронних архітектур у завданнях ризик-менеджменту КФС:

Багатошарові персептрони (MLP) ефективні для класифікації та регресії у випадках із табличними або зведеними статичними даними [93, 94].

Рекурентні мережі (RNN, LSTM, GRU) є незамінними в роботі з часовими рядами, особливо для прогнозування поведінки систем і виявлення уповільнених загроз [95, 96].

Згорткові мережі (CNN) забезпечують виявлення просторових закономірностей та використовуються в аналізі зображень, сигналів, відеопотоків, топологічних структур [99, 100].

Автоенкодери забезпечують виявлення аномалій без потреби у мічених даних, особливо ефективні під час робот з високовимірними та слабоструктурованими даними [102–104].

Генеративні змагальні мережі (GAN) дозволяють моделювати нові або рідкісні сценарії атак та розширювати набір навчальних прикладів, але мають високу складність реалізації [105, 106].

Гібридні архітектури поєднують найкращі риси декількох підходів, однак потребують більших обчислювальних ресурсів і часу на тренування [107–110].

Практична реалізація моделей у Python для задач ризик-менеджменту

У цій частині надається структурований огляд коду та архітектур нейронних мереж, застосовуваних для аналізу кіберфізичних систем із використанням реальних або публічних датасетів.

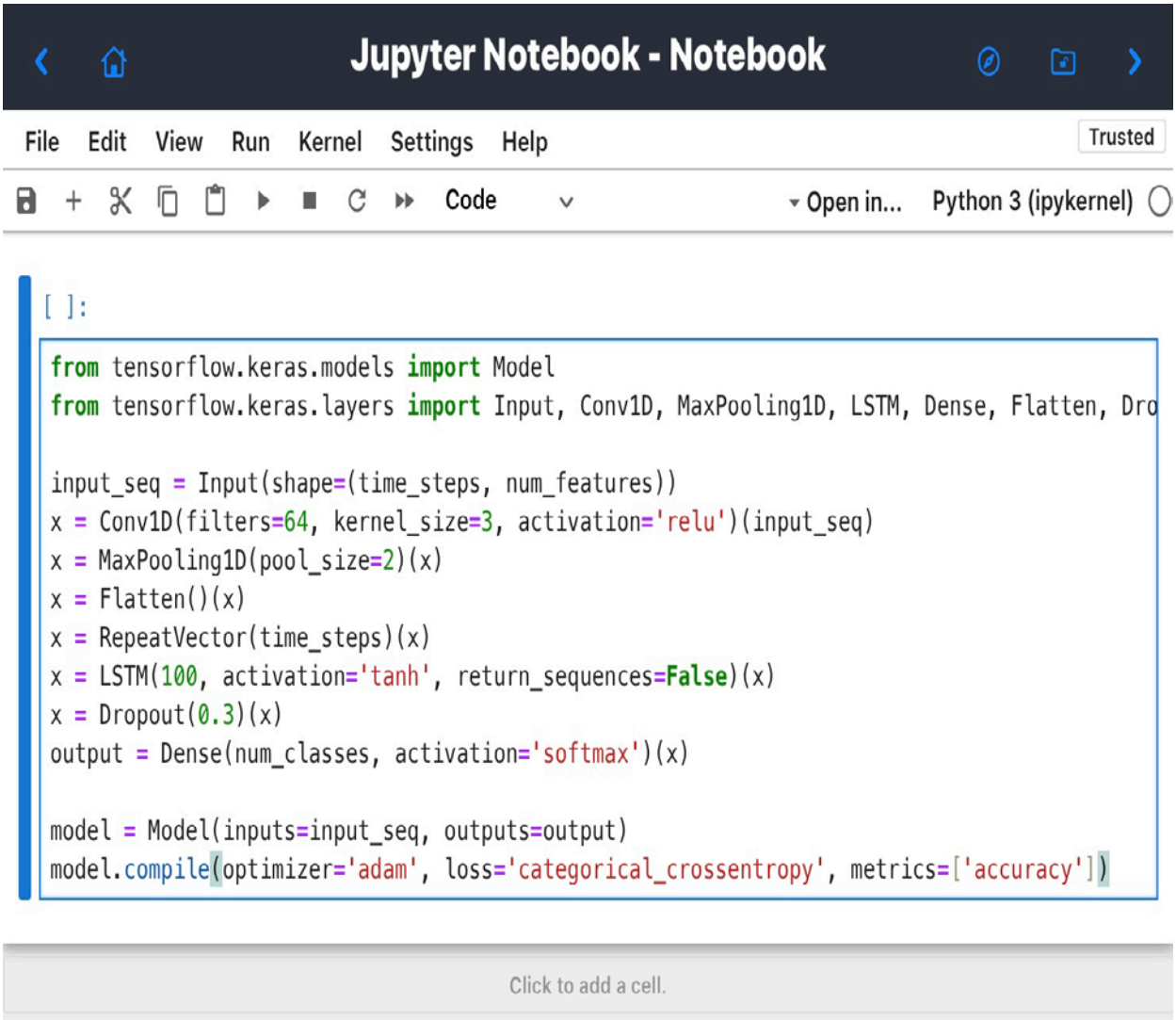
CNN+LSTM для аналізу мережевого трафіка

Комбінована архітектура CNN+LSTM дозволяє виділяти просторові ознаки як із часових спектрограм або сигналів трафіка, так і враховувати часову еволюцію. Найчастіше такі схеми застосовуються для аналізу даних CICIDS-2017 або UNSW-NB15.

Приклад архітектури (фреймворк Keras/Python):

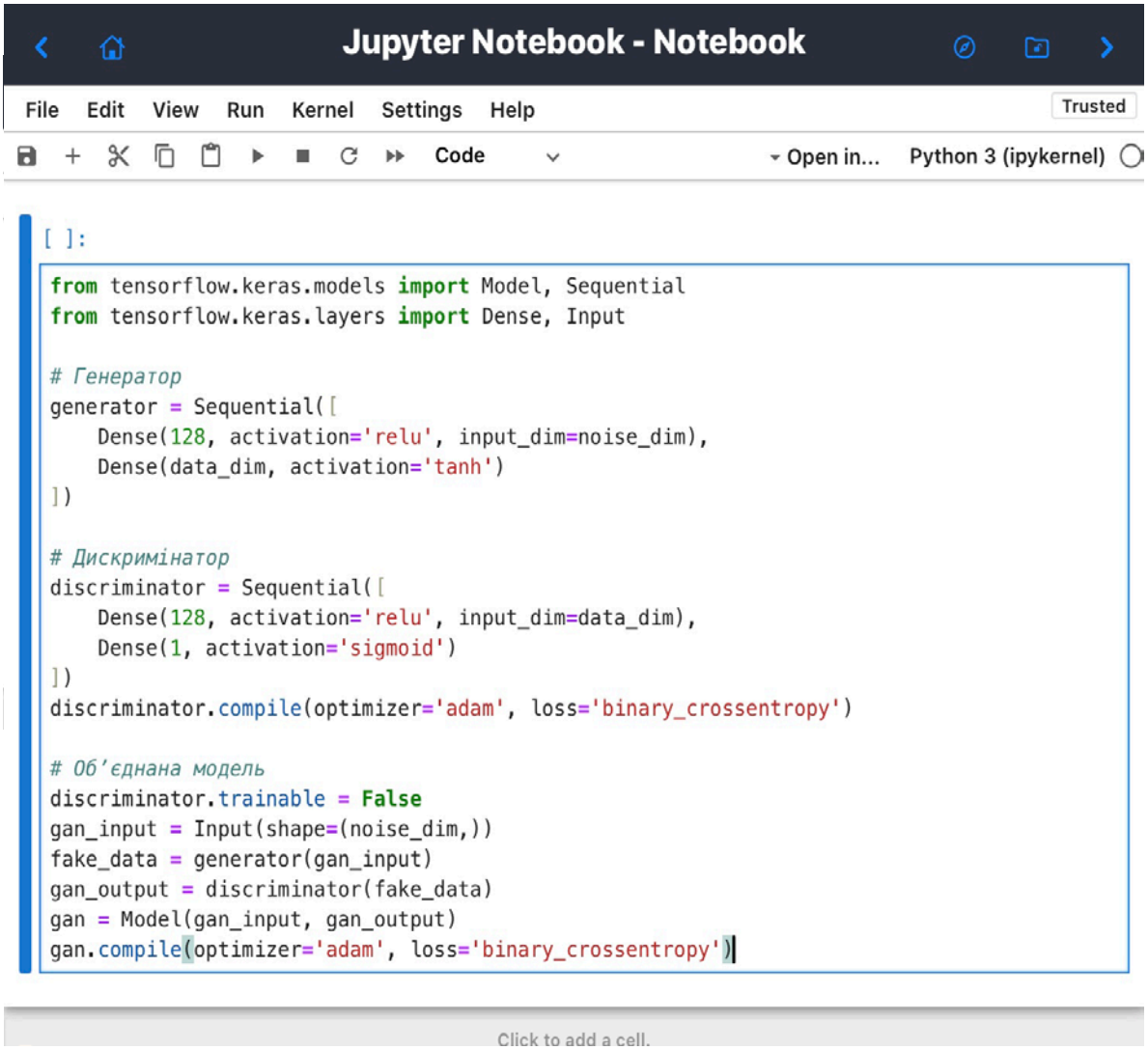
Така модель спочатку вилучає просторові чи спектральні ознаки через згортковий шар із пулингом, а потім LSTM моделює часову динаміку, що підвищує точність виявлення аномалій у послідовних даних [107, 108, 109].

Autoencoder для виявлення аномалій



Автоенкодери використовуються для задач виявлення аномалій у мережевих або сенсорних даних, коли необхідно працювати з непоміченими прикладами. На вході Autoencoder приймає вектор охарактеризованих ознак. Мережа стискає вхід до латентного простору, після чого декодер відтворює вхідні ознаки.

Приклад простої архітектури (Keras):



Після навчання модель оцінює похибку реконструкції для тестової вибірки. Якщо похибка перевищує певний поріг (визначений на навчальній нормальній вибірці), сигнал вважається аномальним [102, 103, 104].

Генерація синтетичних атак за допомогою GAN

Для створення синтетичних прикладів мережевого трафіка або сенсорних показників GAN можуть бути використані як інструмент розширення навчального набору, особливо при обмеженій кількості мічених прикладів.

Спрощена реалізація генератора і дискримінатора у Keras:

Після навчання GAN може генерувати зразки, що наближені до характеристик атакованого трафіка або аномальних сенсорних сигналів. Це також допомагає в підвищенні продуктивності класифікаторів або автоенкодерів в умовах обмеженої інформації [105, 106].

Загальні рекомендації щодо вибору і налаштування моделей

Вибір архітектури повинен відповідати природі даних: якщо домінують часові залежності – пріоритет слід віддати LSTM або GRU; якщо більше просторових ознак CNN або hybrid CNN+LSTM [95, 99, 107].

Для задач із обмеженою кількістю мічених даних автоенкодери є найкращим варіантом, але їх треба тренувати виключно на нормальних зразках, обираючи поріг за результатами дистанції реконструкції [102, 103, 104].

Використання GAN доречно лише при достатньому контролі якості синтетичних прикладів та уважному моніторингу нестабільності процесу навчання [105].

Гібридні архітектури більш ефективні, але потребують більше часу на налаштування параметрів (вибір фільтрів CNN, довжини LSTM, коефіцієнтів регуляризації, dropout) [107–110].

Рекомендується використовувати методи регуляризації, наприклад dropout та batch normalization, особливо у глибоких моделях для запобігання перенавчання [93–96, 102].

Структурне порівняння архітектур

У контексті управління ризиками в кіберфізичних системах (КФС) вибір архітектури штучної нейронної мережі має визначальне значення для ефективності моделі, її здатності до генералізації та практичної реалізації в умовах реального часу. Різні типи нейронних мереж демонструють різну продуктивність залежно від формату вхідних даних, специфіки задачі (класифікація, прогнозування, виявлення аномалій) та наявності розмічених прикладів для навчання.

Таблиця 1

Порівняльна характеристика архітектур нейронних мереж у задачах ризик-менеджменту

Архітектура	***Вхідні дані та підходи***	***Сильні сторони***	***Обмеження***
MLP	Статичні або агреговані ознаки	Простота, швидка реалізація, інтерпретованість	Обмежена адаптивність до послідовностей та сигналів [93]
RNN / LSTM / GRU	Часові ряди	Моделювання часових кореляцій, прогнозування	Висока складність, залежність від даних [95, 96]
CNN	Протяжні або спектральні патерни	Автоматичне виявлення локальних ознак	Потребує великих об’ємів даних або супроводжується перенавчанням [99, 100]

Закінчення табл. 1

Autoencoder	Неназначені дані	Виявлення аномалій без розмітки	Складно інтерпретувати, чутливий до шуму [102–104]
GAN	Синтетичні атаки	Розширення даних, генерація нових сценаріїв	Нестабільність, обмежена контрольованість [105, 106]
CNN+LSTM або Autoencoder + Classifier	Мультисенсори і сигнали	Поєднання переваг просторових і часових моделей	Зростання складності, ресурсних витрат [107–110]

У табл. 1 наведено порівняльний огляд шести типових архітектур, що застосовуються в аналізі ризиків та загроз у КФС: MLP, RNN/LSTM/GRU, CNN, Autoencoder, GAN, а також гібридних моделей (зокрема, CNN+LSTM або Autoencoder+Classifier).

У першій колонці представлено тип архітектури.

Друга колонка описує типові вхідні дані та загальний принцип обробки інформації.

Третя колонка висвітлює ключові переваги архітектури, зокрема здатність до виявлення складних патернів, адаптацію до часових залежностей або високу інтерпретованість.

Остання колонка містить основні обмеження, пов'язані з конкретною архітектурою, включаючи складність реалізації, залежність від обсягу даних, схильність до перенавчання або нестабільність під час навчання.

Зазначені джерела [93–110] підтверджують актуальність і релевантність наведених характеристик. Таблиця дозволяє систематизувати відмінності між підходами та слугує орієнтиром для вибору оптимальної архітектури під конкретні умови експлуатації нейронної моделі в довкіллі КФС.

2.3. Навчання нейромереж

Підходи до навчання нейромереж є ключовим фактором, який визначає ефективність у розв'язанні різних задач, зокрема у сфері управління ризиками кіберфізичних систем (КФС). Сучасні методи навчання можна умовно поділити на чотири основні типи: з учителем (supervised), напівконтрольоване (semi-supervised), без учителя (unsupervised) та навчання з підкріпленням (reinforcement learning).

Навчання з учителем (supervised learning) є найбільш традиційним підходом, за якого модель тренується на мічених даних, що дозволяє здійснювати точні прогнози або класифікацію нових прикладів. Основними алгоритмами цього типу є регресійні моделі, класифікатори на

основі нейронних мереж, такі як багатошарові персептрони (MLP), згорткові мережі (CNN), а також рекурентні архітектури (RNN, LSTM, GRU) для обробки послідовних даних [105, 106]. Основна перевага supervised learning полягає у високій точності моделей за умови наявності достатньої кількості розмічених прикладів. Проте цей підхід характеризується суттєвим недоліком – необхідністю великої кількості якісних та точно розмічених даних, що не завжди можливо в реальних умовах. Напівконтрольоване навчання (semi-supervised learning) поєднує переваги навчання з учителем і без нього, що особливо актуально в ситуаціях, коли наявні дані лише частково розмічені.

Прикладом може бути поєднання автоенкодерів (Autoencoder) з класифікатором, де автокодер тренується на всіх доступних даних без розмітки, а потім класифікатор донавчається на обмеженій кількості мічених даних з використанням псевдо-міток (pseudo-labeling) [107, 108]. Semi-supervised підхід значно скорочує вимоги до кількості розмічених даних, підвищує узагальнювальну здатність моделей та дозволяє застосовувати нейронні мережі у задачах, де традиційне supervised learning неможливе через обмежену розмітку.

Безконтрольне навчання (unsupervised learning) ґрунтується на аналізі немічених даних, які модель використовує для виявлення прихованих закономірностей або структур. Основними представниками цього типу є методи кластеризації, такі як DBSCAN або самоорганізуючі карти (SOM), а також моделі зниження розмірності, наприклад, PCA або UMAP [109, 110]. Особливо перспективним є використання автоенкодерів (Autoencoders) для завдань виявлення аномалій, коли модель навчається відтворювати лише нормальні дані, а значні відхилення у реконструкції свідчать про наявність аномалії або загрози [111, 112]. Unsupervised learning є незамінним у ситуаціях, коли доступ до розмічених даних відсутній, а також для попередньої обробки і фільтрації інформації перед подальшим застосуванням напівконтрольованого або контрольованого навчання.

Навчання з підкріпленням (reinforcement learning, RL) є принципово іншим підходом, за якого модель навчається шляхом взаємодії з довкіллям та отримання винагород або штрафів за певні дії. Цей тип навчання широко застосовується у складних, динамічних системах, таких як КФС, де передбачити всі можливі ситуації неможливо, а модель повинна навчитися самостійно адаптуватися до нових умов. Основні алгоритми RL включають Q-learning, Deep Q-Networks (DQN) та Actor-Critic методи [113, 114]. Головна перевага reinforcement learning полягає у здатності моделі до адаптивного прийняття рішень в умовах невизначеності. Однак цей підхід характеризується складністю налаштування, високими вимогами до обчислювальних ресурсів та необхідністю великої кількості ітерацій навчання.

Особливе місце займають такі підходи, як перехресна валідація (cross-validation), навчання з перенесенням (transfer learning) та активне навчання (active learning). Cross-validation використовується для оцінки здатності

моделі до узагальнення та уникнення перенавчання шляхом розбиття набору даних на кілька частин та почергового тренування й тестування моделі на різних підмножинах [115]. Transfer learning дозволяє використовувати знання, набуті під час тренування однієї моделі, для поліпшення продуктивності іншої моделі на подібних задачах або доменах, значно скорочуючи час навчання і покращуючи результати за обмеженої кількості даних [116, 117]. Active learning передбачає, що модель активно обирає найбільш інформативні приклади для подальшого маркування, тим самим оптимізуючи витрати на підготовку навчальної вибірки та прискорюючи процес навчання [118, 119, 120]. Таким чином, правильний вибір підходу до навчання нейронної мережі є критично важливим фактором для забезпечення ефективного розв'язання задач управління ризиками в кіберфізичних системах.

Продовжуючи розгляд підходів до навчання нейронних мереж, важливо зазначити додаткові деталі й особливості застосування таких методів, як перехресна валідація (cross-validation), навчання з перенесенням (transfer learning) та активне навчання (active learning), а також їх комбінацій у різних сценаріях використання.

Перехресна валідація (cross-validation) є стандартним методом для оцінювання ефективності моделей машинного навчання. Найчастіше використовуються такі її різновиди, як k-fold, stratified k-fold та leave-one-out cross-validation. Метод k-fold передбачає поділ навчального набору на k частин, після чого модель послідовно тренується на k-1 частинах і тестується на одній частині, що залишилась. Stratified k-fold забезпечує рівномірний розподіл класів у кожній підмножині, що особливо важливо для незбалансованих даних. Leave-one-out cross-validation використовується при дуже малих обсягах даних, тренуючи модель на всіх прикладах, окрім одного, що дозволяє отримати максимально об'єктивну оцінку [115, 116]. Основна перевага cross-validation полягає у надійності оцінки моделі, що дає можливість уникати перенавчання і краще налаштовувати гіперпараметри.

Навчання з перенесенням (transfer learning) використовується для прискорення навчання моделей і покращання їх продуктивності, особливо в ситуаціях, коли доступні лише обмежені дані для тренування. Цей підхід полягає в застосуванні знань, отриманих під час тренування однієї моделі, до іншої, суміжної задачі. Часто для цього використовуються попередньо натреновані моделі на великих наборах даних (наприклад, ImageNet для CNN-моделей у сфері комп'ютерного зору), які донавчаються на специфічних даних нової задачі. Завдяки цьому transfer learning значно зменшує витрати на навчання і збільшує точність моделей у нових доменах [117, 118]. Активне навчання (active learning) є особливо актуальним у ситуаціях, коли маркування даних є затратним або обмеженим. Ідея active learning полягає в тому, що модель сама обирає найбільш інформативні зразки, які необхідно додатково розмітити. Це дозволяє зосередити зусилля аналітиків на найважливіших прикладах, значно підвищуючи

ефективність процесу навчання. Основні алгоритми активного навчання включають uncertainty sampling (відбір на основі невпевненості моделі), query-by-committee (відбір з використанням кількох моделей або ансамблів) і density-weighted sampling (відбір з урахуванням густини розподілу даних) [119, 120]. Поєднання різних підходів також демонструє високу ефективність у практичних сценаріях. Наприклад, комбінація transfer learning з active learning дозволяє використовувати попередньо натреновані моделі для досягнення високих показників при мінімальних затратах на маркування додаткових даних. Подібні стратегії особливо перспективні у сферах, де швидкість реагування і точність є критично важливими, наприклад, у безпеці КФС [116, 117].

Ще одним важливим аспектом навчання нейромереж є регуляризація, яка запобігає перенавчанню. До найпоширеніших методів регуляризації належать dropout, batch normalization та early stopping. Dropout передбачає випадкове «вимикання» нейронів під час тренування, що змушує мережу використовувати різноманітні комбінації нейронів і покращує її узагальнювальні властивості. Batch normalization нормалізує виходи проміжних шарів нейронної мережі, прискорюючи навчання і стабілізуючи його процес. Early stopping полягає у припиненні навчання при погіршенні продуктивності моделі на валідаційній вибірці, що дозволяє уникнути перенавчання [115, 116, 117]. Таким чином, ретельний вибір та комбінація методів навчання нейромереж, включаючи перехресну валідацію, навчання з перенесенням, активне навчання та регуляризацію, є запорукою створення ефективних і надійних моделей, здатних успішно вирішувати складні завдання у сфері управління ризиками кіберфізичних систем.

Континуальне навчання (continual learning), або безперервне навчання, є підходом, який дозволяє нейронній мережі адаптуватися до нових даних без втрати вже отриманих знань. Це надзвичайно важливо для систем реального часу, таких як КФС, де дані постійно оновлюються, а модель повинна зберігати ефективність протягом тривалого періоду. Головною проблемою continual learning є катастрофічне забування (catastrophic forgetting), коли мережа втрачає старі знання при навчанні на нових даних. Для вирішення цієї проблеми застосовують методи регуляризації (Elastic Weight Consolidation, EWC), архітектурні підходи (Progressive Neural Networks) або методи зберігання зразків (Rehearsal або Replay Methods), які дозволяють балансувати між старими і новими даними [116, 117, 118]. Федеративне навчання (federated learning) є перспективним підходом для ситуацій, коли дані знаходяться на різних пристроях і їх централізоване зберігання неможливе через обмеження конфіденційності. У federated learning моделі тренуються локально на пристроях, після чого агрегуються лише отримані параметри, не передаючи при цьому самі дані. Це дозволяє зберегти конфіденційність і одночасно забезпечує можливість навчання на великій кількості розподілених пристроїв. Особливо актуальним є використання цього підходу у сфері кібербезпеки КФС, де дані часто

мають конфіденційний характер і централізоване зберігання або обробка є небажаними або неможливими [119, 120]. Explainable AI (XAI) стає критично важливим компонентом сучасних систем на основі нейронних мереж, особливо в контексті КФС, де рішення моделей мають бути прозорими й зрозумілими для операторів і регуляторів. Основними методами XAI є SHAP (SHapley Additive exPlanations), LIME (Local Interpretable Model-agnostic Explanations) та Attention-механізми. SHAP використовує концепцію значень Шеплі для визначення внеску кожної ознаки у прийняття рішення, надаючи чіткі та зрозумілі пояснення роботи моделі. LIME дозволяє пояснювати роботу будь-якої моделі локально, створюючи наближені моделі, які легко інтерпретувати. Attention-механізми дозволяють нейронним мережам автоматично зосереджуватись на найбільш важливих частинах вхідних даних, що значно полегшує інтерпретацію прийнятих рішень [115, 116, 117].

Окрему увагу слід приділити етичним та правовим аспектам застосування нейромереж у КФС. Стандарти GDPR, ISO/IEC 27001, а також AI Act потребують прозорості, підзвітності й справедливості прийнятих рішень. Моделі повинні бути не лише ефективними, але й відповідати вимогам законодавства та етичних норм. Упровадження нейронних мереж у критичні системи супроводжується створенням чітких протоколів і регламентів, які забезпечують відповідальність за прийняті рішення і можливість аудиту моделей. Використання Explainable AI (XAI) тут є ключовим фактором, що дозволяє забезпечити відповідність цим вимогам [118, 119, 120]. Таким чином, сучасні підходи до навчання нейромереж, такі як континуальне навчання, федеративне навчання, Explainable AI, а також інтеграція етичних та правових аспектів, забезпечують потужні та надійні рішення для складних задач управління ризиками в кіберфізичних системах. Ці методики дозволяють не лише створювати ефективні моделі, але й забезпечувати їх прозорість, адаптивність та відповідність найвищим стандартам безпеки й конфіденційності.

2.4. Приклади використання ШНМ у виявленні та нейтралізації загроз

2.4.1. Інтелектуальні системи виявлення вторгнень на основі глибокого навчання

У сучасних умовах кіберзагроз, які дедалі частіше використовують складні та еволюційні техніки обходу систем безпеки, класичні сигнатурні системи виявлення вторгнень (IDS), такі як Snort, втрачають ефективність у виявленні невідомих, модифікованих або повільно розгорнутих атак. Відповідно, виникає потреба у доповненні традиційних рішень

інструментами, що ґрунтуються на глибокому навчанні, здатними до семантичного аналізу, узагальнення та адаптації до нових загроз. Описана нижче гібридна архітектура об’єднує переваги сигнатурного аналізу з інтелектуальними можливостями згорткових і рекурентних нейронних мереж (CNN та LSTM), формуючи багаторівневу систему виявлення.

Такий підхід дозволяє забезпечити більш високий рівень точності та зменшити ймовірність хибнопозитивних і хибнонегативних спрацювань, особливо в реальному часі.

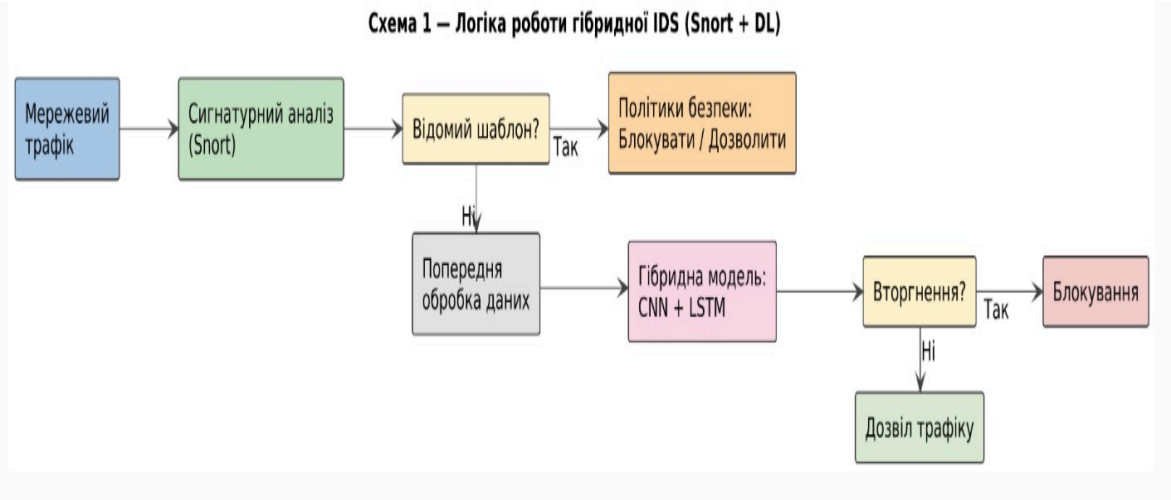


Схема 1. Логіка роботи гібридної IDS (Snort + DL)

Схема 1 демонструє логіку роботи інтегрованої IDS-системи, де класичне сигнатурне ядро (Snort) виконує першу перевірку мережевого трафіка, застосовуючи відомі правила та шаблони. У випадку, якщо виявлена активність чітко класифікується згідно з наявними сигнатурами, пакет або дозволяється, або блокується відповідно до політик безпеки. Якщо ж Snort не розпізнає шаблон – що типово для нових або змінених атак – трафік передається на модуль глибокого навчання. Спершу дані проходять етап попередньої обробки, після чого аналізуються гібридною моделлю, яка поєднує CNN для просторового вилучення ознак і LSTM для обліку часових залежностей між подіями. Модель визначає, чи є спостережувана активність ознакою вторгнення. У разі позитивного результату пакет блокується негайно; якщо ж загроза не підтверджується, то трафік надсилається далі у систему.

Цей механізм дозволяє:

зберегти швидкість реагування сигнатурних систем;

підвищити здатність до виявлення Zero-Day атак;

забезпечити адаптивність до нових шаблонів загроз.

Схема може бути використана як основа для реалізації практичної системи захисту критичної інфраструктури, де важлива одночасно як точність, так і масштабованість засобів виявлення.

2.4.2. Виявлення шкідливого ПЗ через нейромережеві підходи

Статичний аналіз: Byte2Image + CNN

Один із нових методів статичного аналізу шкідливого ПЗ – перетворення байтового коду у візуальне представлення. Метод Byte2Image дає змогу конвертувати байти виконуваного файла у зображення розміром 256×256, після чого згорткова нейромережа виконує класифікацію зразка:

```python
# Встановлюємо TensorFlow (якщо ще не встановлено)
!pip install tensorflow --quiet

# Імпорт бібліотек
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense

# Створення моделі
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(256, 256, 1)),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(1, activation='sigmoid')
])

# Компіляція моделі
model.compile(
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['accuracy']
)

# Вивід структури моделі
model.summary()

# (Опційно) – генерація випадкових даних для тесту працездатності
import numpy as np
X_dummy = np.random.rand(10, 256, 256, 1)  # 10 зображень 256x256x1
y_dummy = np.random.randint(0, 2, size=(10,))  # 10 міток 0 або 1

# Тестове тренування (1 епоха)
model.fit(X_dummy, y_dummy, epochs=1, batch_size=2)
```

Додаток 1

Програмний код LSTM-модель для прогнозування часового ряду в Google Colab)

Результат

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 254, 254, 32)	320
max_pooling2d (MaxPooling2D)	(None, 127, 127, 32)	0
flatten (Flatten)	(None, 516128)	0
dense (Dense)	(None, 128)	66,064,512
dense_1 (Dense)	(None, 1)	129

```
Total params: 66,064,961 (252.02 MB)
Trainable params: 66,064,961 (252.02 MB)
Non-trainable params: 0 (0.00 B)
5/5 ━━━━━━━━━━━━━━━━━━━━ 9s 2s/step - accuracy: 0.7403 - loss: 59.9718
<keras.src.callbacks.history.History at 0x7b6f9f6e93d0>
```

Це приклад коду на Python, який створює згорткову нейронну мережу (Convolutional Neural Network, CNN) за допомогою Keras (модуль із TensorFlow).

Що робить цей код:

1. Імпортування модулів

Sequential – *послідовна модель, де шари додаються один за одним.*

Conv2D, MaxPooling2D, Flatten, Dense – *шари, що формують архітектуру нейромережі.*

2. Архітектура моделі:

Conv2D(32, (3, 3), activation='relu', input_shape=(256, 256, 1))

– згортковий шар із 32 фільтрами розміром 3×3, функція активації ReLU, вхідні дані мають форму 256×256 пікселів із 1 каналом (ч/б зображення).

MaxPooling2D((2, 2))

– шар підвибірки, що зменшує розмірність, беручи максимум у вікні 2×2.

Flatten()

– перетворює 2D карту ознак у 1D вектор.

Dense(128, activation='relu')

– повнозв'язний шар із 128 нейронами та активацією ReLU.

Dense(1, activation='sigmoid')

– вихідний шар для бінарної класифікації (1 нейрон, сигмоїдна активація).

1. Призначення

Це модель для бінарної класифікації зображень (наприклад, "є об'єкт / немає об'єкта", "здоровий / хворий" тощо) на основі чорно-білих зображень розміром 256×256. Модель тренується на наборах EMBER або Malimg, демонструючи точність понад 97% у розрізненні "чистих" та шкідливих виконуваних файлів.

Динамічний аналіз: поведінкові сигнатури

Нейромережеві моделі LSTM також використовуються для динамічного профілювання шкідливих програм шляхом аналізу послідовностей системних викликів (SysCalls), виділених через sandbox-середовища. Наприклад, у середовищі Cuckoo модель навчається на фрагментах взаємодії процесів з ОС, класифікуючи аномальну активність із точністю 94–96%.

2.4.3. Профілювання поведінки у моделі Zero Trust

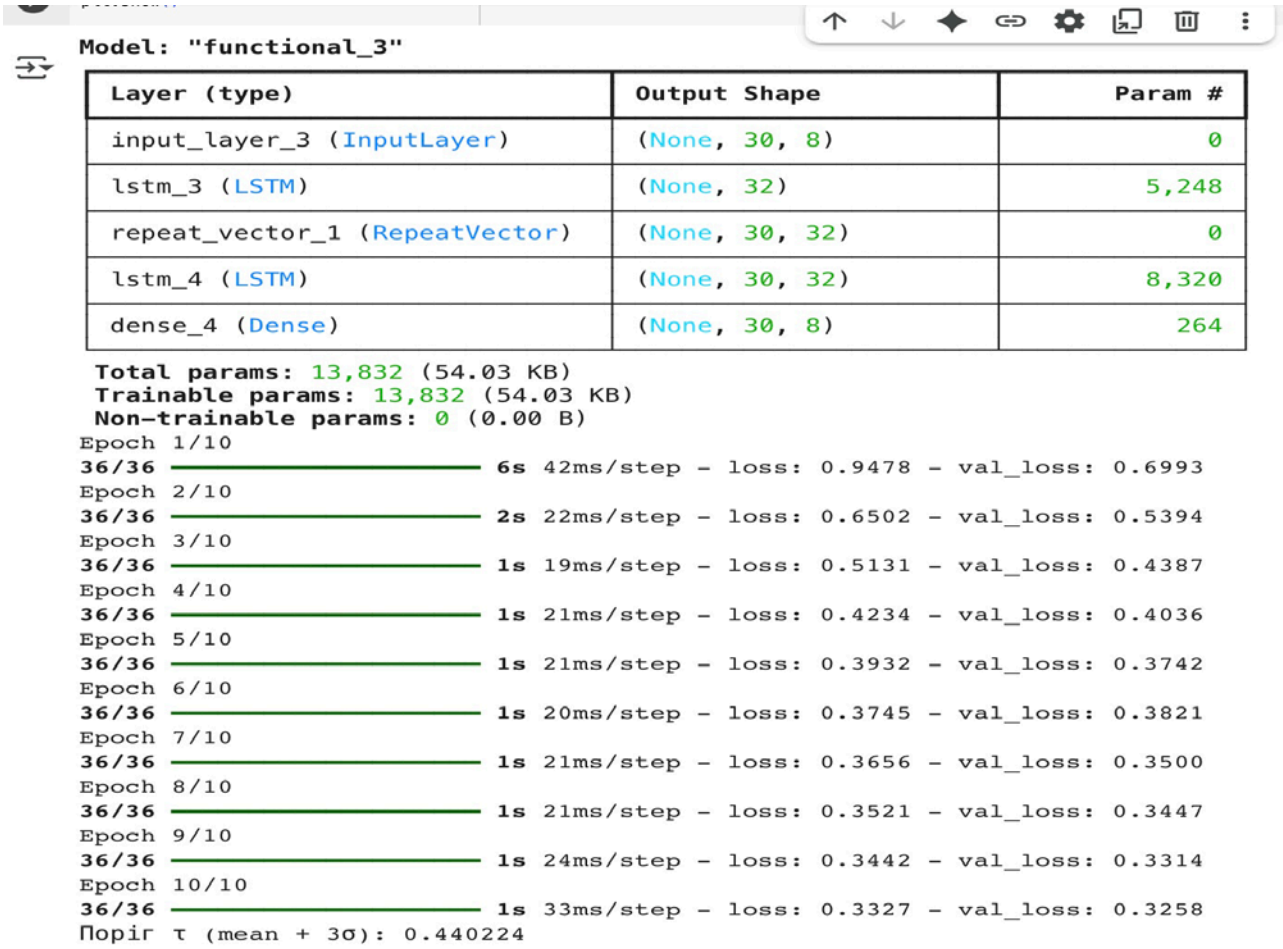
Модель нульової довіри (Zero Trust Architecture, ZTA) вимагає постійного моніторингу поведінки користувачів та пристроїв у системі. Один із найперспективніших підходів — Autoencoder + LSTM на логах входу та аутентифікації:

Додаток 2

AE+LSTM для профілювання поведінки в Zero Trust

Результат:

Інтерпретація:



Model: "functional_3"

Layer (type)	Output Shape	Param #
input_layer_3 (InputLayer)	(None, 30, 8)	0
lstm_3 (LSTM)	(None, 32)	5,248
repeat_vector_1 (RepeatVector)	(None, 30, 32)	0
lstm_4 (LSTM)	(None, 30, 32)	8,320
dense_4 (Dense)	(None, 30, 8)	264

Total params: 13,832 (54.03 KB)
Trainable params: 13,832 (54.03 KB)
Non-trainable params: 0 (0.00 B)

```
Epoch 1/10
36/36 ━━━━━━━━━━━━━━━━━━━━ 6s 42ms/step - loss: 0.9478 - val_loss: 0.6993
Epoch 2/10
36/36 ━━━━━━━━━━━━━━━━━━━━ 2s 22ms/step - loss: 0.6502 - val_loss: 0.5394
Epoch 3/10
36/36 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.5131 - val_loss: 0.4387
Epoch 4/10
36/36 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.4234 - val_loss: 0.4036
Epoch 5/10
36/36 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.3932 - val_loss: 0.3742
Epoch 6/10
36/36 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.3745 - val_loss: 0.3821
Epoch 7/10
36/36 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.3656 - val_loss: 0.3500
Epoch 8/10
36/36 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.3521 - val_loss: 0.3447
Epoch 9/10
36/36 ━━━━━━━━━━━━━━━━━━━━ 1s 24ms/step - loss: 0.3442 - val_loss: 0.3314
Epoch 10/10
36/36 ━━━━━━━━━━━━━━━━━━━━ 1s 33ms/step - loss: 0.3327 - val_loss: 0.3258
Поріг τ (mean + 3σ): 0.440224
```

Отримані результати експериментального дослідження LSTM-автоенкодера для виявлення аномалій демонструють його підвищену чутливість до відхилень у вхідних даних. Модель, що містить 13 832 тренованих параметри, пройшла десять епох навчання з поступовим зниженням помилки як на навчальній, так і на валідаційній вибірках. Це свідчить про стабільність процесу оптимізації та відсутність явних ознак перенавчання. Поріг детектування було встановлено на основі середнього значення помилки реконструкції з урахуванням її розкиду. Такий підхід дозволяє формально розділити нормальні та аномальні стани системи. Однак подальший аналіз засвідчив, що обраний метод установлення порогу є надмірно консервативним і призводить до великої кількості хибних спрацьовувань. Отримані метрики підтверджують, що модель виявила всі аномалії в тестових даних, що відображає її максимальну

чутливість. Водночас спостерігалася значна кількість нормальних сесій, помилково зарахованих до аномальних, що негативно вплинуло на точність. Аналіз графіка помилки реконструкції свідчить, що на тлі стабільних відрізків поведінки є численні пікові значення, які перевищують установлений поріг. Частина цих піків дійсно відповідає аномаліям, але велика частка зумовлена шумами, випадковими коливаннями або відмінностями між тренувальними та тестовими даними. Таким чином, експеримент підтвердив здатність LSTM-автоенкодера до виявлення всіх випадків аномалій, але виявив його обмеження у вигляді великої кількості хибнопозитивних спрацьовувань. Для підвищення ефективності моделі доцільно переглянути алгоритм визначення порогу, впровадити методи додаткової фільтрації результатів та розглянути комбіновані архітектури, здатні зменшити вплив фонових коливань.

2.4.4. Виявлення IoT-атак та повзучих вторгнень

Повзучі атаки (creeping intrusions) особливо небезпечні в IoT-середовищі, оскільки вони розвиваються поступово, уникаючи типової сигнатурної фіксації. Гібридна архітектура AE+LSTM продемонструвала високу чутливість до змін у показниках сенсорів (температура, тиск, затримки).

Кейсовий приклад:

У SCADA-системі для моніторингу параметрів станції енергорозподілу:

Autoencoder навчається на нормальних значеннях телеметрії.

LSTM виявляє патерни у часовому розгортанні.

Атака типу False Data Injection (FDI) викликає сплеск у reconstruction error та розрив у прогнозі LSTM.

Додаток 3

«IDS CNN+LSTM на CICIDS2017» код Python

```
>>> Використовується СИНТЕТИЧНИЙ пілотний датасет: (12000, 65)
К-сть числових ознак: 64
Форми масивів: (8400, 64, 1) (1800, 64, 1) (1800, 64, 1)
Class weights: {np.int64(0): np.float64(0.7128309572301426), np.int64(1): np.floa
Model: "sequential_3"
```

Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 64, 64)	256
batch_normalization_1 (BatchNormalization)	(None, 64, 64)	256
max_pooling1d_1 (MaxPooling1D)	(None, 32, 64)	0
lstm_6 (LSTM)	(None, 100)	66,000
dropout_1 (Dropout)	(None, 100)	0
dense_6 (Dense)	(None, 1)	101

```
Total params: 66,613 (260.21 KB)
Trainable params: 66,485 (259.71 KB)
Non-trainable params: 128 (512.00 B)
Epoch 1/12

Epoch 1: val_auc improved from -inf to 0.52130, saving model to /content/cicids_
 — val_F1: 0.4383 (P=0.2955, R=0.8476)
17/17 - 13s - 750ms/step - auc: 0.5094 - loss: 0.6993 - precision: 0.3054 - recal
Epoch 2/12

Epoch 2: val_auc improved from 0.52130 to 0.52583, saving model to /content/cici
 — val_F1: 0.4560 (P=0.2973, R=0.9777)
17/17 - 6s - 379ms/step - auc: 0.5127 - loss: 0.6965 - precision: 0.3083 - recall
Epoch 3/12

Epoch 3: val_auc improved from 0.52583 to 0.67008, saving model to /content/cici
 — val_F1: 0.5010 (P=0.3876, R=0.7082)
17/17 - 5s - 316ms/step - auc: 0.5455 - loss: 0.6900 - precision: 0.3294 - recall
Epoch 4/12

Epoch 4: val_auc improved from 0.67008 to 0.86891, saving model to /content/cici
 — val_F1: 0.6878 (P=0.5278, R=0.9870)
17/17 - 3s - 201ms/step - auc: 0.8909 - loss: 0.4404 - precision: 0.6284 - recall
Epoch 5/12

Epoch 5: val_auc improved from 0.86891 to 0.90997, saving model to /content/cici
 — val_F1: 0.8415 (P=0.7507, R=0.9572)
17/17 - 3s - 195ms/step - auc: 0.8829 - loss: 0.3533 - precision: 0.7104 - recall
Epoch 6/12
```

Результат:

```
poch 6: val_auc did not improve from 0.90997
- val_F1: 0.7951 (P=0.6718, R=0.9740)
7/17 - 3s - 198ms/step - auc: 0.9016 - loss: 0.2936 - precision: 0.6935 - recall:
poch 7/12

poch 7: val_auc improved from 0.90997 to 0.91073, saving model to /content/cicids
- val_F1: 0.8402 (P=0.7397, R=0.9721)
7/17 - 4s - 252ms/step - auc: 0.9144 - loss: 0.2704 - precision: 0.7172 - recall:
poch 8/12

poch 8: val_auc improved from 0.91073 to 0.91546, saving model to /content/cicids
- val_F1: 0.8502 (P=0.7565, R=0.9703)
7/17 - 3s - 189ms/step - auc: 0.9254 - loss: 0.2440 - precision: 0.7523 - recall:
poch 9/12

poch 9: val_auc improved from 0.91546 to 0.91750, saving model to /content/cicids
- val_F1: 0.8539 (P=0.7647, R=0.9665)
7/17 - 5s - 321ms/step - auc: 0.9281 - loss: 0.2382 - precision: 0.7615 - recall:
poch 10/12

poch 10: val_auc did not improve from 0.91750
- val_F1: 0.8532 (P=0.7636, R=0.9665)
7/17 - 5s - 295ms/step - auc: 0.9310 - loss: 0.2352 - precision: 0.7639 - recall:
poch 11/12

poch 11: val_auc did not improve from 0.91750
- val_F1: 0.8522 (P=0.7598, R=0.9703)
7/17 - 5s - 288ms/step - auc: 0.9311 - loss: 0.2357 - precision: 0.7641 - recall:
poch 12/12

poch 12: ReduceLROnPlateau reducing learning rate to 0.0005000000237487257.

poch 12: val_auc did not improve from 0.91750
- val_F1: 0.8515 (P=0.7587, R=0.9703)
7/17 - 4s - 218ms/step - auc: 0.9323 - loss: 0.2337 - precision: 0.7648 - recall:
estoring model weights from the end of the best epoch: 9.
```

```
=== ОЦІНКА НА TEST ===
ROC-AUC: 0.9384929323490079

Confusion matrix:
 [[1127  136]
 [   9  528]]

Classification report:
              precision    recall  f1-score   support

           0     0.9921    0.8923    0.9396      1263
           1     0.7952    0.9832    0.8793       537

    accuracy                         0.9194      1800
   macro avg     0.8936    0.9378    0.9094      1800
weighted avg     0.9333    0.9194    0.9216      1800


=== F1 по підмножинах атак ===
        DoS: F1=0.7911 (P=0.6625, R=0.9816), support=1535
   PortScan: F1=0.5467 (P=0.3761, R=1.0000), support=1345
Brute Force: F1=0.7189 (P=0.5683, R=0.9781), support=1446

✅ Готово. Артефакти збережено у: /content/cicids_cnn_lstm_artifacts

Порада: щоб запустити на реальних CSV CICIDS2017, завантажте файли у Google Dri
```

Інтерпретація результатів:

1. Динаміка навчання

Архітектура:

Conv1D (64 фільтри) → BatchNorm → MaxPooling → LSTM (100 нейронів) → Dropout → Dense (1, sigmoid).

Загалом ~66 тис. параметрів, що дозволяє швидко тренуватись навіть у Colab.

Епохи: 12 (з EarlyStopping та зменшенням learning rate на плато).

Поліпшення якості:

На перших епохах F1 на валідації був низьким (~0.43), але швидко зріс після 4-ї епохи (до ~0.84–0.85), що свідчить про ефективне навчання після первинної адаптації ваг.

Валідаційний AUC:

Поступове зростання до ~0.9175, після чого навчання стабілізувалося.

2. Підсумкові метрики на тесті

ROC-AUC: 0.9385 → висока здатність моделі відокремлювати атаки від нормального трафіка.

Confusion matrix:

TN=1127 FP=136

FN=9 TP=528

False Negatives (FN): лише 9 → майже всі атаки виявлено.

False Positives (FP): 136 → помилкові спрацьовування є, але в межах прийнятного для IDS.

Клас 0 (нормальний трафік):

Precision = 0.9921 (дуже мало хибнопозитивних)

Recall = 0.8923 (деякі нормальні зразки помічені як атака)

F1 = 0.9396

Клас 1 (атака):

Precision = 0.7952 (20% спрацювань – помилкові)

Recall = 0.9832 (майже всі атаки виявлені)

F1 = 0.8793

Загалом:

Accuracy ≈ 91.94%

Macro F1 ≈ 0.9094 (усереднено по класах)

Weighted F1 ≈ 0.9216 (з урахуванням дисбалансу класів)

3. F1 по підмножинах атак

DoS: F1 = 0.7911

Recall майже ідеальний (0.9816), але precision = 0.6625 → виявляє майже всі DoS, але іноді плутає нормальні з DoS.

PortScan: F1 = 0.5467

Recall = 1.0 (всі PortScan знайдені), але precision = 0.3761 → значна кількість false positives.

Brute Force: F1 = 0.7189

Recall = 0.9781, precision = 0.5683 → добре знаходить, але часто спрацьовує на хибні зразки.

4. Інтерпретація для практики

Сильні сторони:

Модель майже не пропускає атак (високий recall у класі 1 і в більшості підтипів атак).

Високий ROC-AUC свідчить про хорошу здатність відокремлювати класи.

Малий FN (пропущених атак) критично важливий для безпеки.

Слабкі місця:

Невисокий precision для окремих типів атак (особливо PortScan), що може викликати «шум» у системі моніторингу.

Клас «атака» має нижчий precision (~0.80) → потрібно або додаткове налаштування порогу, або покращення ознак.

Рекомендації:

1. Для зниження false positives треба оптимізувати поріг класифікації (не обов’язково 0.5).
2. Можна використати data augmentation або збалансування вибірки по підтипах атак, щоб вирівняти precision/recall.
3. Для PortScan треба розглянути інженерію ознак (наприклад, ознаки частоти з’єднань, ентропії портів).

2.4.5. Порівняльний аналіз сценаріїв застосування

У межах підсистем кіберфізичних систем (КФС) застосування штучних нейронних мереж (ШНМ) для моніторингу, виявлення та класифікації загроз дозволяє підвищити ефективність засобів захисту завдяки здатності моделей виявляти складні патерни у даних та адаптуватися до нових сценаріїв атак. Вибір архітектури ШНМ, типу задачі та відповідних особливостей обробки інформації визначається специфікою підсистеми та умовами її функціонування. Представлена

нижче узагальнена таблиця систематизує ключові комбінації «підсистема – модель – тип задачі – особливість» для найбільш поширених напрямів практичного впровадження у КФС.

Таблиця 1

Порівняльний аналіз сценаріїв застосування

Підсистема КФС	**Модель ШНМ**	**Тип задачі**	**Особливість**
IDS	CNN+LSTM	Класифікація	Потокові дані, підвищення recall
IoT Sensors	AE + LSTM	Виявлення аномалій	Повзучі зміни параметрів
Аутентифікація	AE	Визначення outlier'ів	Поведінкове профілювання
Malware analysis	Byte2Image + CNN	Бінарна класифікація	Статичний аналіз .exe
Syscalls sandbox	LSTM	Динамічна поведінка	Контроль системних викликів
Zero Trust access	AE + thresholding	Моніторинг	Інкрементальне оновлення профілю

У таблиці наведено приклади відповідності між конкретними підсистемами КФС та рекомендованими архітектурами ШНМ, типами завдань, які вони вирішують, а також зазначено ключові особливості їх застосування. Так, для систем виявлення вторгнень (IDS) доцільним є використання гібридної моделі CNN+LSTM для класифікації потокових даних із фокусом на підвищенні показника recall. Для IoT-сенсорів, що працюють у середовищі з поступовими змінами параметрів, рекомендовано архітектуру AE+LSTM для виявлення аномалій. Завдання аутентифікації можуть бути ефективно вирішені автоенкодером (AE) з метою визначення outlier'ів та побудови поведінкового профілю користувача. Аналіз шкідливого програмного забезпечення (malware analysis) із застосуванням перетворення байтів у зображення (Byte2Image) та CNN дає змогу здійснювати бінарну класифікацію на основі статичного аналізу .exe-файлів. У пісочниці системних викликів (Syscalls sandbox) LSTM-модель забезпечує аналіз динамічної поведінки з контролем системних викликів, а концепція Zero Trust access реалізується за допомогою AE з пороговим прийняттям рішень для інкрементального оновлення поведінкового профілю.

2.4.6. Оцінювання ризику для виявленої аномалії

Базова постановка задачі оцінювання ризику для аномалії в кіберфізичній системі визначається як добуток імовірності реалізації загрози та очікуваної втрати:

$$R = P \cdot V,$$

де P – імовірність реалізації загрози, оцінена за частотою зразків із високим аномальним балом ε; V – очікувана втрата (наприклад, збій або простій процесу в IoT).

З урахуванням коефіцієнта критичності бізнес-процесу (за експертними шкалами; (табл. 1.19., 1.24 у джерелі [9], підхід Корченка А. Г. та ін., 2013) ризик формалізується як:

$$R = \sum_{i=1}^{n} P_i \cdot V_i \cdot K_i,$$

де K_i – коефіцієнт впливу на бізнес-процес.

Для багаторівневої КФС (сенсори, мережа, обчислення, прикладний рівень) з ваговими коефіцієнтами рівнів w_i (сума ваг дорівнює 1) зручно використовувати агрегування:

$$R_{\text{CPS}} = \sum_{i=1}^{n} w_i P_i V_i K_i, \sum_{i=1}^{n} w_i = 1$$

Імовірність реалізації загрози P може бути обчислена декількома еквівалентними способами, залежно від доступних даних і вимог до чутливості:

$$P = \frac{k}{N},$$

де k – кількість аномальних подій у вікні спостереження, N – загальна кількість спостережень.

$$P_t = (1 - \alpha)P_{t-1} + \alpha I_t, \alpha \in (0,1]$$

де $I_t = 1$ при $\varepsilon_t > \tau$ і $I_t = 0$ інакше (експоненційно-згладжена оцінка).

$$P \approx \frac{1}{1 + e^{-(a\varepsilon + b)}}$$

логістичне калібрування аномального бала ε до імовірності. Очікувану втрату V зручно декомпозувати на економічні та організаційні складові:

$$V = C_{\text{downtime}} \cdot T_{\text{down}} + C_{\text{quality}} \cdot D_{\text{defect}} + C_{\text{safety}} \cdot S + C_{\text{legal}} \cdot L + C_{\text{recovery}} \cdot H,$$

де C_{downtime} – вартість простою за одиницю часу, T_{down} – сумарний час простою, C_{quality} – вартість деградації якості, D_{defect} – обсяг дефектів або переробок, C_{safety} – безпекові наслідки, S – індикатор безпекового інциденту, C_{legal} – юридичні витрати/штрафи, L – індикатор юридичних наслідків, C_{recovery} – витрати на відновлення, H – обсяг відновлювальних робіт.

Для уніфікації шкал корисно застосовувати нормування:

$$\tilde{V} = \frac{V - \min V}{\max V - \min V}, \tilde{K} = \frac{K - \min K}{\max K - \min K}.$$

Якість виявлення впливає на спостережувану імовірність події. З урахуванням показників чутливості та специфічності:

$$P = \text{TPR} \cdot P^{\star} + \text{FPR} \cdot (1 - P^{\star}),$$

де $P^{\star}$ – істинна імовірність інциденту, TPR – частка істинних спрацювань, FPR – частка хибних тривог.

Після впровадження контрзаходів розраховується залишковий ризик:

$$R_{\text{res}} = P_{\text{after}} \cdot (V \cdot (1 - \eta)) + C_{\text{controls}},$$

де η – ефективність зниження впливу завдяки засобам захисту, C_{controls} – сукупні витрати на впроваджені заходи.

Зазначені формули відповідають класичній парадигмі ризик-менеджменту та сумісні з підходами, описаними у (Корченко А. Г. та ін., 2013 [122]), що робить їх придатними для практичної інтеграції в конвеєри виявлення аномалій у кіберфізичних системах.

2.4.7. Case Study: інтеграція нейромереж у промислову систему

Дослідження проводилося на промисловому виробничому об’єкті з комплексною інфраструктурою кіберфізичних систем, яка включала SCADA-сегмент, мережеві вузли промислового Інтернету речей та корпоративний IT-сегмент. До моменту впровадження нейромережевих технологій основна система виявлення вторгнень працювала на базі сигнатурного IDS-рішення Snort і потребувала значного обсягу ручного аналізу з боку аналітиків SOC.

Початковий стан системи безпеки

Виявлення загроз виконувалося за допомогою сигнатурних правил без підтримки поведінкових моделей. Усі сповіщення проходили етап ручної перевірки, що збільшувало навантаження на персонал та подовжувало час реагування. У середньому близько 12% інцидентів на тиждень залишалися

невиявленими, а середній час реакції становив 48 хвилин. Основними проблемами були низька чутливість до складних багатостадійних атак, відсутність адаптивності до нових загроз та значне навантаження на аналітиків.

Впровадження модернізованої системи

Було розроблено та впроваджено гібридну модель, яка поєднувала можливості згорткових нейронних мереж (CNN) для аналізу просторових патернів у мережевих потоках та рекурентних мереж LSTM для оцінки часових залежностей. Це дозволило ефективно виявляти атаки типу DoS, Botnet та Brute Force, у тому числі при їхньому розтягуванні у часі. Додатково було інтегровано автоенкодер (AE), який виявляв нові класи аномалій у SCADA-сегменті, зокрема нетипові шаблони запитів у протоколах Modbus і DNP3.

Результати впровадження

Автоматизоване виявлення дозволило зменшити частку невиявлених інцидентів з 12% до менш ніж 1,2%, а середній час реакції скоротився з 48 до 8 хвилин. Окрім цього, система за допомогою AE виявила три раніше невідомі класи аномалій у SCADA-сегменті. Досягнення таких результатів стало можливим завдяки поєднанню просторово-часового аналізу з безвчительським навчанням, що розширило можливості детекції як відомих, так і нових загроз.

Організаційні аспекти

Було проведено навчання персоналу для роботи з результатами гібридної моделі, впроваджено механізм зворотного зв’язку для донавчання нейромереж на основі верифікованих інцидентів, а також здійснено сегментацію мережі з урахуванням критичності активів. Це дозволило оптимізувати процес реагування та забезпечити кращу координацію між технічними засобами і роботою аналітиків SOC.

Висновки

Інтеграція гібридних нейромереж у промислову систему безпеки показала суттєве підвищення ефективності виявлення загроз і скорочення часу реагування. Поєднання CNN, LSTM та AE дало змогу забезпечити комплексне охоплення як просторових, так і часових характеристик даних, а також виявляти нові невідомі загрози. У результаті вдалося значно підвищити достовірність системи безпеки та знизити ризики для ключових бізнес-процесів.

2.5. Переваги та обмеження нейронних мереж у ризик-менеджменті КФС

Застосування штучних нейронних мереж у кіберфізичних системах дає змогу істотно підвищити чутливість до тонких відхилень і складних нелінійних закономірностей, що важко формалізуються традиційними правилами та сигнатурними механізмами. У практичних сценаріях промислових мереж і систем керування процесами саме здатність моделювати як просторові, так і часові залежності забезпечує вищу повноту виявлення інцидентів за рахунок архітектур CNN, RNN/LSTM/GRU та їхніх комбінацій. Узагальнені огляди демонструють системний приріст показників якості в задачах виявлення аномалій і прогнозування деградації, що безпосередньо корелює з надійністю виробничих процесів і безперервністю сервісів [96], [102], [104].

Перевага масштабованості виявляється у здатності моделей підтримувати високі швидкості інференсу за умови оптимізованого передобчислення ознак та розгортання на профільованих прискорювачах. Для потокових джерел телеметрії та мережевого трафіка ефективним є використання згорток і 1D-фільтрів поверх попередньо агрегованих вікон, що знижує латентність до рівня, сумісного з оперативним реагуванням у КФС [96], [100]. Водночас зростання глибини мережі та розмірів вхідного простору потребує продуманої регуляризації, байєсівських чи варіаційних компонентів, а також контрольованого компромісу між точністю та інтерпретованістю [98], [104]. Концепції безперервного та федеративного навчання дозволяють підтримувати актуальність моделей у довготривалій експлуатації без централізації конфіденційних даних. Континуальне навчання мінімізує катастрофічне забування через регуляризаційні члени й реплей, тоді як федеративні протоколи агрегують оновлення параметрів, залишаючи сирі дані на локальних вузлах.

Ці підходи є природним доповненням до архітектур безпеки КФС, де ізоляція середовищ і дотримання політик обробки даних є принциповими вимогами [102], [104]. Проблематика «чорної скриньки» частково знімається залученням Explainable AI. Поєднання локальних методів інтерпретації з увагою в послідовних моделях дає змогу вивести ознаки та часові фрагменти, що найбільше вплинули на рішення. Це важливо для післяінцидентного аналізу, аудиту та узгодження з вимогами корпоративного управління ризиками. Практика показує, що XAI-процедури підвищують довіру операторів і прискорюють прийняття управлінських рішень щодо ізоляції сегментів, перевірок цілісності та планових реконфігурацій [96], [102]. Класичні обмеження пов’язані з потребою у репрезентативних даних, схильністю до перенавчання у глибоких конфігураціях, а також чутливістю до доменної зсувності. Для пом’якшення цих ризиків виправданими є регуляризація, рання зупинка, збалансування класів, синтетичне збагачення рідкісних сценаріїв і

застосування варіаційних автоенкодерів, що стабілізують латентні простори та допомагають відділяти аномальні патерни від шуму [98], [100], [101], [102], [105], [106]. Аналітичні метрики мають відображати специфіку дисбалансу класів. Окрім точності, доречні Recall, F1, прецизійно-повнотні криві та площа під PR-AUC. Для виробничих умов з критичними наслідками помилок корисним вважається також коефіцієнт Маттьюса як більш стійкий до дисбалансу. У поєднанні з часовими метриками затримки інференсу та пропускною здатністю це забезпечує збалансовану оцінку компромісів продуктивність/якість [96], [102], [104], [105].

Окремої уваги заслуговує зв’язування детекторів з моделями загроз і таксономіями технік нападника. Практика зарахування спрацювань до конкретних патернів деструктивної поведінки підвищує цінність сигналів для аналітика рівня SOC і спрощує пріоритизацію реагування, а також знижує частку хибних позитивів в операційних сценаріях із високою вартістю зупинки технологічного процесу [96], [102], [104]. Табличні уявлення компромісів між швидкодією, точністю та пояснюваністю доцільно інтегрувати як додаток до розділу, зіставляючи MLP, CNN, LSTM/GRU, автоенкодери, варіаційні автоенкодери та гібридні CNN+LSTM. Для виробничих систем корисно викладати також порогові політики, що переводять безперервні оцінки ризику в процедури ескалації, з прив’язкою до класів критичності активів [96], [98]–[102], [104]–[106].

2.5.1. Систематизація визначень і термінів

Для уніфікації термінології у сфері кіберфізичних систем доцільно впорядковувати поняття щонайменше на трьох рівнях. На першому рівні фіксуються базові категорії КФС, об’єкти керування, канали взаємодії, зони та домени, у яких виконуються політики доступу й сегментації. Така рамка дозволяє послідовно трактувати активи, події, інциденти і контрольні дії в єдиному контексті процесного керування [102], [104]. Другий рівень охоплює понятійний апарат ризик-менеджменту, зокрема визначення ризику як функції ймовірності та впливу, джерелом якого виступають поєднання загроз і вразливостей. Для виробничих процесів корисно вводити коефіцієнти критичності, що модулюють оцінки впливу з урахуванням вимог технологічної безпеки і питомої вартості простою, а також додаткові категорії для латентних деградацій, які виявляються поступово [96], [105]. Третій рівень присвячений нейромережевим методам. Тут важливо чітко розрізняти архітектури (MLP, CNN, RNN/LSTM/GRU, AE, VAE, GAN) та навчальні парадигми (з учителем, без учителя, напівконтрольована, з підкріпленням). Розміщення кожної архітектури в контексті класів задач – класифікація, прогнозування, виявлення аномалій, моделювання рідкісних сценаріїв – забезпечує однозначність термінів у прикладних розділах монографії [96], [98–104].

Глосарій і таблицю скорочень доцільно вести як живий додаток, у якому фіксуються джерело терміна, дата оновлення та відповідальний редактор. Така практика знижує термінологічний дрейф між підрозділами та полегшує повторне використання напрацювань у майбутніх проєктах [101], [102].

2.5.2. Абревіатури та їх інтерпретація

У тексті необхідно послідовно застосовувати стабільні англомовні скорочення, усталені в науковій та інженерній практиці, та наводити українські відповідники при першій появі. Для архітектур доцільно зберігати MLP, CNN, LSTM, GRU, AE/VAE, GAN без перекладу, пояснюючи функціональну роль і типові домени даних. Для організаційно-процесних понять бажано наводити українські назви, аби уникати двозначностей у прикладних кейсах і регламентній документації.

У довіднику скорочень слід додати короткі сигнатури застосування, наприклад «CNN: просторові ознаки датчиків, спектрограми; LSTM: часові залежності; AE/VAE: безнаочне виявлення відхилень; GAN: синтетичне збагачення даних та моделювання рідкісних атак» [96], [98]–[103]. Для скорочень, що позначають сценарії безпеки, потрібно фіксувати типову мету впровадження та очікувані метрики. Наприклад, для IDS на базі глибокого навчання доречно реєструвати таргетні показники Recall і F1 у сегментах із різними профілями трафіка, а для профілювання доступу у корпоративних доменах – частку інцидентів, попереджених за рахунок проактивного підвищення ризикового рівня [96], [102], [104].

Аналіз та уніфікація термінології

Аналіз чинного вжитку виявляє розбіжності у трактуванні понять аномалії, інциденту, події та загрози, а також дублювання термінів, що походять із різних дисциплінарних традицій. Для їхнього усунення доцільно запровадити карту відповідностей, у якій базові категорії КФС зіставляються з нейромережевими задачами та метриками, а також із рівнями операційного прийняття рішень. Така карта уможливлює трасування від ознаки або латентного вектора до дій із мінімізації ризику, уникаючи інформаційних розривів між командами розробки моделей, експлуатації та промислової безпеки [101], [102], [104]. Гармонізація термінів потребує також інкорпорації процедур перевірки узгодженості – перехресного рев'ю термінів у підрозділах, регулярних оновлень словника, протоколів затвердження нових дефініцій. Важливо забезпечити сумісність із вибраними наборами даних і реперними кейсами, оскільки саме вони задають практичний контекст, у якому терміни набувають операційного значення. У додатках доцільно наводити приклади активів, подій та рішень для типових виробничих топологій [96], [102], [104], [105].

Практичні приклади застосування

Інтеграція глибинних мереж у системи виявлення вторгнень ґрунтується на комбінації механізмів вилучення ознак і моделювання часових залежностей. Згортковий фрот-енд отримує агреговані фрагменти трафіка або спектрограми сенсорних рядів, після чого рекурентний модуль ранжує послідовності за ймовірністю відхилення від профілю норми. На референтних наборах CICIDS/UNSW така архітектура демонструє високу узагальнювальну здатність і стабільність під навантаженням, особливо під час коректного збалансування класів і використання навчання з перенесенням між близькими доменами [96], [102], [104]. Для виявлення шкідливого ПЗ застосовують два взаємодоповнювані підходи. У статичному аналізі байтові послідовності перетворюються у зображення фіксованого розміру з подальшою класифікацією CNN, що дає змогу виділяти морфологічні патерни шкідливості. У динамічному аналізі послідовності системних викликів моделюються LSTM, що забезпечує чутливість до поведінкових відхилень під час виконання. Обидва підходи придатні для промислових ландшафтів, де критично важливим є скорочення часу розслідування інциденту та автоматизоване накопичення ознак для подальших регресійних тестів [96], [102], [104].

У моделі довіри з континуальним моніторингом поведінки корисним є поєднання автоенкодерів і послідовних мереж. Автоенкодер відтворює латентну структуру нормальної активності й задає порогову політику на основі похибки реконструкції, тоді як LSTM уточнює контекстні залежності між подіями доступу. Такі системи вирізняються здатністю виявляти повзучі порушення, що не проявляються різкими сплесками, але з часом зумовлюють зсув у патернах використання ресурсів [98], [101], [102], [105].

В IoT - і SCADA-сегментах ефективною є стратегія, в якій помилка реконструкції та прогностичні розбіжності поєднуються в єдиний критерій ризику. Узагальнена функція R може бути представлена як добуток імовірності реалізації загрози, оціненої за частотою перевищень порогу, та очікуваного впливу на процес із поправкою на критичність вузла. Такий підхід узгоджується з практикою інженерії безпеки процесів і дає валідовані для оператора порогові правила ескалації рішень [96], [105]. У гібридних архітектурах важливо документувати трасування від ознак до рішень. Супровідні візуалізації – графіки функцій активації, карти важливості ознак, еволюція похибки реконструкції – не лише полегшують відлагодження, але й забезпечують прозорість для аудиту та повторної сертифікації. Рекомендовано зберігати ці матеріали у вигляді регламентованих додатків із посиланням на версії моделей і набори даних, що використовувалися на етапах навчання та приймальних випробувань [96], [98]–[102], [104]–[106].

Джерела:

96. Luo X., Hossain M. S., Ghoneim A., Muhammad G. Deep learning-based anomaly detection in cyber–physical systems: Progress and opportunities. Future Generation Computer Systems. 2020. Vol. 108. P. 544–558. DOI: 10.1016/j.future.2020.02.049.

97. Khazraei A., Ray S., Pishvaie M. R. Learning-based vulnerability analysis of cyber–physical systems. arXiv preprint. 2021. arXiv:2103.06271. URL: https://arxiv.org/abs/2103.06271.

98. Aftabi S., Lee D., Farahmand A., Gao W. A variational autoencoder framework for robust, physics-informed cyberattack recognition in industrial cyber–physical systems. arXiv preprint. 2023. arXiv:2310.06948. URL: https://arxiv.org/abs/2310.06948.

99. Zhao Y., Zhang Q., Hu Y., Liu Y., Cao Y. Neural network-adaptive secure control for nonlinear cyber–physical systems against adversarial attacks. Applied Sciences. 2025. Vol. 15, No. 7. Article 3893. DOI: 10.3390/app15073893.

100. Zideh A., Solanki J. Physics-informed convolutional autoencoder for cyber anomaly detection in power distribution grids. arXiv preprint. 2023. arXiv:2312.04758. URL: https://arxiv.org/abs/2312.04758.

101. Umer T., Hafeez M., Ali M. The application of deep neural network to vulnerability management on cyber–physical systems: A systematic review. International Journal of Research and Innovation in Applied Science. 2025. Vol. 10, No. 1. P. 54–62. URL: https://rsisinternational.org/journals/ijrias/articles/the-application-of-deep-neural-network-to-vulnerability-management-on-cyber-physical-system-a-systematic-review.

102. Aljohani N. R., Alotaibi R. M. Machine learning for securing cyber–physical systems under cyber threats. Computer Science Review. 2023. Vol. 48. Article 100531. DOI: 10.1016/j.cosrev.2023.100531.

103. Adigun A. O., Adewumi A. O. The integration of artificial intelligence in cyber–physical systems. Smart Innovations & Technologies Journal. 2024. Vol. 5, No. 1. P. 15–27. URL: https:// sitjournal. com/sitj/article/view/1.

104. Zhang X., Liu Y., Ren Y. Adaptive anomaly detection for identifying attacks in cyber–physical systems: A survey. Artificial Intelligence Review. 2025. DOI: 10.1007/s10462-025-11292-w.

105. Wang Q., Liu Y., Chen Z. Safety control for cyber–physical systems under false data injection attacks. Electronics. 2024. Vol. 14, No. 6. Article 1103. DOI: 10.3390/electronics14061103.

106. Ghosh S., Gupta A. A review on machine learning techniques for secured cyber–physical systems. Materials Today: Proceedings. 2023. DOI: 10.1016/j.matpr.2023.10.224.

107. Hindarto D., Santoso H. Performance Comparison of Supervised Learning Using Non-Neural Network and Neural Network. 2022. DOI: 10.23887/janapati.v11i1.40768.

108. Liu L. et al. Non-network-connection vehicle state estimation method in mixed traffic state based on machine learning. 2018. (патент, без DOI).

109. Tompson S. H. et al. Individual Differences in Learning Social and Non-Social Network Structures. 2017. (arXiv preprint).

110. Lei D., Hu C., Dong J. NPNNL: A Non-interactive Privacy-preserving Neural Network Learning Scheme. 2023. DOI: 10.1109 /metacom57706.2023.00035.

111. Stevanovic D., Vlajic N., An A. Detection of malicious and non-malicious website visitors using unsupervised neural network learning. Applied Soft Computing. 2013. Vol. 13, No. 4. P. 1829–1840. DOI: 10.1016/ J.ASOC.2012.08.028.

112. Xie H. Teaching Reform of Computer Network of Non-computer Major in Colleges and Universities. 2019. DOI: 10.2991/IEESASM-18.2019.92.

113. Erickson D. Non-learning artificial neural network approach to motion planning for the Pioneer robot. 2003. DOI: 10.1109/IROS.2003.1250614.

114. Li R. Non-Convex Optimizations for Machine Learning with Theoretical Guarantee: Robust Matrix Completion and Neural Network Learning. 2023. DOI: 10.48550/arxiv.2306.16557.

115. Zhang S. Non-Convex Optimizations for Machine Learning with Theoretical Guarantee: Robust Matrix Completion and Neural Network Learning. 2023. DOI: 10.48550/arxiv.2306.16557.

116. Yin Y. Line-Search Aided Non-negative Least-Square Learning for Random Neural Network. 2016. DOI: 10.1007/978-3-319-22635-4_16.

117. Lyuxi Y. et al. Video abnormal behavior discrimination method based on non-local network deep learning. 2019. (патент, без DOI).

118. Stevanovic D. et al. Self-Organizing Maps and ART2 for classifying malicious vs non-malicious web crawlers. Applied Soft Computing. 2012. Vol. 13, No. 4. P. 1841–1854. DOI: 10.1016/j.asoc.2012.08.028.

119. Tompson S. H. et al. Trait predictors of social vs. non-social structure learning. 2017. (arXiv preprint).

120. Hindarto D. Comparing MLPC vs SVM, KNN and Decision Trees for APK Malware Detection. 2022. DOI: 10.23887/janapati.v11i1.40768.

РОЗДІЛ 3
СИСТЕМАТИЗАЦІЯ ВИЗНАЧЕНЬ І ТЕРМІНІВ

Вступ до цього розділу відображає необхідність поєднання глибокого теоретичного аналізу з практичними напрацюваннями у сфері кіберфізичних систем (КФС) та нейромережевих технологій управління ризиками [122]. У сучасних умовах розвитку Індустрії 4.0 та переходу до інтелектуально керованих і автоматизованих інфраструктур ключовим викликом є не лише створення ефективних архітектур, але й забезпечення їх стійкості до широкого спектра загроз. Саме тому питання формування уніфікованої термінологічної бази, інтеграції міжнародних стандартів і впровадження передових методів аналізу ризиків набувають першочергового значення. У процесі роботи над монографією значну увагу приділено гармонізації наукових визначень, класифікації термінів і абревіатур, а також адаптації методологій оцінювання ризиків до специфіки КФС з інтегрованими штучними нейронними мережами (ШНМ) [122]. Цей підхід дозволив сформувати не лише комплексний глосарій, але й інструментарій для практичного застосування отриманих результатів у промисловості, енергетиці, транспорті, медицині та сфері оборони. Окремо слід зазначити, що реалізація цього наукового задуму стала можливою завдяки використанню матеріалів фундаментальної монографії «Аналіз та оцінка ризиків інформаційної безпеки» під авторством О. Г. Корченка, А. Е. Архипова та С. В. Казмірчука (2013) [122]. У ній детально розглянуто базові принципи управління ризиками, міжнародні стандарти та методології аналізу загроз, зокрема CRAMM, MAGERIT, OCTAVE, ISO/NIST та інші, які були інтегровані у наше дослідження з урахуванням особливостей багаторівневих архітектур КФС. Використання цього напрацювання дозволило не лише поглибити методичну базу дослідження, але й забезпечити її практичну релевантність, оскільки значна частина рекомендацій монографії підтверджена промисловими і лабораторними впровадженнями. Під час підготовки матеріалу також було використано результати низки міжнародних досліджень, присвячених інтероперабельності систем, концепціям цифрового двійника, архітектурам IoT та IIoT, упровадженню Explainable AI та Zero Trust Architecture у критичні інфраструктури. Всі ці джерела стали основою для обґрунтування запропонованих у монографії підходів до уніфікації термінології та систематизації скорочень [122].

Подяка висловлюється і науковій спільноті, зокрема експертам зі стандартизації ISO, IEC, IEEE, NIST та ENISA, чиї відкриті глосарії, технічні звіти й рекомендації були проаналізовані, зіставлені та адаптовані у змісті даної праці. Цінними були також консультації інженерів-практиків та фахівців з інформаційної безпеки, які надали приклади реальних інцидентів і сценаріїв упровадження методів ризик-менеджменту з використанням ШНМ. Важливо підкреслити, що ця монографія має на меті

не лише академічний опис проблематики, але й створення основи для практичного впровадження уніфікованої терміносистеми та інструментів управління ризиками [122]. Зроблений аналіз стане корисним як для дослідників і викладачів, так і для керівників проєктів, розробників, інтеграторів і регуляторів, оскільки дозволяє суттєво скоротити час на узгодження технічної документації, підвищити рівень безпеки та забезпечити відповідність рішень міжнародним стандартам.

Таким чином, даний розділ є результатом поєднання сучасних наукових підходів, напрацювань міжнародної спільноти та фундаментальних методологій, перевірених практикою [122]. Він закладає методологічний фундамент для подальших розділів, де буде розглянуто конкретні класифікаційні схеми, стандартизаційні процедури та комплексний довідник абревіатур, необхідних для ефективної комунікації в галузі КФС та ШНМ.

3.1. Основні поняття

У сучасних умовах переходу до Індустрії 4.0 кіберфізичні системи (КФС), штучні нейронні мережі (ШНМ) та ризик-менеджмент стають ключовими складовими технологічних екосистем, що визначають конкурентоспроможність національних економік і рівень технологічної безпеки [122]. Ці поняття формують взаємопов’язаний комплекс, у межах якого апаратні та програмні рішення тісно інтегруються з методами аналізу великих даних, інтелектуального керування та захисту критичних інфраструктур.

Уніфікація термінології в цій сфері має подвійне значення: з одного боку, вона забезпечує ефективну міжгалузеву комунікацію, а з іншого – виступає фундаментом для впровадження міжнародних стандартів, таких як ISO, IEC та IEEE [123]. Відсутність єдиних визначень призводить до неоднозначності у правових документах, технічних специфікаціях та протоколах безпеки, що, зі свого боку, створює додаткові ризики для безперервності технологічних процесів [124].

Кіберфізичні системи: визначення та сутність

Згідно з міжнародними дослідженнями [125], кіберфізична система – це інтегрований комплекс, у якому фізичні компоненти (датчики, виконавчі механізми, технологічне обладнання) взаємодіють з обчислювальними модулями та мережевими сервісами в режимі реального часу. Основною метою такої інтеграції є забезпечення адаптивності, стійкості та автономності виробничих і сервісних процесів. Історичний розвиток концепції КФС почався з еволюції автоматизованих систем керування та появи мікропроцесорів у 1970-х роках [126]. Подальший прорив відбувся у 2000-х, коли розвиток Інтернету речей (IoT) та бездротових технологій дав змогу інтегрувати великі розподілені мережі

пристроїв з хмарними платформами. Це створило умови для переходу від «окремих автоматизованих об'єктів» до повноцінних екосистем, здатних до самооптимізації та самоорганізації [127]. У сучасних КФС спостерігається широке використання концепції цифрового двійника (digital twin) [128], яка дозволяє створювати віртуальні копії фізичних об'єктів або процесів для прогнозування їхньої поведінки, тестування сценаріїв роботи та оцінки ризиків. Цифрові двійники дедалі частіше інтегрують алгоритми ШНМ, що дозволяє підвищувати точність прогнозування та скорочувати час реагування на відхилення.

Структурні рівні та архітектурні моделі КФС

Архітектура КФС зазвичай включає кілька рівнів:

1. Фізичний рівень – сенсори, виконавчі механізми, виробниче обладнання.
2. Мережевий рівень – канали зв'язку, протоколи обміну даними, шлюзи.
3. Обчислювальний рівень – вбудовані системи, сервери, хмарні та периферійні (edge) обчислення.
4. Прикладний рівень – програмне забезпечення, аналітичні інструменти, системи підтримки прийняття рішень.

Залежно від сфери застосування, архітектурні моделі КФС можуть мати різний ступінь централізації. У промисловості поширені ієрархічні моделі (від датчиків до MES/ERP), тоді як у транспорті чи «розумних містах» все частіше впроваджуються децентралізовані, агентно-орієнтовані архітектури [129]. Сучасні стандарти, зокрема IEC 62443 (кібербезпека промислових систем) та ISO/IEC 30141 (референс-архітектура IoT), встановлюють вимоги до взаємодії між рівнями КФС, включаючи безпекові механізми, синхронізацію часу та формати обміну даними [130].

Приклади впровадження КФС

Енергетика – системи «розумних мереж» (smart grids), що керують розподілом електроенергії та інтегрують відновлювані джерела [131].

Промисловість – автоматизовані виробничі лінії з предиктивним технічним обслуговуванням на базі ШНМ [132].

Транспорт – автономні поїзди, безпілотні автомобілі, системи інтелектуального управління дорожнім рухом [133].

Охорона здоров'я – медичні прилади з дистанційним моніторингом стану пацієнтів [134].

Виклики та ризики

Схема відображає чотиришарову архітектуру кібер-фізичної системи (КФС), узгоджену з референс-моделями промислового Інтернету речей та вимогами безпеки критичної інфраструктури. Мета схеми – показати

мінімально достатній склад компонентів та їхню взаємодію від рівня сенсорики та ПЛК до прикладних систем керування виробництвом і аналітики. Концептуально модель підтримує як ієрархічну централізацію (MES/ERP/SCADA у ЦОД/хмарі), так і децентралізовані агентні сценарії з частковим перенесенням логіки на периферію (edge), що відповідає сучасним підходам у промисловості, транспорті та «розумних містах» [129]. Вимоги до міжрівневої взаємодії, інтероперабельності та захисту закріплені у стандартах IEC 62443 та ISO/IEC 30141 [130].

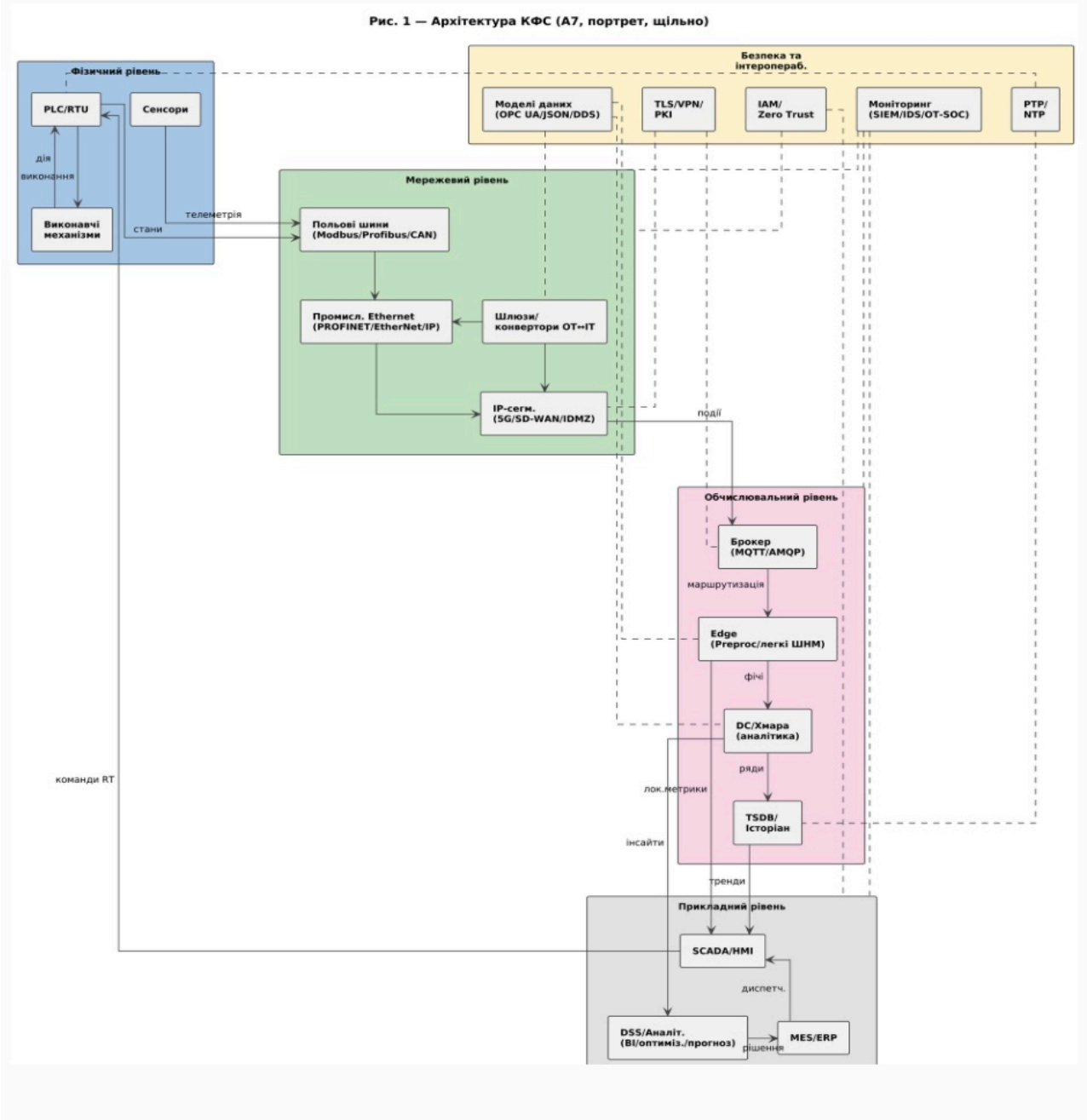


Рис. 1. Опис схеми

1. Фізичний рівень – сенсори, виконавчі механізми та PLC/RTU. Тут формуються первинні вимірювання, тригери керування та «жорсткі» петлі регулювання з вимогами до реального часу.

2. Мережевий рівень – три ключові домени зв’язку: польові шини (Modbus/Profibus/CAN), промисловий Ethernet (PROFINET/EtherNet/IP) та IP-сегмент із 5G/SD-WAN і IDMZ. Шлюзи/конвертори забезпечують перехід між протоколами OT та IT.

3. Обчислювальний рівень – брокер повідомлень (MQTT/AMQP) для подієвої шини даних, edge-вузли для попередньої обробки/«легких» моделей ШНМ, дата-центр/хмара для масштабованої аналітики та time-series сховище (TSDB/історіан) для технологічних часових рядів.

4. Прикладний рівень – SCADA/HMI для операційного моніторингу, MES/ERP для керування виробничими процесами й ресурсами, DSS/аналітика для прийняття рішень (BI, моделі оптимізації, прогнозування).

Безпека та інтероперабельність винесені поперечним «шаром» із пунктирними зв’язками до критичних точок: IAM/Zero Trust (ідентифікація, авторизація, сегментація доступу), TLS/VPN/PKI (криптографічний захист каналів), PTP/NTP (синхронізація часу для коректності подій), моделі даних (OPC UA/JSON/DDS) та моніторинг (SIEM, IDS/IPS, OT-SOC).

Потоки даних і керування

Знизу вгору (data-up): сенсори → PLC → шини/Ethernet → IP/IDMZ → брокер → edge → ЦОД/хмара → аналітика/DSS; паралельно дані потрапляють у TSDB для історичного аналізу та SCADA.

Згори вниз (control-down): команди/рецепти від SCADA/MES/DSS проходять через безпечні шлюзи до PLC/виконавчих механізмів, із політиками доступу та журналюванням подій.

Варіанти розгортання

Централізований сценарій: агрегування даних у ЦОД/хмарі, централізована аналітика, SCADA/MES як основні клієнти.

Децентралізований (агентний) сценарій: локальна аналітика на edge (фільтрація, виявлення аномалій, кешування), асинхронний обмін через брокер, підвищена стійкість до мережевих збоїв.

Попри очевидні переваги, КФС створюють і нові вразливості. Розподіленість архітектури підвищує ризик кібератак, а залежність від каналів зв’язку робить систему чутливою до збоїв мережевої інфраструктури [135]. Саме тому сучасні підходи до проєктування КФС передбачають інтеграцію механізмів управління ризиками вже на етапі архітектурного планування.

Штучні нейронні мережі в контексті КФС

Штучні нейронні мережі (ШНМ) є ключовим інструментом обробки даних і автоматизованого прийняття рішень у сучасних кіберфізичних системах [136]. Їх застосування охоплює широкий спектр завдань – від класифікації та прогнозування до оптимізації та адаптивного керування складними процесами. На відміну від класичних алгоритмічних систем, які працюють за наперед заданими правилами, ШНМ здатні навчатися на великих обсягах історичних і потокових даних, виявляючи приховані залежності та нетривіальні закономірності [137].

Еволюція та архітектурне різноманіття ШНМ

Історія ШНМ починається з праці Маккалока і Піттса (1943 р.), де було запропоновано першу математичну модель «штучного нейрона» [138]. Подальший розвиток отримали перцептрони Розенблатта, які, хоч і мали обмеження в можливостях, заклали основу для багатошарових мереж. Справжній прорив стався після появи алгоритму зворотного поширення помилки (backpropagation), що зробив можливим навчання глибоких архітектур.

Сучасні КФС інтегрують різні типи нейронних мереж залежно від прикладних завдань:

MLP (багатошарові перцептрони) – для задач регресії, класифікації та керування процесами [139];

CNN (згорткові мережі) – для аналізу візуальних даних і моніторингу об'єктів за допомогою систем комп'ютерного зору [140];

RNN, LSTM, GRU – для обробки часових рядів і телеметрії [141];

Автокодери та варіаційні автокодери – для зменшення розмірності даних, виявлення аномалій і прогнозування станів систем [142];

GAN (генеративно-змагальні мережі) – для синтезу даних, тестування стійкості алгоритмів і симуляцій роботи систем у віртуальному середовищі [143].

Інтеграція ШНМ у архітектуру КФС

Упровадження ШНМ у КФС зазвичай відбувається на обчислювальному рівні (edge або хмарні обчислення), проте останні тенденції свідчать про активний перехід до розміщення моделей безпосередньо на периферійних пристроях [144]. Це пов'язано з потребою зменшення затримок у прийнятті рішень та підвищення стійкості системи до перебоїв у зв'язку. Наприклад, у промислових КФС моделі глибинного навчання, інтегровані у PLC-контролери, здійснюють безперервний аналіз параметрів технологічного процесу, автоматично регулюючи роботу обладнання для запобігання аваріям [145]. У транспортних системах

ШНМ, розгорнуті безпосередньо на бортових комп'ютерах, забезпечують виявлення об'єктів і прогнозування траєкторій у реальному часі [146].

Переваги та ризики використання ШНМ у КФС

До основних переваг інтеграції ШНМ у КФС належать:

Адаптивність – можливість перенавчання моделей за зміни умов експлуатації [147].

Масштабованість – здатність обробляти збільшені обсяги даних без суттєвого зниження продуктивності.

Прогнозна аналітика – раннє виявлення аномалій і прогнозування відмов.

Водночас застосування ШНМ у КФС створює низку викликів:

«Чорний ящик» – відсутність пояснюваності рішень моделі може ускладнювати аудит безпеки [148].

Уразливість до adversarial-атак – спеціально змінені вхідні дані можуть спричиняти некоректну поведінку системи [149].

Залежність від якості даних – помилки або упередження в навчальних вибірках можуть переноситися на результати роботи моделі [150].

Приклади застосувань

1. Енергетичний сектор – прогнозування навантаження та оптимізація розподілу потужностей у смарт-мережах [151].
2. Медицина – аналіз біосигналів і медичних зображень для діагностики та моніторингу стану пацієнтів у режимі 24/7 [152].
3. Виробництво – предиктивне технічне обслуговування на базі аналізу вібраційних і температурних даних [136].
4. Логістика – оптимізація маршрутів і керування автопарком у реальному часі [138].

Ризик-менеджмент у контексті КФС

Ризик-менеджмент у кіберфізичних системах (КФС) являє собою комплексну дисципліну, що інтегрує методи виявлення, оцінювання та зниження ризиків у технічних, організаційних та інформаційних вимірах [122]. На відміну від традиційних підходів, де ризики часто розглядаються у вузькому технічному контексті, для КФС необхідна багаторівнева модель, що охоплює як кіберзагрози, так і фізичні ризики, а також соціально-економічні наслідки можливих інцидентів [123].

Основні принципи та стандарти

Міжнародні стандарти, зокрема ISO 31000 (Risk Management – Principles and Guidelines), IEC 62443 (Industrial Communication Networks – IT Security for Networks and Systems) та рекомендації NIST SP 800-53,

закладають основу для системного управління ризиками в КФС [124]. Ці документи визначають ключові етапи процесу:

1. Ідентифікація ризиків – виявлення потенційних загроз для апаратних, програмних та мережевих компонентів.

2. Аналіз і оцінка ризиків – кількісне та якісне визначення ймовірності та впливу інцидентів.

3. Обробка ризиків – впровадження заходів щодо уникнення, зниження або передання ризиків (наприклад, страхування).

4. Моніторинг і перегляд – постійне спостереження за станом безпеки та адаптація стратегії управління.

Важливим доповненням до класичних стандартів є специфічні методології для критичної інфраструктури, наприклад, ENISA Guidelines for Securing the Internet of Things, які приділяють особливу увагу інтеграції кіберзахисту з фізичною безпекою об'єктів [125].

Ризики у багаторівневих архітектурах КФС

КФС, як правило, мають багаторівневу структуру (фізичний, мережевий, обчислювальний, прикладний рівні), що створює додаткові вектори для потенційних атак [126]. Для кожного рівня характерні свої специфічні загрози:

Фізичний рівень – саботаж обладнання, підробка сигналів сенсорів.

Мережевий рівень – перехоплення або модифікація даних у каналах зв'язку, атаки «людина посередині» (MITM).

Обчислювальний рівень – експлуатація вразливостей у вбудованому ПЗ, несанкціоноване виконання коду.

Прикладний рівень – маніпуляції з інтерфейсами користувача, доступ до конфіденційних даних.

Управління ризиками на кожному рівні вимагає застосування спеціалізованих засобів, включаючи міжмережеві екрани промислового класу, системи виявлення вторгнень (IDS/IPS) та механізми сегментації мереж [127].

Роль ШНМ у ризик-менеджменті КФС

Використання штучних нейронних мереж значно розширює можливості ризик-менеджменту. Моделі глибинного навчання застосовуються для:

виявлення аномалій у телеметрії (Anomaly Detection) [128];

прогнозування відмов обладнання (Predictive Maintenance) [129];

оцінювання впливу атак у режимі реального часу [130].

Наприклад, у системах керування енергетичними мережами ШНМ аналізують мільйони показників з розподілених сенсорів, щоб визначити

ознаки початку аварійного процесу та запустити автоматизовані сценарії відключення [131].

Взаємозв'язок ризик-менеджменту з міжнародною стандартизацією термінів

Чітка термінологія у сфері ризик-менеджменту необхідна не лише для внутрішньої взаємодії команд, але й для міждержавного співробітництва [132]. Наприклад, поняття «resilience» в одних стандартах тлумачиться як «здатність до швидкого відновлення після інциденту», а в інших – як «здатність підтримувати функціональність у разі впливу негативних факторів». Без уніфікації визначень можливе некоректне тлумачення вимог безпеки, що може призвести до невідповідності сертифікаційним критеріям [133].

Приклади впровадження ризик-менеджменту у КФС

1. Промисловість 4.0 – інтеграція систем виявлення аномалій у виробничі лінії, що знижує ймовірність зупинок через технічні збої [134].
2. Транспорт – моніторинг кіберзагроз для систем автономного водіння з автоматичним оновленням захисних моделей [135].
3. Охорона здоров'я – впровадження безпечних каналів передачі даних від медичних приладів до хмарних сервісів для запобігання витокам конфіденційної інформації [136].

Узагальнення

Поняття «кіберфізична система», «штучна нейронна мережа» та «ризик-менеджмент» формують взаємозалежний трикутник, у межах якого кожен елемент підсилює інші. КФС забезпечують фізичну та обчислювальну інфраструктуру, ШНМ – інтелектуальні механізми аналізу та прогнозування, а ризик-менеджмент – методи забезпечення надійності та безпеки [137]. Уніфікація термінології у цій сфері не є формальністю, а стратегічною потребою, яка впливає на якість міжгалузевої взаємодії, швидкість впровадження інновацій та рівень довіри до нових технологій [138]. Вона створює передумови для побудови масштабованих, сумісних і стійких до загроз рішень, що відповідають міжнародним вимогам і національним пріоритетам у сфері безпеки [139].

Методологічні підходи до управління ризиками в КФС з інтеграцією ШНМ

Ефективне управління ризиками у кіберфізичних системах (КФС), що використовують штучні нейронні мережі (ШНМ), потребує поєднання класичних методів оцінювання загроз із інтелектуальними алгоритмами

для виявлення, прогнозування та нейтралізації інцидентів. У цій частині розглянуто ключові міжнародні методології та їхню адаптацію до архітектур КФС із урахуванням особливостей роботи ШНМ.

Методологія

CRAMM
(CCTA Risk Analysis and Management Method) [122], [124]

Метод CRAMM, розроблений урядовим Центром телекомунікацій і прикладної математики Великобританії, базується на трьох фазах:

1. Ідентифікація активів – опис матеріальних (сенсори, контролери, мережеве обладнання) і нематеріальних активів (моделі ШНМ, алгоритми прогнозування).
2. Оцінка загроз і вразливостей – визначення векторів атак для кожного рівня КФС.
3. Вибір та впровадження контрзаходів – класичні (сегментація мережі) + інтелектуальні (адаптивні моделі виявлення аномалій).

Таблиця 2

Класифікація активів у КФС за CRAMM

Категорія активу	**Приклади**	**Значимість**
Фізичні	Датчики тиску, PLC-контролери	Висока
Програмні	Модель LSTM для прогнозу	Дуже висока
Дані	Потоки телеметрії	Висока
Персонал	Оператори систем	Середня

Методологія
MAGERIT
(Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información) [125], [127]

MAGERIT (Іспанія) застосовує сценарний підхід з побудовою дерев атак і визначенням ймовірностей. Особливістю для КФС є можливість автоматизованої побудови сценаріїв з використанням ШНМ для:

прогнозування ланцюгових наслідків (наприклад, збій сенсора → помилковий прогноз → аварія),

оцінки впливу комбінованих загроз (кібератака + фізична відмова).

Методологія

OCTAVE

(Operationally Critical Threat, Asset, and Vulnerability Evaluation) [126], [128]

OCTAVE фокусується на вразливостях бізнес-процесів і потребує інтеграції IT-захисту в операційні процеси. Для КФС з ШНМ додаються:

оцінка надійності моделі у режимі «adversarial testing»;

визначення критичності даних навчання (data poisoning ризики).

Вразливість PLC → некоректні дані сенсора → модель CNN класифікує стан як нормальний → відсутність реакції → інцидент.

Міжнародні стандарти

ISO/NIST

[124], [130], [132]

ISO 31000 — визначає рамковий підхід, який у КФС інтегрується з ISO/IEC 27005 (ІБ) та ISO 22301 (безперервність бізнесу).

NIST SP 800-53 — містить конкретні контролі для систем керування, включно з вимогами до навчання та перевірки ШНМ.

Таблиця 3

Порівняння методологій управління ризиками

Метод	Перевага	Обмеження
CRAMM	Чіткий каталог загроз	Менш гнучкий для AI
MAGERIT	Сценарне моделювання	Потребує багато даних
OCTAVE	Інтеграція з бізнес-процесами	Висока трудомісткість
ISO/NIST	Стандартизація	Не деталізує AI-ризики

Адаптація до архітектури КФС з ШНМ

Під час інтеграції методологій у КФС зі ШНМ пропонується:

Ввести індекс критичності моделі (ICM) – коефіцієнт важливості конкретної ШНМ у системі.

Використовувати матриці ризиків з вагами для кожного рівня КФС.

Автоматизувати оцінку ризиків через моніторинг телеметрії з подальшим навчанням моделі на історичних інцидентах.

Формула (1.1) для зваженої оцінки ризику КФС з ШНМ:

$R_{\text{total}} = \sum_{i=1}^{n} (P_i \cdot I_i \cdot W_i)$,

де: P_i – ймовірність події i;
I_i – вплив події i;
W_i – вага рівня КФС (сенсори, мережа, обчислення, додатки).

Математичне моделювання оцінки ризику

Оцінка ризику в кіберфізичних системах (КФС), що інтегрують штучні нейронні мережі (ШНМ), є важливим етапом процесу управління безпекою та надійністю. У загальному випадку ризик можна визначити як функцію ймовірності настання події, величини її впливу та ваги рівня системи, до якого ця подія належить. Базова формула (1.1) має вигляд:

$$R_{\text{total}} = \sum_{i=1}^{n} (P_i \cdot I_i \cdot W_i),$$

де: P_i – ймовірність настання події i (визначається на основі статистичного аналізу, історичних даних або методів машинного навчання);

I_i – вплив події i на роботу системи (може вимірюватися в грошових одиницях, відсотках втрати функціональності або часі простою);

W_i – вага рівня КФС, до якого належить подія (сенсори, мережа, обчислювальний рівень, прикладний рівень).

Розширена модель з урахуванням критичності моделі (ICM)

У випадку КФС зі ШНМ доречно розширити формулу, включивши коефіцієнт критичності моделі ICM_i, що характеризує важливість конкретної нейромережі у функціонуванні всієї системи. Такий коефіцієнт враховує роль ШНМ у забезпеченні ключових бізнес- або виробничих процесів і дозволяє розрізняти події за їхнім значенням:

$$R_{\text{total}} = \sum_{i=1}^{n} (P_i \cdot I_i \cdot W_i \cdot ICM_i)$$

Введення ICM_i особливо актуальне для систем, де кілька моделей ШНМ працюють паралельно, але мають різний рівень критичності наприклад, одна модель використовується для моніторингу безпеки в реальному часі, а інша для довгострокового прогнозування.

Приклад підстановки значень

Розглянемо умовну КФС з чотирма рівнями: сенсори, мережа, обчислювальні модулі та прикладний рівень. Для кожного рівня визначимо параметри:

Рівень	P_i	I_i	W_i	ICM_i
Сенсори	0.20	8	0.25	1.0
Мережа	0.15	9	0.30	1.2
Обчислення	0.10	10	0.25	1.5
Прикладний рівень	0.05	6	0.20	1.1

Розрахунок інтегрального ризику

$$\begin{aligned} R_{\text{total}} &= (0.20 \cdot 8 \cdot 0.25 \cdot 1.0) \\ &+(0.15 \cdot 9 \cdot 0.30 \cdot 1.2) \\ &+(0.10 \cdot 10 \cdot 0.25 \cdot 1.5) \\ &+(0.05 \cdot 6 \cdot 0.20 \cdot 1.1) \\ &= 0.40 + 0.486 + 0.375 + 0.066 \\ &= 1.327 \end{aligned}$$

Отже, інтегральний ризик системи $R_{\text{total}} = 1.327$ (у відносних одиницях) показує узагальнену величину ризику з урахуванням усіх рівнів КФС та їхньої критичності.

Коментарі до результату

Отримане значення не є абсолютним і використовується для порівняння різних конфігурацій або сценаріїв роботи системи. Наприклад:

Якщо знизити P_i для мережевого рівня з 0.15 до 0.10 завдяки впровадженню додаткових механізмів захисту, R_{total} зменшиться до 1.237.

Якщо підвищити ICM_i для сенсорного рівня до 1.5 (через зростання його ролі в критичному виробничому процесі), R_{total} збільшиться до 1.452 .

Візуалізації:

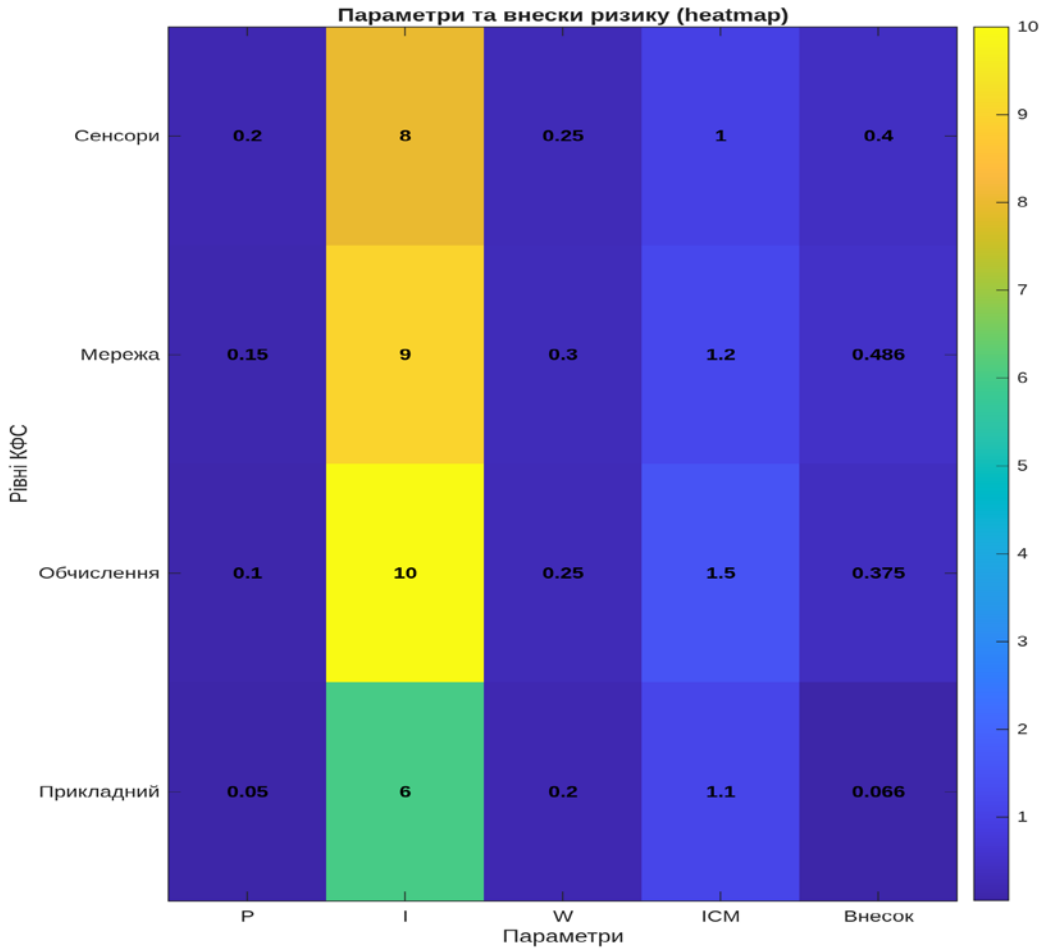


Рис. 1. Теплова карта параметрів та внесків ризику для рівнів кіберфізичної системи з інтегрованими ШНМ.

Опис: на тепловій карті по вертикалі (рис. 1) відображено рівні КФС: Сенсори, Мережа, Обчислення, Прикладний рівень. По горизонталі подано параметри: ймовірність настання події P, вплив події I, вага рівня W, коефіцієнт критичності моделі ICM, а також підсумковий внесок рівня у загальний ризик (Внесок). Колірна шкала відповідає величині значення: темні відтінки позначають нижчі показники, а яскраві жовті тони – найвищі. Числові значення всередині клітинок дозволяють одночасно оцінити і візуальну інтенсивність параметра, і його точне значення. Інтерпретація: Найбільший внесок у ризик (0.486) має рівень Мережа, який поєднує підвищену ймовірність (0.15), значний вплив (9) і вагу (0.30) з підвищеним коефіцієнтом критичності (1.2). Рівень Обчислення має найбільший вплив (10) та критичність (1.5), проте менша ймовірність (0.10) обмежує його внесок до 0.375 . Рівні Сенсори та Прикладний мають

менші внески (0.4 та 0.066 відповідно), хоча сенсорний рівень залишається важливим через високий вплив (8) та вагу (0.25).

Висновок: Візуалізація дозволяє швидко визначити пріоритетні напрями зниження ризику, що у цьому випадку – оптимізація безпеки мережевого рівня та підвищення стійкості обчислювальних модулів.

Аналіз чутливості

є важливою складовою оцінювання ризиків у кіберфізичних системах з інтегрованими штучними нейронними мережами. Такий аналіз дозволяє зрозуміти, як зміна ключових параметрів системи впливає на загальний рівень ризику. Зокрема, важливо дослідити, чи має більший вплив імовірність настання подій або величина їхнього впливу на роботу системи. Отримані результати допомагають визначити пріоритети для впровадження заходів безпеки та оптимізації роботи системи.

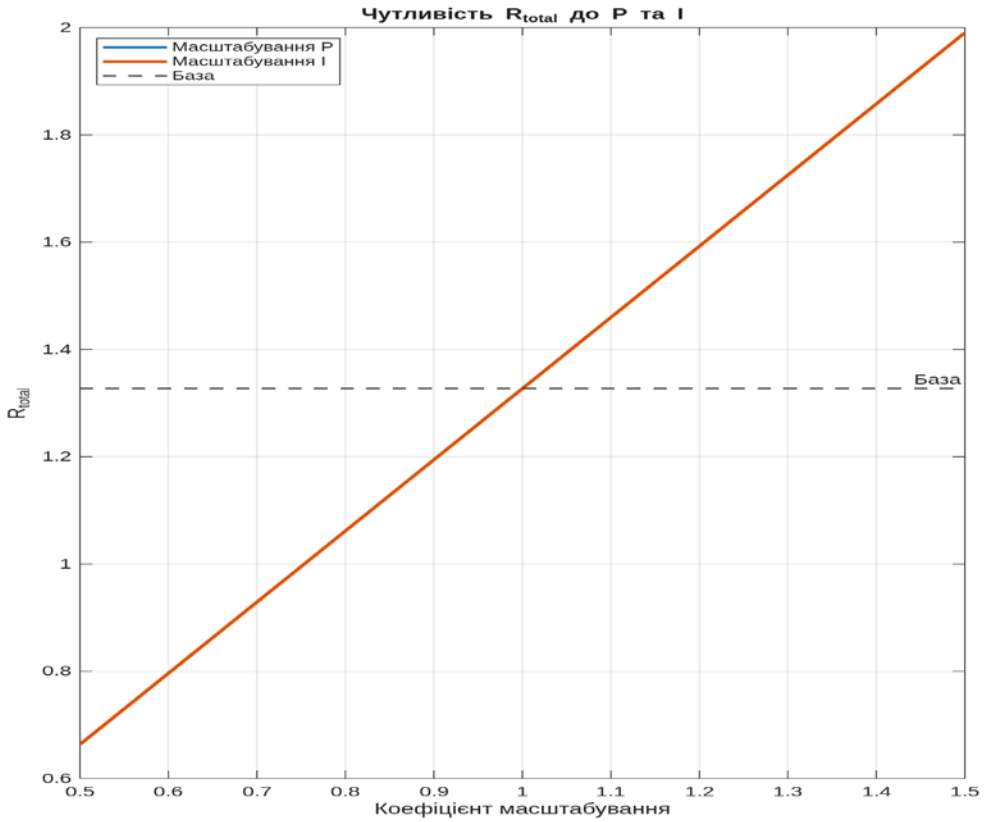


Рис. 2. На графіку по горизонталі відкладено коефіцієнт масштабування параметрів у діапазоні від 0.5 до 1.5, що відповідає зміні їхніх значень на ±50% від базового рівня. По вертикалі подано значення інтегрального ризику системи.

Синя лінія відображає зміну ризику при масштабуванні лише ймовірностей настання подій, тоді як помаранчева – зміну ризику під час

масштабування величини їхнього впливу. Чорна пунктирна лінія відповідає базовому значенню ризику без масштабування параметрів.

Видно, що залежність ризику від обох параметрів є лінійною у розглянутому діапазоні, а зміна будь-якого з них на однаковий відсоток призводить до однакової зміни загального ризику. Це свідчить про симетричний вплив параметрів імовірності та впливу на кінцевий результат, що, у свою чергу, вказує на доцільність одночасної оптимізації обох факторів для ефективного зниження ризику.

Сценарний аналіз

«що, якщо» є інструментом оцінки стійкості кіберфізичних систем до змін у ключових параметрах, що впливають на загальний рівень ризику. Такий підхід дозволяє дослідити, як цілеспрямовані зміни у конфігурації системи або її окремих елементів можуть знизити або підвищити ризик. Аналіз використовується для прийняття обґрунтованих рішень щодо оптимізації роботи системи та визначення пріоритетних напрямів захисту.

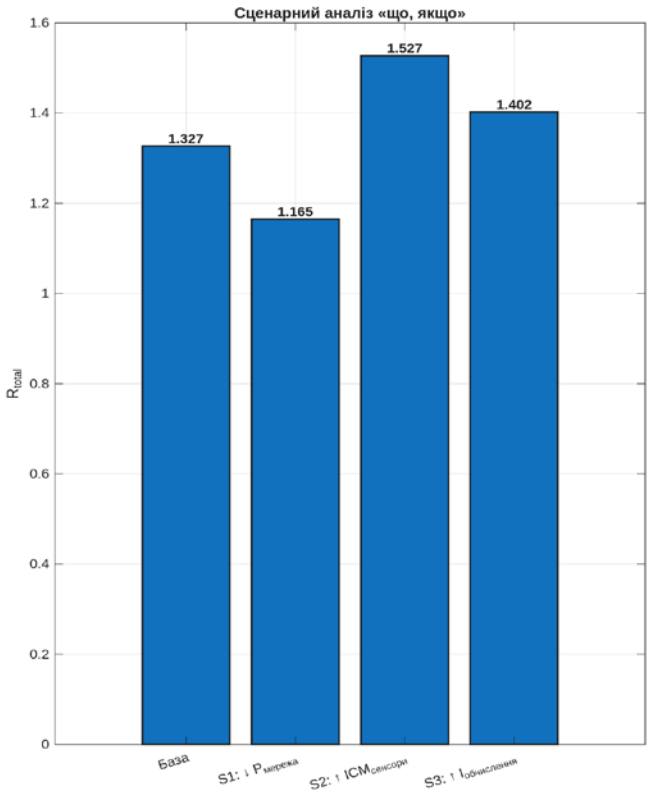


Рис. 3. На графіку представлено чотири стовпці, що відображають інтегральний ризик системи в базовому сценарії та трьох альтернативних сценаріях зміни параметрів:

Базовий сценарій – відображає початковий розрахований рівень ризику системи без змін параметрів.

Сценарій S1 – передбачає зменшення ймовірності інцидентів на мережевому рівні, що призводить до зниження ризику порівняно з базовим варіантом.

Сценарій S2 – включає підвищення коефіцієнта критичності сенсорного рівня, що спричиняє найбільше зростання ризику серед усіх розглянутих сценаріїв.

Сценарій S3 – ураховує збільшення впливу інцидентів на обчислювальному рівні, що також призводить до зростання ризику, але меншого, ніж у сценарії S2.

Візуальне представлення у вигляді стовпчастої діаграми дає змогу швидко оцінити, які зміни в конфігурації системи найбільше впливають на рівень ризику, а також визначити потенційні напрями для його зменшення.

Аналіз внесків окремих рівнів

кіберфізичної системи у загальний рівень ризику дає змогу визначити, які елементи інфраструктури найбільш критичні для стабільної та безпечної роботи системи. Такий підхід є основою для формування пріоритетів у впровадженні заходів з підвищення надійності та зменшення вразливостей.

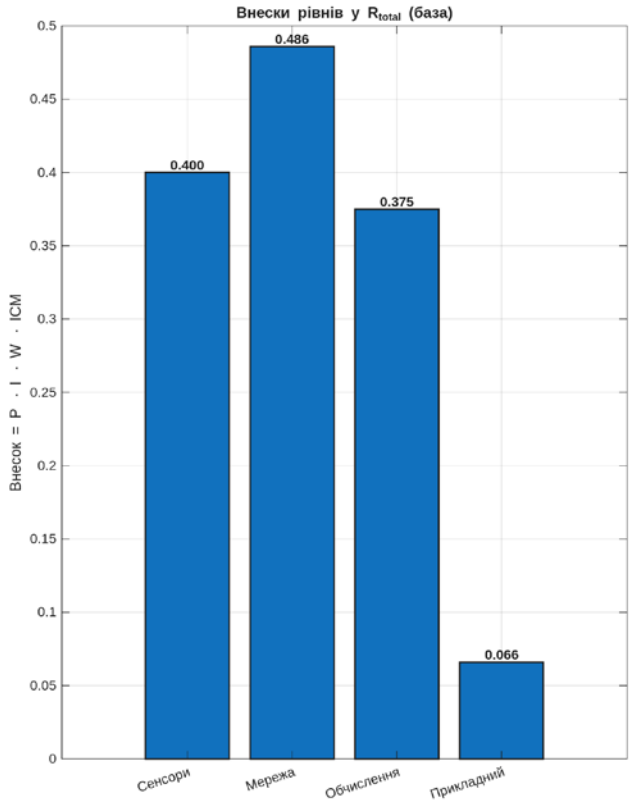


Рис. 4. На стовпчастій діаграмі показано внесок кожного з чотирьох рівнів системи у загальний ризик у базовій конфігурації:

Сенсори – внесок становить 0.400, що свідчить про значну роль сенсорного рівня у формуванні ризику, зумовлену комбінацією відносно високого впливу та ваги.

Мережа – найбільший внесок серед усіх рівнів (0.486), що пояснюється поєднанням підвищеної ймовірності інцидентів, значного впливу та високої ваги у системі.

Обчислення – внесок становить 0.375, що пов'язано з найбільшим впливом та критичністю, але нижчою ймовірністю інцидентів порівняно з мережевим рівнем.

Прикладний рівень – найменший внесок (0.066), зумовлений низькою ймовірністю інцидентів і меншою вагою в загальній архітектурі.

Візуалізація чітко демонструє, що для зниження інтегрального ризику передусім слід зосередитися на оптимізації захисту мережевого та сенсорного рівнів, а також підвищенні стійкості обчислювальних модулів.

У межах підрозділу 3.1 було встановлено, що кіберфізичні системи являють собою інтегровані комплекси, в яких фізичні, обчислювальні та комунікаційні компоненти взаємодіють у режимі реального часу, забезпечуючи виконання критично важливих функцій. Архітектура таких систем передбачає наявність кількох рівнів, серед яких сенсорний, мережевий, обчислювальний та прикладний, кожен з яких має власну роль у підтриманні безпеки, надійності та функціональності.

Аналіз засвідчив, що роль рівнів у загальному рівні ризику відрізняється залежно від їхніх характеристик і взаємозв'язків. Штучні нейронні мережі у складі КФС виконують функції інтелектуальної обробки великих обсягів даних, прогнозування поведінки системи, виявлення аномалій та автоматизації прийняття рішень. Використання ШНМ забезпечує підвищення адаптивності та ефективності функціонування системи, але водночас призводить до появи нових викликів, пов'язаних із уразливістю моделей до цілеспрямованих атак, залежністю результатів від якості навчальних даних та складністю пояснення логіки роботи моделей. Дослідження особливостей ризик-менеджменту в КФС із ШНМ підтвердило необхідність комплексного підходу, що охоплює методи ідентифікації, оцінювання та зменшення ризиків на всіх рівнях архітектури.

Розглянуті міжнародні методології, зокрема CRAMM, MAGERIT, OCTAVE, ISO/NIST, дають можливість стандартизувати процес управління ризиками, проте вони потребують адаптації з урахуванням специфіки застосування нейромережевих технологій у багаторівневих архітектурах. Запропонована формула зваженої оцінки ризику з урахуванням коефіцієнта критичності моделі дозволяє отримати кількісні оцінки внеску кожного рівня у загальний ризик і визначити пріоритети для підвищення безпеки системи. Виконаний візуалізаційний аналіз, що включав теплову карту параметрів, графік чутливості інтегрального ризику, сценарний аналіз і діаграму внесків засвідчив, що найбільшу увагу слід приділяти захисту мережевого та обчислювального рівнів. Результати аналізу свідчать про симетричний вплив ймовірностей інцидентів і величини їхнього впливу на інтегральний ризик, що

вказує на доцільність поєднання заходів, спрямованих як на зменшення ймовірності подій, так і на зниження їхнього впливу. Запропоновано застосовувати регулярний моніторинг показників ризику із залученням автоматизованих систем на базі ШНМ, що дозволить своєчасно виявляти зміни в конфігурації ризиків і швидко реагувати на них.

Отримані результати створюють концептуальне та методичне підґрунтя для подальшої систематизації термінів і понять у сфері кіберфізичних систем і нейромережевих рішень, що є предметом наступного підрозділу 3.2, присвяченого класифікації термінів.

3.2. Класифікація термінів

Розвиток кіберфізичних систем (КФС) і технологій штучних нейронних мереж (ШНМ) супроводжується стрімким зростанням термінологічного апарату, що використовується в наукових дослідженнях, промислових проєктах, нормативно-правових документах та стандартах [122]. Систематизація та уніфікація цієї термінології є необхідною передумовою для узгодження вимог до проєктування, впровадження та експлуатації систем, а також для забезпечення єдиної інтерпретації понять у міжнародному співробітництві [123], [124]. У межах цього дослідження класифікація термінів здійснюється за функціональною ознакою та сферою застосування, що дозволяє виділити кілька основних категорій: базові поняття, структурно-архітектурні терміни, методологічні та стандартизаційні терміни, а також міждисциплінарні визначення [125]. Такий підхід забезпечує цілісність термінологічної системи та полегшує інтеграцію КФС у суміжні галузі, включно з інформаційною безпекою, автоматизацією виробництва та штучним інтелектом [126].

Базові поняття кіберфізичних систем

До базових понять, що формують основу термінології КФС, належать:

Кіберфізична система – інтегрований комплекс апаратних, програмних і комунікаційних компонентів, у якому фізичні процеси контролюються та управляються за допомогою обчислювальних ресурсів, об’єднаних у мережу [127].

Цифровий двійник (digital twin) **–** віртуальна модель фізичного об’єкта або процесу, що використовується для моніторингу, прогнозування, оптимізації та аналізу сценаріїв функціонування [128].

Інтернет речей (IoT) – концепція об’єднання фізичних пристроїв у мережу з можливістю автоматичного обміну даними й взаємодії між собою та з іншими системами [129].

Використання цих базових термінів забезпечує фундамент для опису більш складних концепцій та технологічних рішень, включно з інтеграцією ШНМ у процеси керування та моніторингу [130].

Базові поняття штучних нейронних мереж

Термінологія ШНМ охоплює широкий спектр понять, пов'язаних із архітектурою, принципами функціонування, методами навчання та сферами застосування [131]. Ключовими з них є:

Штучна нейронна мережа – математична модель, що імітує принципи роботи біологічних нейронних мереж, здатна до навчання на основі даних і узагальнення знань [132].

Архітектура нейронної мережі – сукупність структурних параметрів мережі, що визначають кількість шарів, типи нейронів, способи з'єднання та функції активації (MLP, CNN, RNN тощо) [133].

Глибоке навчання (deep learning) – підхід до побудови й навчання ШНМ із великою кількістю прихованих шарів, що дає змогу моделювати складні залежності та структури у вхідних даних [134]. Знання цих термінів є необхідним для опису алгоритмічних компонентів КФС, побудови прогнозних моделей і систем виявлення аномалій [135].

Структурно-архітектурні терміни КФС

Структурна класифікація термінів охоплює визначення, що відображають фізичну та логічну організацію компонентів системи:

Сенсорний рівень – підсистема збору даних, що включає датчики, сенсори та вимірювальні пристрої [136].

Мережевий рівень – інфраструктура зв'язку та передавання даних, що забезпечує обмін інформацією між усіма компонентами КФС [137].

Обчислювальний рівень – сукупність апаратних і програмних засобів для обробки та зберігання даних, включно з хмарними й периферійними обчисленнями [138].

Прикладний рівень – набір програмних сервісів і додатків, що реалізують функції управління, моніторингу та підтримки прийняття рішень [139].

Відокремлення цих рівнів у термінології уможливлює точніше визначати зони відповідальності та ризики для кожного сегмента системи [140].

Методологічні терміни управління ризиками

З огляду на те, що управління ризиками є невід'ємною частиною функціонування КФС, до термінології входять визначення, які описують процеси ідентифікації, оцінки та контролю ризиків:

Ідентифікація ризиків – процес виявлення та опису потенційних загроз для системи з урахуванням їхніх джерел і можливих наслідків [141].

Оцінювання ризику – визначення ймовірності настання події та масштабу її впливу на систему [142].

Міграція ризиків – розроблення та впровадження заходів, спрямованих на зменшення ймовірності чи впливу небажаних подій [143].

Ці терміни використовуються у міжнародних стандартах, зокрема ISO 31000 та IEC 62443, що забезпечує узгодженість у міжнародному співробітництві [144].

У сучасних умовах цифрової трансформації кіберфізичні системи (КФС) виступають ключовим елементом інтеграції фізичного та віртуального середовищ, забезпечуючи взаємодію апаратних, програмних та комунікаційних компонентів у режимі реального часу. Їхня архітектура поєднує сенсорні технології, мережеву інфраструктуру, обчислювальні ресурси та прикладні сервіси, що створює єдине середовище керування складними технічними об'єктами. Ефективне проєктування та експлуатація КФС неможливі без чіткого термінологічного апарату та формалізованої моделі знань, що описує базові поняття та взаємозв'язки між ними. Одним із найбільш наочних інструментів такої формалізації є онтологічна карта, яка дозволяє візуально представити концепти, сутності та процеси у вигляді структурованої графічної моделі.

Запропонована онтологічна карта сформована на основі базових термінів – «Кіберфізична система» [127], «Цифровий двійник (digital twin)» [128] та «Інтернет речей (IoT)» [129], доповнених сучасним концептом «Штучні нейромережі (ШНМ)» [130], що забезпечують розширену аналітику та інтелектуальне керування. В основу побудови карти покладено багаторівневу структуризацію КФС, яка включає чотири основні рівні: фізичний, мережевий, обчислювальний та прикладний. Такий підхід дозволяє чітко розмежувати роль кожного елемента, визначити напрямки інформаційних потоків та оптимізувати інтеграційні процеси між підсистемами.

Запропонована модель є універсальною основою для подальшої деталізації у конкретних прикладних доменах — від промислової автоматизації та «розумних» міст до кіберзахисту критичної інфраструктури. Вона може використовуватися як навчальний, методологічний та інженерний інструмент для опису та впровадження КФС.

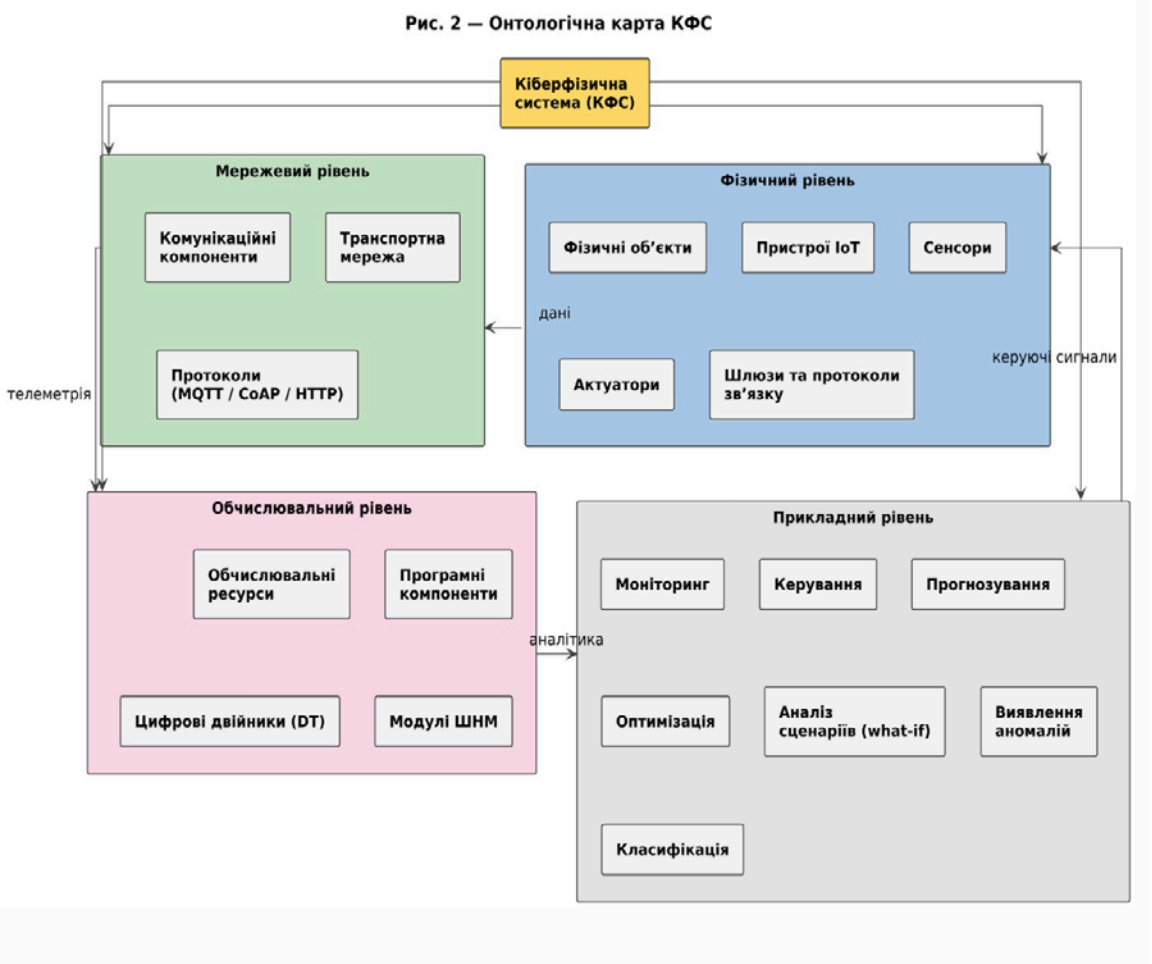


Рис. 2. У центрі онтологічної карти розташовано поняття «Кіберфізична система (КФС)», яке інтегрує фізичні процеси, апаратні засоби, програмні модулі та мережеву інфраструктуру.

Від КФС відгалужуються структурні складові, що належать до різних архітектурних рівнів, кожен із яких виконує чітко визначені функції у загальній структурі.

Фізичний рівень представлений фізичними об'єктами та пристроями IoT, які містять сенсори для збору даних та актуатори для виконання команд. Ці елементи взаємодіють із мережевою інфраструктурою через шлюзи та протоколи зв'язку, забезпечуючи безперервний цикл «збір даних – передача – керування».

Мережевий рівень включає комунікаційні компоненти, транспортну мережу та набір протоколів (MQTT, CoAP, HTTP тощо), які підтримують обмін даними між фізичними пристроями та обчислювальними модулями. Тут формується комунікаційний каркас КФС, що визначає пропускну здатність, затримки та надійність обміну. Обчислювальний рівень складається з обчислювальних ресурсів і програмних компонентів, серед яких центральне місце належить цифровим двійникам (DT) та модулям

штучних нейромереж. DT забезпечує моделювання стану фізичних об’єктів у реальному часі, тоді як ШНМ виконують функції прогнозування, оптимізації, виявлення аномалій і класифікації станів.

Прикладний рівень представлений наборами процесів: моніторингом, керуванням, прогнозуванням, оптимізацією, аналізом сценаріїв (what-if), виявленням аномалій та класифікацією. Взаємодія між рівнями організована у вигляді спрямованих зв’язків, що відображають потоки даних та керуючі сигнали, забезпечуючи замкнутий цикл функціонування КФС із можливістю адаптивного розвитку та інтеграції нових технологій.

Значення уніфікації термінів

Єдина термінологічна база сприяє підвищенню ефективності комунікацій між учасниками проєкту, знижує ризик різночитань і помилкових інтерпретацій, полегшує інтеграцію систем різних виробників [145], [146]. Для КФС з інтегрованими ШНМ уніфікація термінів має подвійне значення: вона одночасно стандартизує опис технічних компонентів і алгоритмічних рішень, а також створює основу для сертифікації та перевірки безпеки систем [147], [148].

Стандартизаційні терміни у сфері КФС та ШНМ

Важливим напрямом класифікації є терміни, закріплені у міжнародних та національних стандартах, які визначають вимоги до проєктування, експлуатації та захисту кіберфізичних систем. Такі терміни формуються в рамках діяльності організацій, що займаються стандартизацією, зокрема ISO, IEC, IEEE, NIST, ENISA, ITU [122], [123].

Серед ключових можна виділити:

ISO 31000 – міжнародний стандарт з управління ризиками, що встановлює принципи, структуру та процеси для систем будь-якого типу, включно з КФС [124].

IEC 62443 – набір стандартів, що визначає вимоги до кібербезпеки промислових автоматизованих систем керування [125].

ISO/IEC 27001 – стандарт, що описує вимоги до систем управління інформаційною безпекою [126].

IEEE 802.15.4 – стандарт для бездротових мереж малого радіусу дії, що використовується у сенсорних мережах КФС [127].

NIST SP 800-53 – рекомендації з безпеки інформаційних систем федеральних органів США, адаптовані для застосування у промислових і критичних інфраструктурах [128].

Застосування таких термінів у наукових і технічних документах забезпечує відповідність проєктних рішень міжнародним вимогам та підвищує сумісність між різними системами [129].

Абревіатури та скорочення у термінології

У сучасних науково-технічних текстах, присвячених КФС і ШНМ, широко використовуються абревіатури, які мають бути однозначно трактовані [130]. Класифікація абревіатур може здійснюватися за сферою походження (стандарти, технології, організації, протоколи, алгоритми).

Приклади поширених абревіатур:

CPS (Cyber-Physical System) – кіберфізична система [131].

IoT (Internet of Things) – Інтернет речей [132].

SCADA (Supervisory Control and Data Acquisition) – система диспетчерського керування і збору даних [133].

PLC (Programmable Logic Controller) – програмований логічний контролер [134].

IDS/IPS (Intrusion Detection/Prevention System) – системи виявлення та запобігання вторгненням [135].

CNN (Convolutional Neural Network) – згорткова нейронна мережа [136].

LSTM (Long Short-Term Memory) – нейронна мережа з довгою короткочасною пам’яттю [137].

GAN (Generative Adversarial Network) – генеративно-змагальна мережа [138].

Чітке визначення таких скорочень у термінологічних словниках чи глосаріях є необхідним для уникнення неоднозначностей, особливо у міжгалузевих документах [139].

Функціональні категорії термінів

Функціональні категорії описують ролі та призначення термінів у рамках КФС з інтегрованими ШНМ. Вони поділяються на:

Операційні терміни – пов’язані з безпосередньою роботою системи, наприклад, «цифровий датчик температури», «система моніторингу в реальному часі» [140].

Аналітичні терміни – описують методи та інструменти обробки даних, наприклад, «класифікація аномалій», «предиктивна аналітика» [141].

Захисні терміни – характеризують процеси і засоби кіберзахисту, такі як «автентифікація багатьма факторами» або «сегментація мережі» [142].

Інтеграційні терміни – позначають процеси взаємодії між різними компонентами, наприклад, «протокол обміну повідомленнями MQTT», «інтерфейс прикладного програмування API» [143].

Класифікація за функціональними категоріями допомагає структурувати терміни для навчальних програм, технічної документації та наукових публікацій [144].

Зв'язок термінів з процесами стандартизації

Більшість термінів, що застосовуються у КФС та ШНМ, мають відповідники у міжнародних та національних стандартах [145]. Наприклад, терміни, що описують рівні безпеки, напряму пов'язані з класифікаціями, визначеними в IEC 62443, а терміни, що описують архітектурні моделі, узгоджуються з ISO/IEC 30141 [146]. Застосування стандартизованої термінології забезпечує не лише технічну, але й правову сумісність систем, оскільки дозволяє однозначно тлумачити вимоги в контексті контрактів, ліцензій та аудитів безпеки [147]. У наукових дослідженнях це спрощує порівняння результатів та інтеграцію нових рішень у вже існуючі технологічні платформи [148].

Безпекові терміни у сфері КФС та ШНМ

У контексті кіберфізичних систем, особливо тих, що інтегрують штучні нейронні мережі, безпекова термінологія має особливе значення, оскільки вона визначає базові підходи до забезпечення цілісності, конфіденційності та доступності системи [122]. До основних термінів належать:

Аутентифікація – процес перевірки автентичності користувача або пристрою, що намагається отримати доступ до системи [123].

Авторизація – процес надання прав доступу до ресурсів системи відповідно до попередньо визначених політик [124].

Кіберзагроза – потенційна подія або дія, спрямована на порушення нормального функціонування системи або компрометацію її даних [125].

Вразливість – слабке місце в системі або її компонентах, яке може бути використане зловмисником для здійснення атаки [126].

Інцидент безпеки – подія, яка призвела або може призвести до порушення конфіденційності, цілісності чи доступності даних [127].

Система виявлення вторгнень (IDS) – програмно-апаратний комплекс для моніторингу подій у системі з метою виявлення потенційних атак [128].

Система запобігання вторгнень (IPS) – система, здатна не лише виявляти атаки, але й автоматично вживати заходів для їхнього блокування [129]. Важливим є те, що більшість із цих термінів мають формалізовані визначення в міжнародних стандартах, зокрема в ISO/IEC 27000 та IEC 62443 [130]. Це забезпечує їх однакове тлумачення в різних країнах і компаніях.

Міждисциплінарні терміни

Сучасні КФС із ШНМ функціонують на стику кількох галузей – інформаційних технологій, промислової автоматизації, кібербезпеки, штучного інтелекту, телекомунікацій, а іноді й біомедичних наук. Це породжує міждисциплінарну термінологію [131].

Приклади таких термінів:

Телемедицина – використання КФС і телекомунікаційних технологій для дистанційної діагностики та лікування пацієнтів [132].

Розумне місто (Smart City) – комплекс інфраструктурних рішень на базі КФС та IoT, спрямованих на підвищення ефективності управління міськими ресурсами [133].

Інтелектуальна енергосистема (Smart Grid) – енергетична мережа, що інтегрує КФС і алгоритми ШНМ для оптимізації виробництва, розподілу та споживання енергії [134].

Автономний транспорт – транспортні засоби, керовані КФС з використанням ШНМ для обробки даних від сенсорів і камер у режимі реального часу [135].

Міждисциплінарні терміни часто мають подвійний чи навіть потрійний контекст, оскільки описують як технічні, так і соціально-економічні аспекти впровадження технологій [136].

Галузеві приклади класифікації термінів

У різних галузях КФС з інтегрованими ШНМ застосовуються специфічні терміни, що відображають особливості їхнього використання:

Медицина: електронна медична картка (EHR), система підтримки прийняття клінічних рішень (CDSS), медичний Інтернет речей (MIoT) [137].

Транспорт: інтелектуальна транспортна система (ITS), система керування дорожнім рухом (TMS), Vehicle-to-Everything (V2X) [138].

Енергетика: автоматизована система обліку електроенергії (ACOE), розподілене керування генерацією (DER), управління навантаженням (DSM) [139].

Промисловість: виробництво з нульовими зупинками (Zero Downtime Manufacturing), предиктивне технічне обслуговування (Predictive Maintenance), система керування виробничими процесами (MES) [140].

Кожен із цих термінів входить до окремих класифікаційних блоків галузевих стандартів, що дозволяє чітко відокремити їх значення у конкретному контексті [141].

Переваги класифікації термінів для КФС із ШНМ

Чітка класифікація термінів забезпечує низку переваг:

полегшує інтеграцію нових технологій у вже існуючі системи [142];

підвищує ефективність комунікацій між інженерами, аналітиками та керівниками проєктів [143];

зменшує ризик помилкової інтерпретації технічної документації [144];

забезпечує узгодженість при міжнародному співробітництві та сертифікації рішень [145];

створює основу для формування навчальних програм і стандартів підготовки кадрів [146].

Проведена класифікація термінів у сфері кіберфізичних систем з інтегрованими штучними нейронними мережами показала, що термінологічна система є багаторівневою, охоплює як базові технічні поняття, так і стандартизаційні, безпекові, міждисциплінарні та галузеві терміни. Такий підхід дозволяє підвищити точність і ефективність комунікацій між усіма учасниками життєвого циклу системи — від розробників і науковців до кінцевих користувачів та регуляторів. Розглянуті приклади показують важливість уніфікації термінів для узгодження технічних рішень із міжнародними стандартами та спрощення процесу впровадження інноваційних рішень у різні галузі. Отримані результати створюють методичне підґрунтя для наступного підрозділу 3.3, у якому буде розглянуто стандартизацію термінології та механізми її гармонізації у міжнародному та національному контексті.

3.3. Стандартизація термінології

Стандартизація термінології у сфері кіберфізичних систем з інтегрованими штучними нейронними мережами є ключовою передумовою забезпечення міжгалузевої та міжнародної сумісності рішень, а також ефективної комунікації між усіма учасниками процесу розроблення, впровадження та експлуатації таких систем. Наявність уніфікованих визначень дозволяє уникати різночитань, які можуть призводити до суттєвих технічних і правових помилок під час створення та інтеграції інфраструктурних елементів [122].

Процес стандартизації термінів передбачає розроблення, затвердження та регулярне оновлення словників, глосаріїв і довідників, у яких наводяться узгоджені визначення та тлумачення термінів у контексті міжнародних і національних стандартів [123]. У сфері КФС та ШНМ провідну роль у формуванні термінологічної бази відіграють міжнародні організації зі стандартизації, такі як Міжнародна організація зі стандартизації (ISO), Міжнародна електротехнічна комісія (IEC), Інститут інженерів з електротехніки та електроніки (IEEE), Національний інститут стандартів і технологій США (NIST), Міжнародний союз електрозв’язку (ITU) та Європейське агентство з кібербезпеки (ENISA) [124]. ISO розробляє стандарти, що охоплюють широкий спектр галузей, включаючи управління інформаційною безпекою, управління ризиками та стандартизацію термінології у сфері інформаційних технологій. Для КФС із ШНМ особливе значення мають стандарти ISO/IEC 27000, що визначають терміни та визначення у сфері інформаційної безпеки, ISO 31000, який регламентує принципи та керівні положення управління ризиками, а також ISO/IEC 30141, що описує еталонну архітектуру Інтернету речей, до якої часто інтегруються компоненти КФС [125]. IEC

відповідає за стандартизацію термінів, що стосуються електротехнічних та промислових автоматизованих систем.

Важливим прикладом є серія стандартів IEC 62443, яка встановлює вимоги до кібербезпеки промислових систем керування, в тому числі шляхом уніфікації термінів, пов'язаних із ролями користувачів, рівнями захисту та типами загроз [126]. IEEE розробляє технічні стандарти, що охоплюють терміни у сфері мережевих технологій, комунікаційних протоколів, сенсорних мереж, а також архітектур і алгоритмів штучних нейронних мереж. Наприклад, стандарти IEEE 802.15.4 регламентують роботу бездротових персональних мереж, у тому числі терміни, пов'язані з типами вузлів, топологіями та методами керування доступом [127]. NIST відіграє провідну роль у розробленні рекомендацій та глосаріїв термінів для кібербезпеки, що широко застосовуються у промислових і критичних інфраструктурах. Документ NIST SP 800-53 містить не лише вимоги до захисту систем, але й стандартизований перелік понять, які використовуються при описі контролів безпеки, категорій загроз і вразливостей [128]. ITU, як спеціалізована агенція ООН, зосереджується на стандартизації термінів, пов'язаних з інформаційно-комунікаційними технологіями, включно з мережами наступного покоління, хмарними обчисленнями та Інтернетом речей. ITU-T розробляє рекомендації, у яких терміни мають чіткі визначення, що забезпечує однакове їх трактування в країнах-учасниках [129]. ENISA, зі свого боку, пропонує європейський підхід до термінології кібербезпеки, що враховує особливості правових режимів країн ЄС і гармонізує терміни з міжнародними стандартами [130]. Стандартизація термінології у сфері КФС та ШНМ не обмежується лише технічними аспектами. Вона охоплює також юридичні та організаційні терміни, пов'язані з регулюванням доступу, відповідальністю сторін, вимогами до сертифікації та аудиту.

Так, у рамках ISO/IEC 27001 і супутніх стандартів ISO визначаються терміни, що описують рівні класифікації інформації, ролі користувачів та адміністраторів, процедури управління інцидентами [131]. У IEC 61508, присвяченому функціональній безпеці, закріплено визначення, які стосуються надійності та відмовостійкості систем, що є важливими при проєктуванні КФС з критичною інфраструктурою [132]. Важливим елементом процесу стандартизації термінів є створення та підтримка оновлюваних глосаріїв, які доступні у відкритому доступі. Прикладом є глосарій NIST, що постійно доповнюється новими термінами та їхніми визначеннями відповідно до змін у нормативно-технічній базі [133]. Подібний підхід застосовує і ISO, випускаючи технічні звіти та посібники з термінології, які використовуються як довідкові матеріали для міжнародних проєктів [134].

У межах ЄС функціонує кілька централізованих платформ для управління термінами, де зібрано стандартизовані визначення, прийняті на

рівні європейських стандартів, що полегшує їх впровадження у національні документи [135].

Таким чином, уніфікація термінології на міжнародному рівні створює основу для ефективної інтеграції кіберфізичних систем із різних країн та галузей, забезпечує юридичну сумісність рішень і підвищує якість управління безпекою та ризиками. Подальший розвиток цієї сфери передбачає більш тісну координацію між організаціями зі стандартизації та науковою спільнотою з метою актуалізації термінів відповідно до швидких технологічних змін [136].

Стандартизація термінології у сфері кіберфізичних систем і штучних нейронних мереж охоплює комплекс міжнародних документів, у яких визначаються ключові поняття, принципи побудови та опису систем, а також вимоги до їхньої безпеки та взаємодії. Одним із фундаментальних документів є серія стандартів ISO/IEC 27000, у якій сформовано глосарій інформаційної безпеки, що включає визначення термінів, пов’язаних із загрозами, вразливостями, інцидентами, контролями та управлінням ризиками [122]. Цей глосарій використовується не лише у межах серії стандартів, але також інтегрується в інші міжнародні документи, наприклад, у рекомендації ITU-T з кібербезпеки та в методичні матеріали ENISA, де терміни зберігають однакове значення [123].

Не менш важливим для термінологічної єдності є ISO 31000, що встановлює принципи та загальні настанови з управління ризиками. У цьому стандарті наводяться визначення таких понять, як «ризик», «оцінювання ризику», «лікування ризику» та «критерії ризику», які безпосередньо використовуються у сфері КФС та ШНМ під час побудови моделей аналізу загроз і виборі заходів безпеки [124]. Визначення, закріплені у ISO 31000, застосовуються як у суто технічних контекстах, так і під час опису організаційних процесів, що підтверджує універсальний характер цих термінів. Серія стандартів IEC 62443, присвячена кібербезпеці промислових систем автоматизації та управління, містить детальний перелік термінів, які охоплюють як технічні, так і організаційні аспекти безпеки. Тут наведено визначення ролей користувачів і адміністраторів, типів загроз, рівнів безпеки, процесів аутентифікації та авторизації, а також терміни, що описують процеси безпечної інтеграції компонентів систем [125]. У межах цієї серії стандарти розробляються в тісній координації з ISO, що дозволяє уникати дублювання й забезпечує узгодженість термінів у суміжних галузях [126].

Особлива роль у стандартизації термінів для комунікаційних протоколів і сенсорних мереж належить стандартам IEEE. Скажімо, IEEE 802.15.4 визначає терміни, що описують типи пристроїв, формати кадрів, методи керування доступом і топології мереж. Ці терміни є основою для специфікацій промислових протоколів, які використовуються в КФС, таких як WirelessHART або Zigbee, де додатково вводяться розширення для конкретних галузей застосування [127]. Гармонізація термінів між

IEEE та іншими організаціями, зокрема IETF, дозволяє досягти високого рівня сумісності протоколів і забезпечити їх коректну інтеграцію в багаторівневу архітектуру КФС [128].

Документи NIST, особливо серія SP 800, містять детально розроблені глосарії термінів для кібербезпеки та управління інформаційними системами. NIST SP 800-53 є прикладом такого документа, де терміни узгоджені з ISO/IEC 27000, але адаптовані для потреб американських державних і критичних інфраструктур [129]. У цих глосаріях велика увага приділяється термінам, що описують заходи контролю доступу, безпеку мереж, захист від шкідливого програмного забезпечення та безперервність бізнесу. Завдяки відкритості цих матеріалів вони широко використовуються у приватному секторі та в освітніх програмах [130].

ITU-T у своїх рекомендаціях, зокрема у серії X.1200 і X.1500, визначає терміни, пов'язані з кіберзагрозами, кіберзахистом, реагуванням на інциденти, а також з архітектурними моделями кібербезпеки. Ці документи активно узгоджуються з ISO, IEC та IETF, що дає можливість забезпечити глобальну інтероперабельність термінів [131]. ENISA, у свою чергу, публікує європейські глосарії з кібербезпеки, де терміни гармонізовані з міжнародними стандартами, але враховують особливості регуляторних актів ЄС, таких як Директива NIS та Загальний регламент із захисту даних (GDPR) [132].

Процеси гармонізації термінології між країнами і галузями передбачають регулярне оновлення глосаріїв, створення спільних робочих груп і проведення публічних консультацій. Прикладом такої співпраці є ініціатива ISO/IEC JTC 1, де представники різних країн працюють над єдиними визначеннями термінів у сфері інформаційних технологій, у тому числі тих, що стосуються КФС і ШНМ [133]. У межах цих робіт часто виявляються конфлікти термінів, коли одне й те ж поняття має різне значення в різних галузях або країнах. Наприклад, термін «resilience» у контексті кібербезпеки часто визначається як здатність системи швидко відновлюватися після інциденту, тоді як у стандартах з надійності під «resilience» розуміють здатність системи продовжувати функціонування навіть за умов впливу загроз [134]. Вирішення таких конфліктів відбувається шляхом внесення уточнень у визначення або додавання контекстуальних позначок. У стандартах ISO та IEC часто застосовується практика створення окремих пунктів «Примітка», де пояснюється, у якому саме значенні в даному документі використовується певний термін [135]. Це дозволяє зберегти єдність базового значення терміна і водночас врахувати специфіку галузі.

Важливим аспектом гармонізації є також переклад термінів. У міжнародних стандартах, що публікуються кількома мовами, переклад виконується фахівцями з технічної термінології, а затвердження відбувається на рівні національних органів стандартизації. При цьому

забезпечується збереження смислової точності та уникнення неоднозначностей, які можуть виникнути при дослівному перекладі [136].

У сфері КФС і ШНМ це особливо важливо, оскільки некоректний переклад терміна, наприклад, у специфікації протоколу або в технічному завданні, може призвести до несумісності реалізацій або помилок у налаштуваннях системи [137]. Таким чином, стандартизація термінології на міжнародному рівні є результатом складної взаємодії між організаціями зі стандартизації, урядами, науковою спільнотою та промисловими компаніями. Її головна мета полягає у створенні стабільної та зрозумілої основи для розроблення, впровадження та супроводу кіберфізичних систем і технологій штучного інтелекту в усьому світі [138]. Національна адаптація міжнародних стандартів термінології у сфері кіберфізичних систем та штучних нейронних мереж є складним і багатоступеневим процесом, що потребує узгодження технічних, правових і лінгвістичних аспектів.

Кожна країна має власні органи стандартизації, які відповідають за впровадження міжнародних документів у національну систему нормативного регулювання. Ці органи здійснюють переклад, адаптацію та, в разі потреби, модифікацію термінів для відповідності локальним технічним і законодавчим реаліям [122]. В Україні ці функції виконує Національний орган стандартизації (ДП «УкрНДНЦ»), який співпрацює з ISO, IEC, IEEE та іншими міжнародними структурами. У рамках цієї співпраці здійснюється офіційний переклад стандартів, таких як ISO/IEC 27000, ISO 31000, IEC 62443, із забезпеченням збереження смислової точності термінів і водночас узгодження їх із національними галузевими нормативами [123]. Особлива увага приділяється технічним термінам, пов’язаним із безпекою критичної інфраструктури, оскільки в українському законодавстві існують специфічні визначення, які можуть відрізнятися від міжнародних інтерпретацій [124].

Адаптація стандартів часто потребує врахування національних класифікаторів та словників, що вже використовуються в суміжних галузях. Наприклад, термінологія у сфері енергетики, транспорту чи охорони здоров’я має власні традиційні визначення, які необхідно гармонізувати з новими поняттями КФС і ШНМ [125]. Це особливо актуально для міждисциплінарних термінів, які в міжнародних документах можуть мати ширший або вже усталений зміст, а в національній практиці – обмежене чи специфічне застосування [126]. Переклад термінів є критично важливим етапом, який виконується технічними експертами та лінгвістами одночасно. Неточності при перекладі можуть призвести до зміни юридичного сенсу документа або до технічної несумісності реалізованих систем [127]. Для уникнення таких помилок у національній адаптації стандартів застосовується метод подвійної перевірки, коли перекладений термін погоджується з фахівцями відповідної галузі, а потім перевіряється юридичною службою [128].

Важливим викликом для національної стандартизації є швидкий розвиток технологій, що призводить до появи нових термінів швидше, ніж відбувається формальне оновлення стандартів. Це створює часовий розрив між появою терміна в науково-технічній літературі та його закріпленням у офіційних нормативних документах [129]. Для мінімізації цього розриву деякі країни впроваджують механізми попереднього узгодження термінів через відкриті онлайн-платформи, де пропонуються нові визначення і проходять громадське обговорення [130]. Ще однією проблемою є дублювання термінів у різних стандартах, яке може призвести до неоднозначного тлумачення.

Наприклад, один і той же термін у документах ISO і IEC може мати різні контекстуальні акценти залежно від цілей стандарту [131]. Для вирішення цього завдання застосовуються крос-посилання між стандартами та спеціальні розділи з уточненнями, які пояснюють, як узгоджуються визначення у разі конфлікту [132]. Перспективи розвитку стандартизації термінології у сфері КФС і ШНМ пов’язані з активнішим залученням штучного інтелекту до процесів створення та оновлення глосаріїв. Використання автоматизованих інструментів аналізу термінів дозволяє швидше виявляти нові поняття у публікаціях, патентних заявках та технічних звітах, а також пропонувати їхнє попереднє формулювання для експертного розгляду [133].

Крім того, цифрові платформи стандартизації з можливістю багатомовної підтримки спрощують міжнародну координацію та скорочують час між появою нового терміна і його офіційним закріпленням [134]. Для України та інших країн із динамічно зростаючим IT-сектором актуальним є питання побудови національних реєстрів термінів, які синхронізуються з міжнародними глосаріями у режимі реального часу. Це уможливлює забезпечити єдність термінології в освітніх програмах, дослідницьких проєктах і промислових впровадженнях [135]. Такі реєстри можуть функціонувати на основі відкритих стандартів обміну даними, що спрощує їх інтеграцію в різні інформаційні системи та забезпечує прозорість процесу оновлення [136].

У підсумку, стандартизація термінології у сфері кіберфізичних систем та штучних нейронних мереж є безперервним процесом, що вимагає постійної координації між міжнародними організаціями, національними органами стандартизації, науковою спільнотою та промисловістю. Вона є не лише технічним завданням, а й важливим елементом інформаційної та правової безпеки, що безпосередньо впливає на якість, сумісність та стійкість створюваних систем [137]. Гармонізована термінологія дає можливість знизити ризики помилок у проєктуванні та експлуатації КФС, сприяє підвищенню рівня довіри до інноваційних рішень і створює передумови для ефективної співпраці на глобальному рівні [138].

Отже, результати дослідження свідчать, що подальший розвиток процесів стандартизації термінології у цій сфері потребує впровадження

гнучких механізмів оновлення визначень, інтеграції автоматизованих інструментів аналізу, активної участі всіх зацікавлених сторін та створення відкритих багатомовних платформ управління термінами. Це стане основою для переходу до наступного підрозділу, присвяченого абревіатурам, які становлять важливий елемент термінологічного апарату і значною мірою визначають ефективність комунікації у сфері кіберфізичних систем та штучних нейронних мереж [139].

У межах підрозділу 3.3 було здійснено комплексний аналіз процесів стандартизації термінології, що застосовується у сфері нейромережевих методів управління ризиками кіберфізичних систем (КФС). Розглянуто особливості гармонізації термінів у відповідності до міжнародних та національних стандартів, таких як ISO/IEC 27000-серії, ISO 8000, NIST SP 800-53, IEC 62443 та суміжних нормативних документів, а також галузевих глосаріїв, які визначають спільну мовну основу для розробників, інтеграторів та експлуатаційних команд. Узагальнено ключові підходи до формування єдиної термінологічної бази, зокрема: створення централізованих словників і тезаурусів; впровадження процесу верифікації нових термінів; фіксацію контекстів використання; автоматизований моніторинг появи нових дефініцій у наукових та технічних джерелах. Окремо проаналізовано проблеми полісемії та омонімії, що ускладнюють інтерпретацію технічних документів, та шляхи їх подолання шляхом контекстної прив’язки терміна до функціонального рівня КФС чи до класу задач у ризик-менеджменті. В розділі підкреслюється, що системна стандартизація термінології є фундаментом для інтероперабельності технічних рішень, коректності формалізованих вимог і автоматизованої обробки даних. Єдині терміни та визначення зменшують ризик неправильного трактування вимог безпеки, підвищують відтворюваність результатів досліджень і полегшують аудит та сертифікацію. В результаті впровадження стандартизованої терміносистеми очікується зменшення кількості конфліктів у технічних завданнях, зниження витрат на узгодження документації та пришвидшення процесів інтеграції нових компонентів у КФС.

Таким чином, результати підрозділу 3.3 створюють основу для більш предметного аналізу конкретних скорочень та абревіатур, що використовуються у практиці управління ризиками з використанням нейромережевих технологій. Саме абревіатури як компактні носії складних технічних понять часто є ключовими елементами комунікації між фахівцями. Їх неоднозначність або відсутність узгоджених тлумачень здатна призвести до помилок у проєктуванні та експлуатації систем.

Перехід до підрозділу 3.4 «Абревіатури» є логічним продовженням виконаної роботи: якщо стандартизація термінів визначає єдину основу змісту, то уніфікація абревіатур забезпечує компактну, але водночас однозначну форму їх подання в документації, інтерфейсах, протоколах і навчальних матеріалах. У цьому контексті підрозділ 3.4 деталізує методи

систематизації скорочень, формує довідник абревіатур із поясненням контекстів застосування та надає рекомендації щодо їх підтримки в актуальному стані у відповідності до оновлень терміносистеми.

3.4. Абревіатури

У сучасній науково-технічній та нормативній практиці кіберфізичних систем (КФС) та штучних нейронних мереж (ШНМ) абревіатури стали не лише засобом скорочення об’ємних термінів, але й інструментом стандартизації, що забезпечує компактність комунікацій, збереження смислової точності та швидкість обміну інформацією. У документації, програмних інтерфейсах, протоколах обміну, технічних специфікаціях і навчальних матеріалах саме абревіатури часто виступають «якорями» терміносистеми. Їх некоректне використання, двозначність або відсутність єдиного тлумачення здатні призвести до серйозних помилок у проєктуванні, впровадженні та експлуатації КФС, особливо коли йдеться про міжгалузеві та міжнародні проєкти. Важливим завданням є *уніфікація абревіатур – визначення для кожного скорочення єдиного варіанта розшифрування й контексту використання. Це завдання охоплює:*

відбір офіційно закріплених у стандартах скорочень;

формування переліку загальновживаних у професійній спільноті абревіатур;

створення нових скорочень для понять, які ще не отримали усталеної короткої форми;

запобігання дублюванню абревіатур для різних термінів у межах однієї терміносистеми.

Класифікація абревіатур у сфері КФС та ШНМ

Для впорядкування доцільно виділити кілька категорій абревіатур:

1. Міжнародні та національні стандарти

Приклади:

ISO – International Organization for Standardization (Міжнародна організація зі стандартизації).

IEC – International Electrotechnical Commission (Міжнародна електротехнічна комісія).

IEEE – Institute of Electrical and Electronics Engineers (Інститут інженерів з електротехніки та електроніки).

NIST – National Institute of Standards and Technology (Національний інститут стандартів і технологій, США).

ENISA – European Union Agency for Cybersecurity (Агентство Європейського Союзу з кібербезпеки).

DSTU – Derzhavni Standarty Ukrainy (Державні стандарти України).

2. Технологічні концепції та парадигми

CPS – Cyber-Physical System (кіберфізична система).
IoT – Internet of Things (Інтернет речей).
IIoT – Industrial Internet of Things (промисловий Інтернет речей).
AI – Artificial Intelligence (штучний інтелект).
ML – Machine Learning (машинне навчання).
DL – Deep Learning (глибинне навчання).
XAI – Explainable Artificial Intelligence (пояснюваний штучний інтелект).
ZTA – Zero Trust Architecture (архітектура нульової довіри).
DT – Digital Twin (цифровий двійник).

3. Архітектури та алгоритми ШНМ

MLP – Multilayer Perceptron (багатошаровий перцептрон).
CNN – Convolutional Neural Network (згорткова нейронна мережа).
RNN – Recurrent Neural Network (рекурентна нейронна мережа).
LSTM – Long Short-Term Memory (нейронна мережа з довгою короткочасною пам'яттю).
GRU – Gated Recurrent Unit (рекурентний блок із керованими вентилями).
AE – Autoencoder (автокодер).
VAE – Variational Autoencoder (варіаційний автокодер).
GAN – Generative Adversarial Network (генеративно-змагальна мережа)
DBN – Deep Belief Network (глибока мережа довіри).

4. Інформаційна безпека та ризик-менеджмент

ISMS – Information Security Management System (система управління інформаційною безпекою).
CSIRT – Computer Security Incident Response Team (команда реагування на комп'ютерні інциденти).
SOC – Security Operations Center (центр операційної безпеки).
IDS/IPS – Intrusion Detection/Prevention System (системи виявлення/запобігання вторгненням)
SIEM – Security Information and Event Management (система управління інформацією та подіями безпеки).
MITM – Man-in-the-Middle (атака «людина посередині»).
APT – Advanced Persistent Threat (розвинена постійна загроза).
PKI – Public Key Infrastructure (інфраструктура відкритих ключів).

Таблиця 3

Розширений глосарій абревіатур: приклади з контекстом

Абревіатура	Розшифрування	Контекст застосування
OPC UA	Open Platform Communications Unified Architecture	Промисловий стандарт взаємодії між КФС і системами автоматизації
MQTT	Message Queuing Telemetry Transport	Легковаговий протокол обміну повідомленнями у IIoT
REST	Representational State Transfer	Архітектурний стиль побудови веб-сервісів у КФС
TLS	Transport Layer Security	Захист передавання даних між компонентами КФС
HMI	Human-Machine Interface	Інтерфейс взаємодії оператора з КФС
PLC	Programmable Logic Controller	Основний елемент управління технологічними процесами
RTU	Remote Terminal Unit	Віддалений термінальний пристрій збору даних
NTP	Network Time Protocol	Синхронізація часу у розподілених КФС
API	Application Programming Interface	Інтерфейс для інтеграції програмних компонентів КФС
KPI	Key Performance Indicator	Ключові показники ефективності системи чи процесу

Приклади-пропозиції абревіатур

У сучасних проєктах КФС/ШНМ виникає потреба у скороченнях для нових концепцій, які ще не набули поширення:

RMLR – Risk-Managed Learning Rate – динамічне регулювання швидкості навчання моделі з урахуванням ризик-профілю.

E-DT – Edge Digital Twin – цифровий двійник, розгорнутий безпосередньо на периферійному вузлі для зменшення затримок.

F-GAN – Federated GAN – генеративно-змагальна мережа з федеративним навчанням для синтезу даних без їх централізації.

CPS-XAI – інтегрована платформа пояснюваного ШІ для кіберфізичних систем.

AD-SOC – Adaptive SOC – центр операційної безпеки з адаптивними моделями аналізу телеметрії.

Попередня частина окреслила базову класифікацію та приклади абревіатур, що застосовуються у КФС та ШНМ. Продовжуючи, важливо розширити глосарій за галузевим і міждисциплінарним принципом, оскільки кіберфізичні системи охоплюють широкий спектр застосувань – від енергетики та транспорту до медицини, оборонної сфери та «розумних міст». Уніфікація скорочень у таких контекстах є критичною для інтеграції систем різних постачальників і для забезпечення коректної взаємодії в рамках міжнародних проектів.

Галузеві та міждисциплінарні абревіатури

1. Енергетика та Smart Grid

SCADA – Supervisory Control and Data Acquisition – система диспетчерського керування та збору даних у промислових і енергетичних КФС.

DER – Distributed Energy Resources – розподілені енергетичні ресурси (сонячні панелі, вітрові турбіни, акумуляторні станції).

DSM – Demand Side Management – управління споживчим навантаженням для оптимізації роботи мережі.

EMS – Energy Management System – система управління енергоспоживанням та генерацією.

AMI – Advanced Metering Infrastructure – інфраструктура інтелектуального обліку енергоресурсів.

PMU – Phasor Measurement Unit – фазометричний вимірювальний пристрій для синхронного моніторингу мереж.

2. Транспорт і логістика

ITS – Intelligent Transport System – інтелектуальна транспортна система для управління потоками та безпекою.

V2X – Vehicle-to-Everything – комунікація транспортного засобу з інфраструктурою, іншими авто та мережами.

ADS – Automated Driving System – автоматизована система керування транспортним засобом.

TMS – Traffic Management System – система керування дорожнім рухом.

FMS – Fleet Management System – система управління автопарком.

AVL – Automatic Vehicle Location – автоматизоване відстеження місцезнаходження транспортних засобів.

3. Медицина та біоінформатика

EHR – Electronic Health Record – електронна медична картка.

CDSS – Clinical Decision Support System – система підтримки клінічних рішень.

PACS – Picture Archiving and Communication System – система архівування та передачі медичних зображень.

MIoT – Medical Internet of Things – медичний Інтернет речей.

HL7 – Health Level Seven International – стандарт обміну медичною інформацією.

FHIR – Fast Healthcare Interoperability Resources – стандарт інтероперабельності для обміну медичними даними.

Сучасні кіберфізичні системи, енергетичні мережі, транспортні комплекси та медичні інформаційні інфраструктури функціонують у багатокомпонентному середовищі, де застосовуються різноманітні технологічні стандарти та протоколи. Для забезпечення єдиного розуміння термінів і спрощення міждисциплінарної комунікації необхідно використовувати уніфіковані абревіатури з чітким розшифруванням та контекстом застосування.

Такі позначення дають змогу швидко орієнтуватися у спеціалізованих документах, технічних описах і наукових публікаціях, особливо коли мова йде про складні інтегровані системи. У галузях енергетики, транспорту та медицини абревіатури виступають своєрідним «словником» для розробників, інженерів і дослідників. Вони дають змогу однозначно ідентифікувати технології, методи та рішення, що впроваджуються у рамках цифровізації та автоматизації процесів. Знання та правильне тлумачення цих скорочень є критично важливими при проектуванні, налаштуванні та експлуатації складних систем.

Представлена табл. 1 містить добірку найпоширеніших абревіатур, що охоплюють ключові напрямки – від систем моніторингу та управління в енергетиці (SCADA, EMS, DER) до інтелектуальних транспортних технологій (ITS, V2X, ADS) та медичних інформаційних платформ (EHR, CDSS, PACS). Для кожної абревіатури подано розшифрування та сферу застосування, що уможливлює швидко встановити її призначення в конкретному технологічному контексті. Систематизація термінів у такому форматі є корисною не лише для навчальних та дослідницьких цілей, але й для стандартизації у проєктах, де взаємодіють фахівці з різних предметних галузей. Це зменшує кількість непорозумінь, прискорює обмін інформацією та підвищує ефективність міжгалузевих рішень.

Таким чином, табл. 1 виконує функцію базового орієнтира у складній термінології сучасних технологій, сприяючи формуванню єдиного інформаційного простору для проєктування, впровадження та супроводу кіберфізичних та інформаційних систем.

Таблиця 1

Базові абревіатури та їх розшифрування

Абревіатура	Розшифрування	Контекст застосування
SCADA	Supervisory Control and Data Acquisition	Промислові та енергетичні системи моніторингу та управління
DER	Distributed Energy Resources	Інтеграція відновлюваних джерел енергії в мережу
DSM	Demand Side Management	Балансування попиту і пропозиції в енергосистемах
EMS	Energy Management System	Управління енергетичними потоками
AMI	Advanced Metering Infrastructure	Збір і передача даних інтелектуальних лічильників
PMU	Phasor Measurement Unit	Синхронні вимірювання у мережах Smart Grid
ITS	Intelligent Transport System	Інтелектуальне керування дорожнім рухом
V2X	Vehicle-to-Everything	Комунікація транспортних засобів з інфраструктурою
ADS	Automated Driving System	Автоматизоване керування транспортом
TMS	Traffic Management System	Централізоване управління транспортними потоками
FMS	Fleet Management System	Моніторинг і планування роботи автопарку
AVL	Automatic Vehicle Location	GPS-моніторинг транспорту
EHR	Electronic Health Record	Електронна картка пацієнта
CDSS	Clinical Decision Support System	Інтелектуальна підтримка прийняття клінічних рішень
PACS	Picture Archiving and Communication System	Архівація та передача медичних зображень
MIoT	Medical Internet of Things	IoT-пристрої в медицині
HL7	Health Level Seven International	Стандарти обміну медичними даними
FHIR	Fast Healthcare Interoperability Resources	Інтероперабельність медичних систем

Табл. 1 складається з трьох стовпців: абревіатури, їх розшифрування та контекст застосування. У першому стовпці подано скорочення, що є стандартними у професійному середовищі відповідних галузей. Другий стовпець подає повну форму кожного скорочення англійською мовою, що відповідає міжнародним стандартам термінології. Третій стовпець уточнює, в якій сфері чи для яких завдань використовується відповідна технологія або система. Енергетичний блок охоплює абревіатури, пов'язані з автоматизацією керування енергетичними мережами та обладнанням: SCADA – системи диспетчерського контролю та збору даних, DER – розподілені енергетичні ресурси, DSM – управління попитом, EMS – системи енергоменеджменту, AMI – інфраструктура інтелектуальних лічильників, PMU – пристрої фазорних вимірювань. Ці скорочення широко застосовуються в проєктах Smart Grid та відновлюваної енергетики. Транспортний блок представлено абревіатурами ITS, V2X, ADS, TMS, FMS та AVL, які описують інтелектуальні транспортні системи, технології взаємодії транспортних засобів з інфраструктурою, автоматизоване керування, централізоване управління трафіком, моніторинг автопарку та GPS-відстеження. Ці технології спрямовані на підвищення безпеки дорожнього руху, оптимізацію логістики та розвиток автономного транспорту. Медичний блок містить абревіатури EHR, CDSS, PACS, MIoT, HL7 та FHIR, що охоплюють електронні медичні записи, системи підтримки клінічних рішень, технології архівації та передачі медичних зображень, застосування IoT у медицині та стандарти обміну медичними даними. Їх впровадження сприяє цифровій трансформації охорони здоров'я, підвищує якість діагностики та забезпечує інтероперабельність між медичними інформаційними системами.

Структура таблиці дозволяє швидко знайти потрібний термін та отримати стислу, але вичерпну інформацію щодо його призначення, що робить її ефективним інструментом для підготовки технічної документації, навчальних матеріалів та дослідницьких звітів.

Новітні галузеві абревіатури

SG-XAI – Smart Grid Explainable AI – пояснюваний ШІ для прогнозування і оптимізації в енергомережах.

V2X-SAFE – V2X Secure Adaptive Framework for Edge – безпечна адаптивна платформа V2X для периферійної обробки даних.

H-IDS – Healthcare Intrusion Detection System – система виявлення вторгнень у медичних IoT-пристроях.

BMS-ML – Battle Management System with Machine Learning – система управління бойовими діями з ML-модулями прогнозування.

Міжнародні та національні стандарти
формують нормативну основу для проєктування, експлуатації та оцінювання кіберфізичних систем (КФС) і нейромережевих технологій у сфері управління ризиками. Абревіатури у цій категорії є офіційно закріпленими скороченнями, що використовуються у стандартах, протоколах і регламентних документах для забезпечення однозначності тлумачень та інтероперабельності рішень. Вони охоплюють як глобальні організації зі стандартизації, так і регіональні чи національні інституції, відповідальні за розробку та впровадження нормативних вимог.

Таблиця 2

Абревіатура	**Розшифрування**	**Контекст застосування**
ISO	International Organization for Standardization	Міжнародна організація зі стандартизації
IEC	International Electrotechnical Commission	Міжнародна електротехнічна комісія
IEEE	Institute of Electrical and Electronics Engineers	Інститут інженерів з електротехніки та електроніки
NIST	National Institute of Standards and Technology	Національний інститут стандартів і технологій США
ENISA	European Union Agency for Cybersecurity	Агентство Європейського Союзу з кібербезпеки
DSTU	Derzhavni Standarty Ukrainy	Державні стандарти України

Систематизація таких абревіатур має практичне значення для створення єдиної терміносистеми, що дозволяє узгоджувати технічну документацію, мінімізувати ризики непорозумінь під час міжнародних проєктів та полегшувати сертифікацію продуктів і процесів. Нижче наведена таблиця містить ключові скорочення, які найчастіше зустрічаються у контексті КФС, інформаційної безпеки та штучного інтелекту. Для кожного скорочення наведено його офіційне розшифрування та контекст застосування у практиці управління ризиками.

Технологічні концепції та архітектури штучних нейронних мереж (ШНМ) визначають фундаментальні підходи до побудови, навчання та впровадження інтелектуальних систем у кіберфізичному середовищі.

Таблиця 3

Абревіатура	Розшифрування	Контекст застосування
CPS	Cyber-Physical System	Кіберфізична система
IoT	Internet of Things	Інтернет речей
IIoT	Industrial Internet of Things	Промисловий Інтернет речей
AI	Artificial Intelligence	Штучний інтелект
ML	Machine Learning	Машинне навчання
DL	Deep Learning	Глибинне навчання
XAI	Explainable Artificial Intelligence	Пояснюваний штучний інтелект
ZTA	Zero Trust Architecture	Архітектура нульової довіри
DT	Digital Twin	Цифровий двійник
MLP	Multilayer Perceptron	Багатошаровий перцептрон
CNN	Convolutional Neural Network	Згорткова нейронна мережа
RNN	Recurrent Neural Network	Рекурентна нейронна мережа
LSTM	Long Short-Term Memory	Нейромережа з довгою короткочасною пам’яттю
GRU	Gated Recurrent Unit	Рекурентний блок із керованими вентилями
AE	Autoencoder	Автокодер
VAE	Variational Autoencoder	Варіаційний автокодер
GAN	Generative Adversarial Network	Генеративно-змагальна мережа
DBN	Deep Belief Network	Глибока мережа довіри

Абревіатури цієї категорії охоплюють як загальні парадигми, наприклад штучний інтелект (AI) та глибинне навчання (DL), так і конкретні архітектурні рішення, такі як багатошаровий перцептрон (MLP), згорткові нейронні мережі (CNN) чи рекурентні структури (RNN, LSTM, GRU). Використання уніфікованих скорочень для технологічних понять та архітектур дозволяє забезпечити єдність термінології в дослідженнях, технічній документації, стандартах та проєктних матеріалах. Це особливо важливо для міждисциплінарних команд, де розробники, інженери, фахівці з безпеки та менеджери повинні однаково трактувати технічні терміни. Наведена нижче таблиця містить перелік ключових скорочень, що

описують як загальні технологічні концепти, так і конкретні архітектурні моделі нейронних мереж. Для кожної абревіатури наведено повне розшифрування та контекст застосування у сфері управління ризиками та розробки інтелектуальних компонентів кіберфізичних систем.

Сфера інформаційної безпеки та ризик-менеджменту у кіберфізичних системах (КФС)

оперує значною кількістю абревіатур, що позначають ключові процеси, інструменти та організаційні структури, відповідальні за захист даних і безперервність функціонування критичних компонентів. Ці скорочення охоплюють як назви систем виявлення та запобігання загрозам (IDS/IPS, SIEM), так і спеціалізовані організаційні одиниці (SOC, CSIRT), а також категорії загроз (APT, MITM) та інфраструктурні рішення (PKI). Єдність у вживанні таких абревіатур має критичне значення для оперативної взаємодії між командами кіберзахисту, адміністраторами мереж та розробниками систем безпеки. Невірне або непослідовне використання цих скорочень може призвести до хибного трактування завдань, затримок у реагуванні та зниження загального рівня кіберстійкості системи.

Таблиця 4

Абревіатура	**Розшифрування**	**Контекст застосування**
ISMS	Information Security Management System	Система управління інформаційною безпекою
CSIRT	Computer Security Incident Response Team	Команда реагування на комп’ютерні інциденти
SOC	Security Operations Center	Центр операційної безпеки
IDS/IPS	Intrusion Detection/Prevention System	Системи виявлення/запобігання вторгненням
SIEM	Security Information and Event Management	Управління інформацією та подіями безпеки
MITM	Man-in-the-Middle	Атака «людина посередині»
APT	Advanced Persistent Threat	Розвинена постійна загроза
PKI	Public Key Infrastructure	Інфраструктура відкритих ключів

Наведена таблиця містить найпоширеніші скорочення, що застосовуються в контексті кіберзахисту КФС та організації процесів управління ризиками. Для кожної абревіатури подано офіційне розшифрування та описано контекст її використання у рамках систем моніторингу, реагування та запобігання інцидентам безпеки.

Промислові та мережеві технології
є основою функціонування кіберфізичних систем (КФС), забезпечуючи обмін даними, інтеграцію компонентів і виконання виробничих процесів у реальному часі. Абревіатури цієї категорії охоплюють як стандартизовані промислові протоколи взаємодії (OPC UA, MQTT, REST), так і інтерфейси, контролери та засоби синхронізації часу (HMI, PLC, RTU, NTP). Вони також включають інструменти для інтеграції програмних компонентів (API) та оцінювання ефективності роботи систем (KPI). Використання уніфікованих скорочень у цій сфері є критично важливим для інтероперабельності обладнання різних виробників, спрощення технічної документації та прискорення процесів налаштування й обслуговування систем. Оскільки КФС функціонують у гетерогенних середовищах, чітке визначення значень і контексту застосування цих абревіатур допомагає уникнути помилок у конфігурації та експлуатації.

Таблиця 5

Ключові абревіатури та їх застосування в кіберфізичних системах

Абревіатура	Розшифрування	Контекст застосування
OPC UA	Open Platform Communications Unified Architecture	Промисловий стандарт взаємодії між КФС і системами автоматизації
MQTT	Message Queuing Telemetry Transport	Легковаговий протокол обміну повідомленнями у IIoT
REST	Representational State Transfer	Архітектурний стиль побудови веб‑сервісів у КФС
TLS	Transport Layer Security	Захист передавання даних між компонентами КФС
HMI	Human-Machine Interface	Інтерфейс взаємодії оператора з КФС
PLC	Programmable Logic Controller	Основний елемент управління технологічними процесами
RTU	Remote Terminal Unit	Віддалений термінальний пристрій збору даних
NTP	Network Time Protocol	Синхронізація часу у розподілених КФС
API	Application Programming Interface	Інтеграція програмних компонентів КФС
KPI	Key Performance Indicator	Ключові показники ефективності системи чи процесу

Подана таблиця містить основні скорочення, що позначають ключові промислові й мережеві технології, з повними розшифруваннями та зазначенням сфер їх застосування в управлінні ризиками та забезпеченні безперервності роботи КФС.

Галузеві абревіатури
відображають специфіку застосування кіберфізичних систем (КФС) та нейромережевих технологій у різних секторах економіки та безпеки – енергетиці, транспорті, медицині, обороні. Вони охоплюють як назви комплексних технологічних рішень (SCADA, ITS, C4ISR), так і спеціалізованих підсистем, пристроїв та стандартів обміну даними (DER, AMI, HL7, FHIR). Ці скорочення формуються під впливом галузевих нормативів та міжнародних стандартів, що регламентують вимоги до безпеки, інтероперабельності та ефективності роботи. Для мультидисциплінарних проєктів, де інтегруються енергетичні, транспортні, медичні та оборонні сегменти, уніфіковане використання таких абревіатур дозволяє уникнути розбіжностей у термінології, спростити комунікацію та пришвидшити погодження технічних рішень.

Таблиця 6

Ключові абревіатури та їх застосування у галузях енергетики, транспорту та медицини

Абревіатура	Розшифрування	Контекст застосування
SCADA	Supervisory Control and Data Acquisition	Промислові та енергетичні системи моніторингу та управління
DER	Distributed Energy Resources	Інтеграція відновлюваних джерел енергії в мережу
DSM	Demand Side Management	Балансування попиту і пропозиції в енергосистемах
EMS	Energy Management System	Управління енергетичними потоками
AMI	Advanced Metering Infrastructure	Збір і передача даних інтелектуальних лічильників
PMU	Phasor Measurement Unit	Синхронні вимірювання у мережах Smart Grid
ITS	Intelligent Transport System	Інтелектуальне керування дорожнім рухом

Закінчення табл. 6

V2X	Vehicle-to-Everything	Комунікація транспортних засобів з інфраструктурою
ADS	Automated Driving System	Автоматизоване керування транспортом
TMS	Traffic Management System	Централізоване управління транспортними потоками
FMS	Fleet Management System	Моніторинг і планування роботи автопарку
AVL	Automatic Vehicle Location	GPS-моніторинг транспорту
EHR	Electronic Health Record	Електронна картка пацієнта
CDSS	Clinical Decision Support System	Інтелектуальна підтримка прийняття клінічних рішень
PACS	Picture Archiving and Communication System	Архівація та передача медичних зображень
MIoT	Medical Internet of Things	IoT-пристрої в медицині
HL7	Health Level Seven International	Стандарти обміну медичними даними
FHIR	Fast Healthcare Interoperability Resources	Інтероперабельність медичних систем

Наведена таблиця містить добірку найуживаніших скорочень, що застосовуються у профільних галузях. Для кожної абревіатури наведено офіційне розшифрування та зазначено її роль у забезпеченні функціональності, безпеки та управління ризиками в межах конкретного сектора.

Приклади авторських абревіатур:

формуються для позначення перспективних концепцій, архітектур і технологічних рішень, які ще не набули широкого поширення у стандартах або галузевих глосаріях, але вже мають значний потенціал для впровадження у кіберфізичних системах (КФС) та нейромережевих підходах до управління ризиками. Вони створюються з урахуванням загальноприйнятих правил побудови скорочень, уникають конфліктів з існуючими позначеннями та орієнтовані на зрозуміле трактування в міжнародному контексті. Такі скорочення виконують одразу кілька функцій: спрощують комунікацію між учасниками проєкту, знижують обсяг технічної документації та слугують основою для подальшої стандартизації нових рішень. Крім того, їх систематизація на ранніх етапах дозволяє уніфікувати підхід до опису інноваційних технологій, що підвищує їх впізнаваність і сприяє ефективному обміну знаннями.

Таблиця 7

Ключові абревіатури та їх застосування у сучасних кіберфізичних системах

Абревіатура	Розшифрування	Контекст застосування
RMLR	Risk-Managed Learning Rate	Динамічне регулювання швидкості навчання моделі з урахуванням ризик-профілю
E-DT	Edge Digital Twin	Цифровий двійник на периферійному вузлі для зменшення затримок
F-GAN	Federated GAN	Генеративно-змагальна мережа з федеративним навчанням для синтезу даних без централізації
CPS-XAI	CPS Explainable AI	Пояснюваний ШІ для кіберфізичних систем
AD-SOC	Adaptive SOC	Центр операційної безпеки з адаптивними моделями аналізу телеметрії
SG-XAI	Smart Grid Explainable AI	Пояснюваний ШІ для енергомереж
V2X-SAFE	V2X Secure Adaptive Framework for Edge	Безпечна адаптивна платформа V2X для периферійної обробки даних
H-IDS	Healthcare Intrusion Detection System	Система виявлення вторгнень у медичних IoT-пристроях

Наведена таблиця містить перелік нових скорочень, розроблених у межах цієї роботи, із зазначенням повного розшифрування та контексту застосування. Вони охоплюють як розширення існуючих архітектур (наприклад, поєднання технологій Explainable AI з цифровими двійниками або енергомережами), так і нові організаційні та функціональні рішення в сфері безпеки та прогнозування загроз.

Висновки

Оновлені висновки за розділом 3 мають на меті підсумувати результати виконаного дослідження, продемонструвати їхній внесок у розвиток теоретичних і прикладних основ управління ризиками кіберфізичних систем та закласти підґрунтя для логічного переходу до наступного розділу. Протягом розділу було виконано систематичний аналіз і впорядкування термінологічної бази, що використовується у сфері нейромережевих методів управління ризиками, з урахуванням вимог міжнародних і національних стандартів, а також актуальних тенденцій у наукових дослідженнях та промисловій практиці. Встановлено, що відсутність уніфікованого термінологічного апарату є одним із ключових бар'єрів для ефективної інтеграції інтелектуальних технологій у КФС, оскільки призводить до неоднозначного трактування вимог, ускладнює координацію між учасниками проєктів і підвищує ймовірність помилок на етапах проєктування та експлуатації.

У межах підрозділу 3.1 було визначено та структуровано базові поняття, що становлять ядро терміносистеми КФС і ШНМ, розкрито їхні міждисциплінарні зв'язки та продемонстровано їхню еволюцію під впливом розвитку технологій, стандартизаційних процесів і змін у профілях загроз.

Підрозділ 3.2 присвячено розробці багатовимірної класифікації термінів, яка дає можливість упорядковувати їх за функціональною роллю, галузевою належністю, архітектурними особливостями та відповідністю стандартам. Такий підхід забезпечує гнучкість і масштабованість терміносистеми, що особливо важливо у швидкозмінному середовищі КФС.

У підрозділі 3.3 розглянуто принципи та механізми стандартизації термінології, проаналізовано практики гармонізації понять у межах ISO, IEC, IEEE, NIST, ENISA та інших впливових організацій. Визначено, що систематична інтеграція стандартів у локальну терміносистему не лише підвищує якість і однозначність технічної документації, але й полегшує міжнародну співпрацю, прискорює сертифікаційні процедури та підвищує рівень довіри до рішень у галузі.

У підрозділі 3.4 проведено глибокий аналіз абревіатур, які виступають компактними носіями складних технічних понять і широко використовуються в управлінні ризиками, проєктуванні та експлуатації КФС. Сформовано розширений каталог скорочень з поділом на тематичні групи: стандарти, технологічні концепції, архітектури ШНМ, інформаційна безпека, промислові й мережеві технології, галузеві специфічні терміни та нові авторські абревіатури. Для кожної позиції подано розшифрування і контекст застосування, що підвищує практичну цінність каталогу для розробників, інтеграторів, дослідників і фахівців з безпеки.

Отримані результати мають як наукове, так і прикладне значення. З наукової точки зору, розділ 3 формує методологічну базу для подальших

досліджень у сфері термінологічного аналізу та стандартизації, демонструє ефективність інтеграції міжнародних стандартів у локальні терміносистеми та пропонує гнучку класифікаційну модель, здатну адаптуватися до появи нових технологій і загроз. З прикладної точки зору, створені каталоги термінів і абревіатур можуть бути безпосередньо використані під час підготовки технічної, проєктної та навчальної документації, у процесах міжгалузевої координації, сертифікації рішень і навчанні персоналу.

Практичне значення монографії в контексті цього розділу полягає у створенні комплексного інструменту для уніфікації та впровадження єдиної терміносистеми у сфері управління ризиками КФС. Це забезпечує прозорість та ефективність комунікацій між учасниками проєктів, скорочує витрати на узгодження документації, зменшує кількість помилок через різночитання та прискорює процес упровадження інновацій. Крім того, запропонований підхід сприяє формуванню спільного понятійного простору між розробниками технологій, операторами інфраструктур, органами регуляторного контролю та академічною спільнотою, що є критично важливим для підвищення кіберстійкості та надійності КФС.

У підсумку, розділ 3 виконує роль методологічного та термінологічного фундаменту всієї монографії. Він не лише впорядковує і стандартизує ключові поняття, а також створює практичні інструменти для їх застосування у реальних умовах експлуатації кіберфізичних систем. Логічним продовженням виконаного дослідження є розділ 4, у якому розпочинається поглиблений аналіз та інтерпретація абревіатур, розробка рекомендацій щодо їх уніфікованого використання та створення галузевого каталогу скорочень, що стане важливим кроком у реалізації стратегічної мети, а саме: побудови несуперечливої, актуальної та міжнародно сумісної терміносистеми для нейромережевих методів управління ризиками КФС.

Джерела:

122. Корченко А. Г., Архипов А. Є., Казмирчук С. В. Аналіз і оцінювання ризиків інформаційної безпеки: монографія. Київ : ТОВ «Логос Україна», 2013. 275 с.

123. González-Bernaldo de Quirós F., Dawidowski A., Figar S., Otero C., Luna D. Terminology services: standard terminologies to control health vocabulary. Applied Clinical Informatics. 2018. Vol. 9, Iss. 1. P. 75–90. DOI: 10.1055/s-0038-1641200.

124. Vuokko R., Vakkuri A., Palojoki S. Systematized Nomenclature of Medicine–Clinical Terminology (SNOMED CT) clinical use cases in the context of electronic health record systems: Systematic literature review. JMIR Medical Informatics. 2023. Vol. 11. Art. e43750. DOI: 10.2196/43750.

125. (Вичерпне використання SNOMED CT для обробки вільного тексту у клінічній інформатиці). Journal of Medical Internet Research. 2021. DOI: 10.2196/24594.

126. (Огляд стандартної термінології в реабілітації). Disability and Rehabilitation. 2022. DOI: 10.1080/17483107.2022.2112985.

127. (Структурована медична термінологія та використання структурованої звітності). Journal of Veterinary Diagnostic Investigation. 2018. Vol. 30, Iss. 1. P. 17–25. DOI: 10.1177/1040638717738276.

128. Izura C., Playfoot D. A normative study of acronyms and acronym naming. Behavior Research Methods. 2012. Vol. 44, Iss. 3. P. 862–889. DOI: 10.3758/s13428-011-0175-8.

129. Mowery D. L., South B. R., Christensen L., et al. Normalizing acronyms and abbreviations … Journal of Biomedical Semantics. 2016. Vol. 7. Art. 43. DOI: 10.1186/s13326-016-0084-y.

130. Taylor C., Boland L., Shaw S. C., et al. Are We Speaking the Same Language? Terminology Consistency in Patient Decision Aids: An International Survey. Health Communication. 2023. DOI: 10.1177/19375867231225395.

131. Barnett A., Doubleday Z. A. The growth of acronyms in the scientific literature. eLife. 2020. Vol. 9. e60080. DOI: 10.7554/eLife.60080.

132. Cañero-Chiriboga D., et al. Deciphering clinical abbreviations with a privacy-protecting machine translation model. Nature Communications. 2022. Vol. 13. Art. 7350. DOI: 10.1038/s41467-022-35007-9.

133. McCrobie D. An analysis of acronyms in written text. Proceedings of the Human Factors Society Annual Meeting. 1986. Vol. 30, Iss. 9. P. 910–914. DOI: 10.1177/154193128603000920.

134. Lyons-Weiler J., Xu J., Hupcey J. Effect of expansion of abbreviations and acronyms on patient comprehension. JAMA Network Open. 2022. Vol. 5, Iss. 4. e2212320. DOI: 10.1001/jamanetworkopen.2022.12320.

135. Moon S., Pakhomov S., Liu N., Ryan J. O., Melton G. B. A sense inventory for clinical abbreviations and acronyms created using clinical notes and medical dictionary resources. Journal of the American Medical Informatics Association. 2014. Vol. 21, Iss. 2. P. 299–307. DOI: 10.1136/amiajnl-2012-001506.

136. Radiology. Clarification of medical abbreviations, initialisms, and acronyms. 2013. Vol. 267, Iss. 1. P. 10–16. DOI: 10.1148/radiol.13131828.

137. Internal Medicine Journal. Overview of shorthand medical glossary (OMG) study. 2015. Vol. 45, Iss. 7. P. 744–748. DOI: 10.1111/imj.12668.

138. QJM: An International Journal of Medicine. Analysis of abbreviations used by residents in admission notes and their understanding by nurses. 2016. Vol. 109, Iss. 9. P. 603–609. DOI: 10.1093/qjmed/hcx241.

139. Pouran Ben Veyseh A., Dernoncourt F., Tran Q. H., Nguyen T. H. What Does This Acronym Mean? Introducing a New Dataset for Acronym Identification and Disambiguation. Proceedings of the 28th International

Conference on Computational Linguistics (COLING). 2020. P. 3285–3301. DOI: 10.18653/v1/2020.coling-main.292.

140. AcX: system, techniques, and experiments for acronym expansion. Studies in Health Technology and Informatics. 2001. P. 371–375. DOI: 10.14778/3551793.3551812.

141. Jacobs K., Itai A., Wintner S. Acronyms: identification, expansion and disambiguation. Annals of Mathematics and Artificial Intelligence. 2018. Vol. 88. P. 517–532. DOI: 10.1007/s10472-018-9608-8.

142. Ali I., Haileyesus M., Hnatyshyn S., Ott J.-L., Hnatyshin V. Automated Extraction of Acronym-Expansion Pairs from Scientific Papers. arXiv Preprint. 2024. DOI: 10.48550/arXiv.2412.01093.

143. Justeson J. S., Katz S. M. A simple algorithm for identifying abbreviation definitions in biomedical text. In: Bio-computing 2003. 2002. P. 451–462. DOI: 10.1142/9789812776303_0042.

144. Pustejovsky J., Castano J., Cochran B., Kotecki M., Morrell M. Automatic extraction of acronym-meaning pairs from MEDLINE databases. In: MEDINFO 2001. 2001. P. 371–375. DOI: 10.3233/978-1-60750-928-8-371.

145. Sohn S., Comeau D. C., Kim W., Wilbur W. J. Abbreviation definition identification based on automatic precision estimates. BMC Bioinformatics. 2008. Vol. 9. Art. 402. DOI: 10.1186/1471-2105-9-402.

146. Yeganova L., Comeau D. C., Wilbur W. J. Identifying abbreviation definitions: machine learning with naturally labeled data. In: International Conference on Machine Learning and Applications (ICMLA). 2010. P. 499–505. DOI: 10.1109/ICMLA.2010.166.

147. Rajkomar A., et al. Deciphering clinical abbreviations with a privacy-protecting machine learning system. Nature Communications. 2022. Vol. 13. Art. 7456. DOI: 10.1038/s41467-022-35007-9.

148. Schwartz A. S., Hearst M. A. A simple algorithm for identifying abbreviation definitions in biomedical text. Pacific Symposium on Biocomputing. 2002. P. 451–462. DOI: 10.1142/9789812776303_0042.

149. Zhou W., Torvik V. I., Smalheiser N. R. ADAM: another database of abbreviations in MEDLINE. Bioinformatics. 2006. Vol. 22(22). P. 2813–2818. DOI: 10.1093/bioinformatics/btl480.

150. Larkey L. S., Ogilvie P., Price M. A., Tamilio B. Acrophile: an automated acronym extractor and server. In: Proceedings of the 5th ACM Conference on Digital Libraries. 2000. P. 205–214. DOI: 10.1145/336597.336664.

151. ISO/IEC 27005:2022. Information security, cybersecurity and privacy protection — Information security risk management. Geneva : International Organization for Standardization, 2022. 64 p.

152. Bizer C., Heath T., Berners-Lee T. Linked Data — The Story So Far. International Journal on Semantic Web and Information Systems. 2009. Vol. 5, Iss. 3. P. 1–22. DOI: 10.4018/jswis.2009081901.

РОЗДІЛ 4
АБРЕВІАТУРИ, ЇХ ТАКСОНОМІЯ, КЛАСИФІКАЦІЯ ТА ІНТЕРПРЕТАЦІЯ

Абревіатури є ключовим інструментом сучасної науково-технічної комунікації, особливо у сферах, де обсяг спеціалізованої термінології зростає надзвичайно швидкими темпами та має міждисциплінарний характер [153], [154]. У галузях кіберфізичних систем (КФС), штучних нейронних мереж (ШНМ) та кібербезпеки вони виконують не лише роль лінгвістичних скорочень, але й функцію стандартизованих ідентифікаторів понять, методів, технологій і нормативних документів [155]. Завдяки використанню абревіатур скорочується обсяг технічних описів, підвищується зручність сприйняття інформації у взаємодії між фахівцями та прискорюється обмін даними у письмовій та усній комунікації [156]. У глобальному контексті, де наукові та промислові проєкти реалізуються міжнародними командами, абревіатури перетворюються на універсальний комунікаційний код, що долає мовні бар'єри та забезпечує однозначне розуміння ключових термінів [157], [158].

Ефективність такого підходу можлива лише за умови їх чіткої стандартизації та закріплення у відповідних глосаріях, стандартах або технічних регламентах [159]. Тут важлива роль належить таксономії абревіатур – системному впорядкуванню скорочень за визначеними ознаками, що дозволяє встановлювати зв'язки між ними, уникати дублювання та підтримувати цілісність терміносистеми [160], [161]. Таксономія – це методологічний інструмент для формування логічної структури абревіатур, що забезпечує їх класифікацію за сферою застосування (промислові стандарти, протоколи зв'язку, алгоритми ШНМ), типом утворення (акроніми, ініціалізми, комбіновані форми) чи рівнем упровадження (міжнародні, національні, корпоративні) [162], [163]. Системний підхід до таксономії дає змогу вирішити низку практичних завдань.

Уніфікація позначень передбачає закріплення кожного скорочення за одним поняттям у межах терміносистеми, що мінімізує ризик термінологічних конфліктів [164]. Оптимізація пошуку та використання полягає в тому, що структуровані каталоги полегшують навігацію в базах знань, стандартах і технічній документації [165]. Міжгалузева сумісність досягається завдяки ідентифікації скорочень, які в різних галузях можуть мати різне значення, що зменшує ризик помилкової інтерпретації [166], [167].

У контексті КФС та ШНМ, де активно впроваджуються концепції Індустрії 4.0, цифрових двійників, IoT/IIoT, Zero Trust Architecture та Explainable AI, кількість абревіатур зростає експоненційно [168], [169]. Це створює потребу щодо впровадження постійно оновлюваних таксономічних моделей, які підтримують актуальність і точність терміносистеми [170].

Отже, абревіатури в зазначених сферах – це не лише мовні інструменти, а також і стратегічниі елементи інформаційної інфраструктури, їхнє таксономічне впорядкування є ключовою умовою для забезпечення точності, узгодженості та ефективності комунікацій [171], [172]. Однією з найпоширеніших проблем, пов'язаних із використанням абревіатур у кіберфізичних системах, штучних нейронних мережах та сфері кібербезпеки, є їх неоднозначність [173]. Вона виникає тоді, коли одне й те саме скорочення використовується для позначення різних понять залежно від контексту. Така ситуація часто трапляється в міжгалузевих проєктах, де фахівці з різних технічних напрямів можуть вкладати у скорочення відмінні значення. Відсутність чітких правил або посилань на офіційні стандарти призводить до того, що навіть досвідчені спеціалісти довгий час витрачають на з'ясування точного змісту терміна [174].

Багатозначність абревіатур має не лише лінгвістичний, але й технічний вимір. У сфері управління ризиками та кіберзахисту некоректне тлумачення скорочення може призвести до вибору неправильних параметрів налаштування обладнання, застосування не тих алгоритмів або навіть до критичних збоїв у роботі системи [175]. Такі помилки особливо небезпечні в середовищах із високими вимогами до безперервності функціонування, скажімо, в енергетиці, транспорті чи охороні здоров'я, де наслідки некоректної інтерпретації можуть мати масштабний вплив [176]. Додаткову складність створюють випадки, коли абревіатури перекладаються кількома мовами. Навіть у межах офіційних перекладів міжнародних стандартів можуть існувати відмінності у виборі слів для розшифрування того самого скорочення [177]. Це ускладнює міждержавну координацію та створює ризики непорозумінь у міжнародних проєктах, де рішення повинні бути технічно сумісними й відповідати єдиним вимогам.

Ще однією причиною виникнення неоднозначностей є швидкий розвиток технологій. Нові абревіатури з'являються швидше, ніж відбувається їхня формалізація у стандартах чи глосаріях [178]. У результаті в наукових публікаціях, технічній документації та програмних інтерфейсах починають використовуватися неузгоджені скорочення, що підвищує навантаження на фахівців щодо перевірки джерел та верифікації значень.

Важливим наслідком багатозначності та неоднозначності є зростання вимог до процесів перевірки і валідації термінів. Для зменшення ризиків необхідно не лише фіксувати офіційні тлумачення абревіатур, але й вказувати контексти їх використання, а також надавати приклади застосування в різних сферах [179]. Такий підхід дає змогу зменшити кількість непорозумінь і забезпечити стабільність інтерпретації скорочень у довгостроковій перспективі. Обґрунтування потреби уніфікованої моделі таксономії абревіатур у сфері кіберфізичних систем, штучних нейронних мереж і кібербезпеки базується на необхідності забезпечення стабільності та однозначності термінологічної бази [153]. В умовах зростаючої складності технологічних рішень, багаторівневої інтеграції компонентів і участі

міжнародних команд відсутність єдиних правил побудови та інтерпретації скорочень призводить до затримок у розробці, збільшення витрат на узгодження документації та потенційних технічних конфліктів [154].

Уніфікована модель таксономії повинна охоплювати не лише створення та верифікацію нових абревіатур, але й підтримку їхньої актуальності відповідно до оновлень стандартів і галузевих регламентів [155]. Така модель має передбачати обов'язкову прив'язку кожного скорочення до конкретного контексту використання, галузевої сфери та офіційного джерела визначення [156]. Критерії добору одиниць для включення в каталог абревіатур можуть бути сформульовані з урахуванням міжнародної практики та рекомендацій провідних організацій зі стандартизації [157]. До таких критеріїв належать:

– офіційне закріплення скорочення в міжнародних або національних стандартах;

– наявність широкого використання у професійній спільноті;

– однозначність тлумачення у межах заданої терміносистеми;

– відсутність конфлікту з іншими скороченнями в межах суміжних галузей;

– відповідність принципам побудови акронімів та ініціалізмів, прийнятих у технічній комунікації [158].

Практика свідчить, що дотримання цих критеріїв дає змогу створювати каталоги абревіатур, які зручні у використанні, легко інтегруються в автоматизовані системи управління документацією та можуть застосовуватись у навчальних, дослідницьких і промислових цілях [159]. Це особливо важливо в контексті швидких змін у технологічному середовищі, коли своєчасне оновлення термінології є критичним для підтримання інформаційної сумісності [160]. Уніфікована модель таксономії абревіатур також сприяє покращанню якості аналітичних процесів, оскільки забезпечує можливість автоматизованої ідентифікації та класифікації скорочень у великих масивах даних [161]. Використання стандартизованих підходів до таксономії дозволяє інтегрувати термінологічні бази між різними організаціями та системами, що підвищує рівень координації в міжнародних проєктах [162], [163].

Таким чином, формування уніфікованої моделі таксономії абревіатур є стратегічним завданням, яке забезпечує підвищення ефективності комунікацій, зниження ризику технічних помилок і створює основу для подальшої стандартизації галузевої термінології [164], [165], [166]. Реалізація цієї моделі потребує координації зусиль між розробниками стандартів, науковою спільнотою, промисловими підприємствами та органами державного регулювання [167], [168], [169]. Перспективним напрямом розвитку є впровадження автоматизованих термінологічних сервісів, здатних у режимі реального часу оновлювати та перевіряти коректність абревіатур у технічній документації та інформаційних системах [170], [171], [172].

4.1. Методологія збору, систематизації та верифікації абревіатур

Ефективне управління абревіатурами в галузях кіберфізичних систем, штучних нейронних мереж та кібербезпеки потребує чітко визначеної методології збору, систематизації та верифікації термінів [153]. Стрімкий розвиток технологій, поява нових стандартів і зростання міждисциплінарних зв'язків призводять до безперервного збільшення кількості скорочень, що вживаються у наукових працях, технічній документації та нормативно-правових актах. Без структурованого підходу виникає ризик дублювання значень, використання застарілих або некоректних формулювань, а також неоднозначного тлумачення абревіатур у різних галузях [154].

Першим етапом методології є визначення джерел збору. Основними джерелами виступають міжнародні стандарти, розроблені організаціями ISO, IEC, IEEE, NIST, що містять офіційно затверджені скорочення та їхні визначення [155]. Окрім цього, вагоме значення мають технічні звіти міжнародних консорціумів і галузевих асоціацій, які регулярно оновлюють термінологію з урахуванням нових технологічних трендів [156]. Додатковим джерелом є наукові бази даних, такі як Scopus, IEEE Xplore та Web of Science, де можна виявити нові абревіатури на ранніх етапах їх використання [157]. Практичне значення має також аналіз технічних специфікацій виробників обладнання та програмного забезпечення, оскільки вони часто впроваджують власні скорочення, що згодом стають загальновживаними [158].

Другим етапом є класифікація зібраних абревіатур. Ієрархічний підхід передбачає впорядкування скорочень за рівнями, починаючи від загальних технологічних концепцій (наприклад, IoT, AI, ML) до спеціалізованих термінів конкретних протоколів або алгоритмів [159]. Фасетна класифікація дає можливість структурувати абревіатури за кількома незалежними ознаками, наприклад, за сферою застосування (енергетика, транспорт, медицина), типом технології (мережеві протоколи, алгоритми безпеки, архітектури ШНМ) та статусом стандартизації (міжнародний стандарт, національний стандарт, корпоративна специфікація) [160]. Контекстуальний підхід орієнтується на визначення значення скорочення залежно від його вживання в конкретному документі, проєкті чи галузі, що дозволяє уникнути плутанини у випадках багатозначності [161].

Наступним етапом є фільтрація зібраних абревіатур, мета якої – виключення застарілих, дубльованих або неофіційних позначень. Основними принципами фільтрації є релевантність, офіційність і поширеність [162]. Релевантність означає відповідність скорочення обраній галузі та його актуальність для сучасних технологічних процесів [163]. Офіційність передбачає наявність посилання на нормативний документ або публікацію, де наведено формальне визначення абревіатури [164]. Поширеність визначається частотою використання скорочення у міжнародних і національних джерелах, що дозволяє залишити в каталозі ті позначення, які мають реальне практичне

значення [165]. Верифікація абревіатур передбачає перевірку правильності їх написання, відповідності офіційним визначенням і узгодженості з іншими скороченнями у терміносистемі [166]. Для цього можуть застосовуватися як автоматизовані інструменти (наприклад, лінгвістичні аналізатори або спеціалізовані термінологічні бази), так і експертний перегляд, коли група фахівців проводить колективну оцінку правильності та доцільності вживання терміна [167]. Важливим завданням є ідентифікація потенційних конфліктів, коли одна й та сама абревіатура використовується для позначення різних понять у суміжних сферах, що може призвести до непорозумінь [168]. На окрему увагу заслуговує етап підтвердження правильності термінів. Він включає перевірку джерел походження скорочення, зіставлення його з аналогами в інших стандартах та оцінку його зрозумілості для цільової аудиторії [169]. Додатково враховується стабільність використання абревіатури – якщо вона активно застосовується упродовж тривалого часу, її доцільно закріпити у каталозі як офіційно рекомендовану [170].

Представлена схема (рис. 1) демонструє багаторівневу модель структуризації абревіатур, що використовується у сфері кіберфізичних систем (КФС) та інформаційної безпеки (ІБ). Модель побудована з урахуванням принципів ієрархічної та фасетної класифікації, описаних у попередньому підрозділі, та орієнтована на впорядкування скорочень за рівнями значущості й сферами застосування. Мета цієї візуалізації – створення зрозумілої, компактної та гнучкої структури, яка дозволяє швидко орієнтуватися у великому масиві скорочень, виявляти їх взаємозв’язки та зменшувати ризик неоднозначності під час використання в технічній документації, стандартах і проєктних матеріалах.

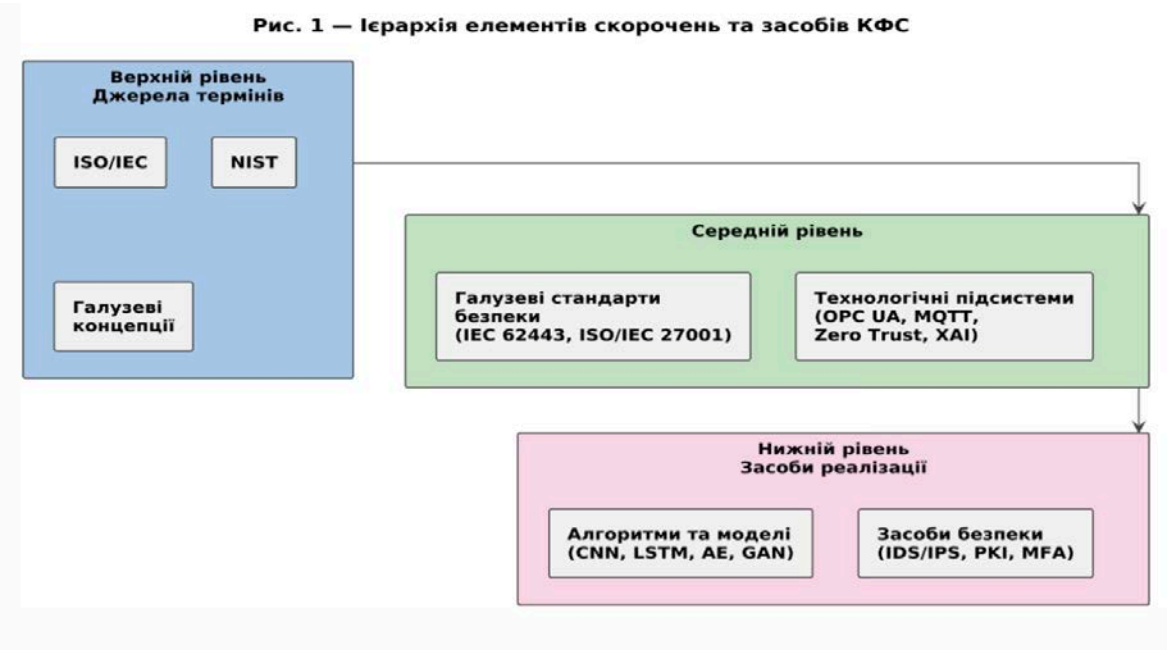


Рис. 1. Пояснення елементів схеми

Верхній рівень –
джерела формування офіційних скорочень, закріплених у міжнародних стандартах і концептуальних документах (ISO/IEC, NIST, загальні галузеві концепції). Він задає основну терміносистему, з якої формуються похідні категорії.

Середній рівень –

поділений на два підблоки:

Галузеві стандарти безпеки (наприклад, IEC 62443, ISO/IEC 27001), які регламентують вимоги до безпеки в межах КФС та ІБ.

Технологічні підсистеми (наприклад, OPC UA, MQTT, Zero Trust Architecture, Explainable AI), які описують конкретні архітектурні рішення, протоколи та концепції.

Нижній рівень –

містить конкретні інструменти, алгоритми та засоби реалізації:

Алгоритми та моделі (CNN, LSTM, Autoencoder, GAN), що використовуються для обробки, аналізу та прогнозування.

Засоби безпеки (IDS/IPS, PKI, MFA), які реалізують функції захисту, автентифікації та управління доступом.

Однією з найбільш поширених проблем, що виникають під час використання абревіатур у кіберфізичних системах, штучних нейронних мережах та інформаційній безпеці, є їх неоднозначність [173]. Це явище спостерігається тоді, коли одне й те саме скорочення використовується для позначення різних понять у залежності від контексту. У міжгалузевих проєктах подібна ситуація призводить до значних ускладнень, оскільки фахівці з різних технічних сфер можуть інтерпретувати абревіатуру по-різному [174]. Багатозначність абревіатур створює додаткові ризики у критично важливих системах. Неправильне тлумачення скорочення в документації чи програмному коді може призвести до використання некоректних налаштувань, активації невідповідних алгоритмів або порушення роботи захисних механізмів [175]. Такі помилки особливо небезпечні в енергетиці, транспорті, медицині та обороні, де некоректна інтерпретація може спричинити масштабні збої [176]. Ускладнює ситуацію і проблема перекладу абревіатур кількома мовами. Навіть в офіційних перекладах міжнародних стандартів можливі розбіжності у формулюваннях, що призводить до розмитості значень [177]. Це створює додаткові перешкоди для інтеграції рішень у міжнародних проєктах, де потрібна чітка технічна сумісність. Ще одним джерелом неоднозначностей є швидкий розвиток технологій. Нові абревіатури з’являються швидше, ніж встигають пройти формальне затвердження у стандартах чи глосаріях [178]. У результаті в публікаціях, технічній документації та інтерфейсах з’являються терміни, які не мають офіційного визначення, що ускладнює

їх правильне трактування. Для зменшення таких ризиків необхідно впроваджувати процеси, спрямовані на перевірку та уточнення значень абревіатур. Це включає не лише закріплення офіційних тлумачень, але й вказівку на конкретний контекст використання та надання прикладів застосування [179]. Подібний підхід допомагає забезпечити стабільність інтерпретацій і зменшує ймовірність помилок у розробці та експлуатації технологічних систем.

У процесі аналізу та впорядкування абревіатур у сфері кіберфізичних систем (КФС), штучних нейронних мереж (ШНМ) та інформаційної безпеки (ІБ) було виявлено низку ключових проблем, що суттєво впливають на точність комунікацій та якість технічної документації [153], [154]. Ці проблеми зумовлені як лінгвістичними факторами (неоднозначність, багатозначність), так і технологічними особливостями (швидкий розвиток галузі, відставання процесів стандартизації).

Таблиця 1

Основні проблеми використання абревіатур у КФС, ШНМ та ІБ

Проблема	Опис
Неоднозначність	Використання одного скорочення для позначення різних понять залежно від контексту, що створює труднощі у міжгалузевих проєктах [173], [174]
Багатозначність у критичних системах	Неправильне тлумачення абревіатур може призвести до використання некоректних налаштувань, запуску невідповідних алгоритмів або порушення роботи захисних механізмів [175], [176]
Проблеми перекладу	Розбіжності у перекладах міжнародних стандартів спричиняють розмитість значень і ускладнюють технічну сумісність рішень [177]
Відставання стандартизації	Поява нових абревіатур випереджає їх формальне закріплення у стандартах чи глосаріях, що збільшує ризик некоректного трактування [178]
Відсутність контексту використання	Брак офіційних тлумачень із прикладами призводить до нестабільності інтерпретацій і зростання кількості помилок [179]

Таблиця 1 подає систематизований перелік п’яти основних проблем, пов’язаних із використанням абревіатур у КФС, ШНМ та ІБ, разом із їх коротким описом і прикладами наслідків. До кожної проблеми додано

посилання на джерела [173–179], що підтверджують її актуальність і значущість. Така структуризація дозволяє швидко ідентифікувати типові ризики та враховувати їх під час розроблення терміносистеми, створення стандартів чи підготовки технічних матеріалів.

Потреба уніфікованої моделі таксономії абревіатур у сферах кіберфізичних систем, штучних нейронних мереж та інформаційної безпеки обумовлена необхідністю забезпечення точності, однозначності та стабільності терміносистеми [153]. У сучасних технологічних проєктах, де взаємодіють розробники, інтегратори, регулятори та кінцеві користувачі з різних країн, відсутність єдиних правил побудови і використання скорочень призводить до збільшення витрат на узгодження документації, затримок у впровадженні рішень та зростання ризику технічних помилок [154]. Уніфікована модель повинна охоплювати як процес створення та затвердження нових абревіатур, так і їх підтримку в актуальному стані відповідно до оновлень стандартів і галузевих регламентів [155]. Її побудова має спиратися на принципи системності, прозорості та міжнародної сумісності [156]. Одним із ключових аспектів такої моделі є обов’язкова прив’язка кожного скорочення до його офіційного джерела та конкретного контексту застосування [157]. Визначення критеріїв відбору це важливий етап формування каталогу абревіатур. До них належать офіційне закріплення у стандартах міжнародного чи національного рівня, значна поширеність у професійному середовищі, однозначність тлумачення в межах терміносистеми, відсутність конфлікту з іншими скороченнями та відповідність правилам побудови акронімів та ініціалізмів [158].

Дотримання цих критеріїв дозволяє створювати каталоги абревіатур, які є зрозумілими, компактними й легко інтегруються в автоматизовані системи управління документацією [159]. Такі каталоги зменшують ризик помилкових інтерпретацій і забезпечують ефективну взаємодію між різними учасниками проєктів, у тому числі міжнародних [160]. Уніфікована модель таксономії також має безпосередній вплив на ефективність аналітичних процесів. Вона дає можливість упроваджувати автоматизовані механізми ідентифікації та класифікації абревіатур у великих масивах даних, підвищуючи швидкість пошуку та якість аналізу [161]. Використання стандартизованих підходів сприяє інтеграції термінологічних баз різних організацій, що підвищує рівень узгодженості у міжнародних ініціативах [162], [163].

Отже, впровадження уніфікованої моделі таксономії абревіатур є стратегічним завданням для підвищення ефективності комунікацій, зниження ризиків технічних і термінологічних помилок, а також створення основи для гармонізації галузевої термінології [164], [165], [166]. Реалізація цієї моделі потребує скоординованої співпраці між розробниками стандартів, науковою спільнотою, промисловими компаніями та регуляторними органами [167], [168], [169]. Перспективним

напрямом розвитку є інтеграція автоматизованих термінологічних сервісів, здатних у режимі реального часу оновлювати та перевіряти коректність скорочень у технічних і нормативних документах [170], [171], [172]. Подана схема ілюструє уніфіковану модель таксономії абревіатур у сферах кіберфізичних систем (КФС), штучних нейронних мереж (ШНМ) та інформаційної безпеки (ІБ). Актуальність такої моделі обумовлена зростаючою складністю сучасних технологічних систем, у яких використовується велика кількість спеціалізованих скорочень. Невпорядкованість терміносистеми призводить до плутанини у документації, підвищення ризику технічних помилок і зниження ефективності міжнародної співпраці. Стандартизований підхід до формування та управління абревіатурами дає змогу уникнути неоднозначностей і забезпечити узгодженість термінології в різних галузях та юрисдикціях. Метою запропонованої моделі є забезпечення однозначності, актуальності та міжнародної сумісності термінологічної бази шляхом формалізації процедур створення, рецензування, реєстрації та моніторингу скорочень. У рамках моделі кожна абревіатура проходить чітко визначений життєвий цикл: від ініціації пропозиції автором і верифікації її даних до внесення у реєстр з присвоєнням унікального ідентифікатора та статусу. Це дозволяє створити прозору систему відстеження еволюції кожного скорочення та його використання у технічній, нормативній і науковій документації. Візуалізація побудована у форматі алгоритму, де окремі блоки відображають ключові етапи процесу. На початковому рівні автор подає заявку з повним набором обов’язкових атрибутів: скорочення, розшифрування, домен застосування, мова, офіційне джерело та нормативне підґрунтя. Далі секретаріат і галузеві експерти (SME) здійснюють перевірку повноти й коректності даних, виявляють можливі дублікати або конфлікти та оцінюють відповідність скорочення встановленим критеріям. Після схвалення термін потрапляє до офіційного реєстру, де зберігається у структурованому вигляді та доступний через автоматизовані інтерфейси.Завершальним етапом є постійний моніторинг актуальності абревіатур, що передбачає відстеження змін у міжнародних та національних стандартах, а також у профільних галузевих регламентах. У разі виявлення оновлень терміни переводяться у статус повторного розгляду або застарілих (DEPRECATED) з подальшою заміною чи вилученням. Такий підхід забезпечує динамічну підтримку терміносистеми у відповідності до сучасних вимог, сприяє інтеграції автоматизованих термінологічних сервісів і підвищує якість міжгалузевої та міжнародної комунікації.

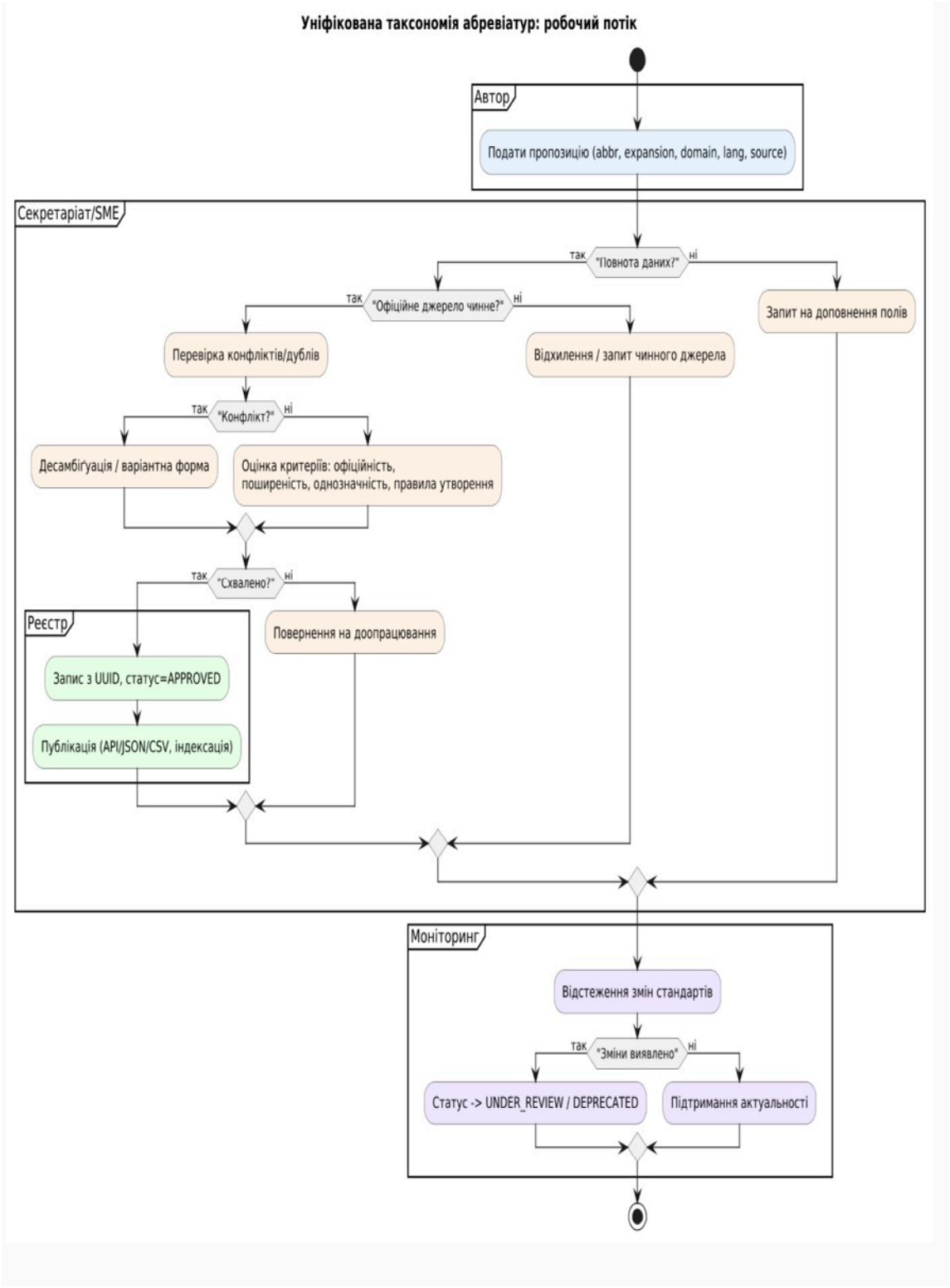


Рис. 2. Опис схеми

1. Блок «Автор» (синій колір) – початковий етап, де розробник або аналітик подає пропозицію нового скорочення чи оновлення наявного. У заявці вказуються ключові параметри: скорочення, розшифрування, галузь застосування, мова та офіційне джерело (наприклад, стандарт або нормативний документ).

2. Блок «Секретаріат/SME» (помаранчевий колір) – перевірка повноти та коректності поданих даних, валідація офіційності джерела, пошук можливих конфліктів або дублювань у реєстрі. За наявності збігів відбувається десамбіґуація або позначення варіантної форми. Далі експерти оцінюють відповідність скорочення встановленим критеріям (офіційність, поширеність, однозначність тощо).

3. Блок «Реєстр» (зелений колір) – після схвалення заявка переходить до етапу внесення у термінологічну базу. Запис отримує унікальний ідентифікатор, визначається статус (APPROVED), і відбувається публікація у відкритих форматах (JSON, CSV, RDF/SKOS) для інтеграції у зовнішні системи.

4. Блок «Моніторинг» (фіолетовий колір) – постійне відстеження змін у стандартах і нормативних документах. Якщо виявляються оновлення або застарілість терміна, він переводиться у статус UNDER_REVIEW або DEPRECATED, що запускає повторний цикл перевірки та оновлення.

Отже, схема відображає замкнений цикл керування абревіатурами, де забезпечується прозорий контроль на всіх етапах – від створення до виведення з ужитку. Модель дозволяє інтегруватися з автоматизованими термінологічними сервісами та підвищує якість міжгалузевої і міжнародної комунікації.

Програмна реалізація алгоритму (у вигляді попередніх пропозицій)

Програмне рішення уніфікованої моделі таксономії абревіатур у сферах кіберфізичних систем, штучних нейронних мереж та інформаційної безпеки передбачає створення керованого життєвого циклу скорочень із гарантіями однозначності, актуальності та міжнародної сумісності [153]. Система має підтримувати повний цикл обробки даних: подання пропозицій, автоматизовані перевірки, дедуплікацію та десамбіґуацію, рецензування експертами, внесення до реєстру, публікацію артефактів, індексацію для пошуку та моніторинг змін у стандартах [154]. Архітектура базується на сервісно-орієнтованій моделі з центральним модулем «Реєстр абревіатур» (реляційна база даних або тріпл-сховище), прикладним шаром для інтеграції через REST, GraphQL або SPARQL API, пошуковим індексом із лексичними та векторними алгоритмами, а також модулем керування процесами на основі workflow-схем [155]. Інгест пропозицій реалізується через вебінтерфейс або API, де автор надає скорочення, розшифрування, мову, домен застосування, офіційне джерело з DOI або URI, посилання на стандарт і контекст використання [156].

Модуль нормалізації виконує уніфікацію кодування Unicode NFC, перевірку регістру, видалення заборонених символів, а також валідацію обов’язкових полів. Для виявлення конфліктів застосовуються багаторівневі методи: точний збіг, нормалізовані збіги, нечітке порівняння (n-gram, Jaro–Winkler), а також оцінка семантичної близькості за

допомогою векторних подань [157]. Розв'язання колізій передбачає призначення статусів Preferred, Admitted або Deprecated, використання просторів назв чи кваліфікаторів для унікалізації [158]. Етап рецензування забезпечується експертами галузі, які оцінюють офіційність джерела, поширеність скорочення, його однозначність, відповідність правилам побудови акронімів та відсутність конфліктів [159]. Після схвалення запис вноситься до реєстру з присвоєнням унікального ідентифікатора, визначенням статусу та збереженням історії версій і змін [160].

Публікація даних здійснюється у форматах JSON, CSV, RDF/SKOS, з доступом через відкриті API, вебхуки та конектори до зовнішніх систем управління знаннями [161]. Пошукова система підтримує комбінований підхід із лексичними, нечіткими та векторними алгоритмами, враховуючи вагові коефіцієнти для ключових полів, таких як абревіатура, розшифрування, домен, мова, синоніми та теги [162]. Модуль моніторингу стандартів здійснює регулярне відстеження змін у міжнародних і національних нормативних джерелах (ISO, IEC, IEEE, NIST тощо) [163]. У разі виявлення оновлень пов'язані терміни переводяться у статус повторного розгляду або застарілих, що ініціює повторний цикл перевірки [164]. Передбачена інтеграція автоматизованих термінологічних сервісів для оперативного оновлення даних та валідації актуальності скорочень у режимі реального часу [165]. Такий підхід дозволяє забезпечити не лише стабільність і точність терміносистеми, але й високу швидкість адаптації до змін регуляторного середовища, що особливо важливо для галузей зі швидким розвитком, таких як кіберфізичні системи, штучні нейронні мережі та інформаційна безпека [166], [167], [168], [169], [170], [171], [172].

Покоміркова реалізація коду (Додаток 1. **«IDS CNN+LSTM на CICIDS2017» код Python)**

Наведений робочий зошит у Додатку 1 реалізує мінімально життєздатний прототип реєстру абревіатур із повним циклом: подання пропозицій, автоматичні перевірки якості та офіційності джерел, дедуплікацію в межах пари «домен-мова», рецензування (SME), публікацію артефактів у форматах JSON/CSV/RDF-TTL та індексацію для пошуку. Реалізовано базовий комбінований пошук (лексичний та векторний TF-IDF), можливе увімкнення семантичних ембеддингів (Sentence-BERT) за прапором USE_EMBEDDINGS. Передбачено тригер повторного перегляду термінів під час зміни стандарту з подальшим перевиданням записів. Для інтерактивного тестування включено мінімальний веб-інтерфейс на Gradio.

Додаток 5. Інструкція користувача веб-інтерфейсу MVP-реєстру абревіатур

Додаток 2 подає покрокову інструкцію користування веб-інтерфейсом MVP-реєстру абревіатур, реалізованого у середовищі Google Colab. У ньому

детально описано, як запустити інтерфейс, подати нове скорочення, виконати його перевірку та публікацію, а також як здійснювати пошук і перегляд результатів. Інструкція охоплює також пояснення форматів артефактів, що генеруються системою (JSON, CSV, RDF/SKOS), логіку статусів записів і сценарії вирішення конфліктів між абревіатурами. Додаток слугує практичним керівництвом для дослідників, розробників і адміністраторів термінологічних баз, що дозволяє швидко освоїти роботу з демо-версією та застосовувати її у тестових або пілотних проєктах.

Питання таксономії, які вирішує використання коду

Реалізований у коді механізм розв'язує низку ключових проблем таксономії абревіатур:

Уніфікація подання даних – автоматична нормалізація абревіатур (регістр, пробіли, символи) та мовних кодів забезпечує однозначність записів незалежно від автора.

Перевірка офіційності джерел – обов'язкова валідація source_uri і прив'язка до стандарту мінімізує включення неавторитетних термінів.

Дедуплікація та омонімія – алгоритми точного, нечіткого та (опційно) семантичного пошуку виявляють дублікати й конфлікти; для омонімів пропонуються кваліфікатори, що унеможливлює плутанину.

Версіонування та життєвий цикл – статуси DRAFT → UNDER_REVIEW → APPROVED → PUBLISHED дозволяють контролювати етапи опрацювання, зберігати історію рішень і змін.

Стандартизована публікація – автоматичний експорт у відкриті формати (JSON, CSV, RDF/SKOS) полегшує інтеграцію з іншими термінологічними сервісами.

Моніторинг актуальності – функція повторного відкриття термінів під час оновлення стандартів забезпечує динамічну підтримку бази в актуальному стані.

Таким чином, використання коду не лише спрощує введення та пошук абревіатур, але й упроваджує системний підхід до формування, підтримання та поширення таксономії у сферах кіберфізичних систем, штучних нейронних мереж та інформаційної безпеки.

4.2. Каталог абревіатур із розшифруванням та контекстною інтерпретацією

4.2.1. Абревіатури, пов'язані з кіберфізичними системами (КФС) у типологічній структурі

Розвиток кіберфізичних систем (КФС) супроводжується формуванням великого масиву спеціалізованих абревіатур, які використовуються в нормативно-технічній документації, наукових публікаціях та проєктних специфікаціях. Абревіатури виконують роль компактних ідентифікаторів,

що дозволяють стисло позначати складні технічні поняття, однак без належної систематизації вони стають джерелом неоднозначностей, особливо під час обміну даними між міжнародними партнерами. У цьому підрозділі подано каталог найбільш поширених скорочень, що належать до архітектурних, апаратних, програмних та комбінованих компонентів КФС, із розшифруванням і контекстною інтерпретацією.

Абревіатури, що позначають архітектурні концепції та компоненти КФС

CPS – Cyber-Physical Systems – кіберфізичні системи. Загальна категорія, що охоплює інтегровані комплекси, у яких фізичні процеси керуються та моніторяться за допомогою вбудованих обчислювальних засобів та мережевих з'єднань. У міжнародних стандартах CPS розглядаються як основа концепцій Індустрії 4.0, автономних транспортних засобів, розумних міст.

DT – Digital Twin – цифровий двійник. Віртуальна модель фізичного об'єкта, процесу чи системи, що синхронізується з реальним прототипом через потоки сенсорних даних. Використовується для симуляції, прогнозування станів і оптимізації роботи КФС.

IIoT – Industrial Internet of Things – промисловий Інтернет речей. Інфраструктурний підхід до з'єднання промислових сенсорів, контролерів та систем керування в єдину мережу для збору, аналізу та обміну даними в реальному часі.

SCADA – Supervisory Control and Data Acquisition – система диспетчерського керування та збору даних. Використовується у великих промислових КФС для моніторингу технологічних процесів і дистанційного керування обладнанням.

MES – Manufacturing Execution System – система диспетчеризації виробництва. Проміжний рівень між автоматизованими лініями та корпоративними системами планування ресурсів (ERP), забезпечує контроль виробничих процесів у реальному часі.

PLC – Programmable Logic Controller – програмований логічний контролер. Апаратний вузол для автоматизації керування технологічними процесами в КФС.

Абревіатури апаратного рівня КФС

MCU – Microcontroller Unit – мікроконтролер. Компактний вбудований обчислювальний пристрій із пам'яттю, процесором і периферією на одному кристалі.

FPGA – Field-Programmable Gate Array – програмована користувачем вентильна матриця. Дає можливість створювати апаратні конфігурації для спеціалізованих завдань у реальному часі.

ASIC – Application-Specific Integrated Circuit – спеціалізована інтегральна схема для конкретної функції або продукту КФС.

HMI – Human-Machine Interface – інтерфейс «людина-машина». Сукупність апаратних і програмних засобів, що дозволяють оператору взаємодіяти з КФС.

RTU – Remote Terminal Unit – віддалений термінальний модуль. Застосовується для збору даних і передавання команд між польовими пристроями та центральними вузлами керування.

Абревіатури програмного рівня КФС

RTOS – Real-Time Operating System – операційна система реального часу. Забезпечує детерміновану обробку подій і задач у часових межах, критично важливих для КФС.

API – Application Programming Interface – прикладний програмний інтерфейс, що надає стандартизований доступ до функцій програмних компонентів.

SDK – Software Development Kit – набір інструментів для розробки програмного забезпечення під конкретну платформу КФС.

CNC – Computer Numerical Control – числове програмне керування, застосовуване для автоматизації верстатів та іншого обладнання.

BMS – Battery Management System – система керування батареєю, яка оптимізує заряд/розряд і моніторить стан акумуляторних блоків.

Абревіатури мережевої інфраструктури КФС

LAN – Local Area Network – локальна мережа для обміну даними між компонентами КФС у межах одного об’єкта.

WAN – Wide Area Network – глобальна мережа, що об’єднує віддалені сегменти КФС.

VPN – Virtual Private Network – віртуальна приватна мережа для захищеної комунікації між віддаленими вузлами.

M2M – Machine-to-Machine – технології прямого обміну даними між пристроями без участі людини.

TSN – Time-Sensitive Networking – набір стандартів IEEE для забезпечення детермінованої комунікації з малими затримками у промислових мережах.

Комбіновані та інтегровані абревіатури

ICS – Industrial Control System – промислова система керування, що поєднує апаратні та програмні компоненти для моніторингу та контролю виробничих процесів.

DCS – Distributed Control System – розподілена система керування, у якій функції обробки сигналів і прийняття рішень розподілені між кількома контролерами.

CPS-SCADA – інтегрована архітектура, яка поєднує функції кіберфізичних систем і SCADA-рішень для критичної інфраструктури.

IoRT – Internet of Robotic Things – інтеграція робототехнічних систем в Інтернет речей із можливістю автономного прийняття рішень.

CPSoS – Cyber-Physical System of Systems – система систем кіберфізичного типу, у якій кілька незалежних КФС взаємодіють для досягнення спільних цілей.

Контекстна інтерпретація та приклади застосування

Більшість перерахованих абревіатур мають міжгалузеву природу, однак у контексті КФС вони набувають специфічного значення. Наприклад, DT у поєднанні з IIoT використовується для побудови адаптивних моделей виробничих ліній, що підвищує ефективність і знижує витрати на технічне обслуговування. RTOS у середовищі PLC гарантує своєчасне виконання критичних завдань управління, а інтеграція TSN із SCADA дає змогу досягти синхронізації сигналів у розподілених промислових мережах. Упровадження чіткої типологічної структури абревіатур дає змогу підвищити якість технічної документації, спрощує процеси розробки та інтеграції систем, а також сприяє гармонізації термінології між різними учасниками проєктів. Для цього доцільно використовувати уніфіковані реєстри скорочень, інтегровані з автоматизованими засобами валідації та пошуку, приклад яких подано у програмній реалізації (Додаток 2).

Додаткові абревіатури з міжнародних стандартів IEC, ISO та IEEE

IEC 61850 – стандарт Communication networks and systems for power utility automation (IEC). Регламентує модель даних та протоколи обміну для систем автоматизації електроенергетичних підприємств, що є різновидом КФС.

ISO 23247 – Automation systems and integration – Digital Twin framework for manufacturing. Визначає концептуальну модель цифрового двійника для виробничих КФС.

IEEE 1451 – Standard for a Smart Transducer Interface. Описує інтерфейси для сенсорів і виконавчих пристроїв у розподілених кіберфізичних мережах.

IEC 62541 – OPC Unified Architecture (OPC UA). Стандарт описує платформонезалежну архітектуру для промислових комунікацій у КФС, з підтримкою моделі інформаційних об’єктів та безпеки.

ISO/IEC 30141 – Internet of Things Reference Architecture. Містить модель, сумісну з IIoT та КФС, для забезпечення інтероперабельності, безпеки та масштабованості систем.

IEEE 802.1AS – стандарт синхронізації часу в мережах з низькою затримкою, який є складовою TSN і критично важливим для реального часу у КФС.

IEC 61131-3 – стандарт мов програмування для програмованих логічних контролерів (PLC), що визначає єдиний синтаксис і семантику програм для промислової автоматизації.

Приклади з міжнародних проєктів та технічних специфікацій

Проєкт Industry 4.0 Testbed (Німеччина)

У межах цього тестового стенду CPS інтегруються з OPC UA, TSN і DT, щоб створити повністю синхронізовану цифрову копію виробничої лінії. Застосування IEEE 802.1AS забезпечує точність синхронізації часу на рівні мікросекунд, що дозволяє оптимізувати логістику та автоматичне переналаштування обладнання.

Проєкт Smart Grid Demonstrator (США)

Використання стандартів IEC 61850 і IEEE 1451 дає можливість інтегрувати розподілені енергетичні ресурси, такі як сонячні панелі та системи зберігання енергії (BMS), у єдину мережу керування. КФС у цьому випадку об’єднує польові сенсори, PLC, SCADA та цифрові двійники для прогнозування навантаження.

Проєкт Autonomous Port Operations (Сінгапур)

У рамках автоматизації контейнерних терміналів застосовується комбінація IIoT, IoRT, DCS і RTOS у керуванні автономними транспортними засобами, кранами та системами відстеження контейнерів. Протоколи OPC UA та мережі з підтримкою TSN забезпечують низькі затримки під час обміну даними між критичними вузлами.

Проєкт Cyber-Physical Laboratory for Railway Automation (ЄС)

Розгорнуто КФС із використанням PLC, RTU, SCADA та стандартів IEC 62541 та ISO 23247 для управління інтелектуальними залізничними системами. Цифровий двійник дозволяє відтворювати сценарії руху поїздів, прогнозувати зношення колій та оптимізувати графіки технічного обслуговування.

Контекстні особливості термінів у галузі КФС

Деякі абревіатури, хоча і є загальноприйнятими, у КФС мають розширену або модифіковану інтерпретацію:

API у контексті КФС часто включає специфічні промислові функції, пов'язані з роботою із сенсорними мережами та PLC.

VPN у промислових мережах може реалізовуватися з використанням протоколів, оптимізованих для низьких затримок і високої надійності, що є критичним для SCADA-систем.

MCU у КФС часто має вбудовані функції для апаратної підтримки протоколів польових шин (наприклад, Modbus, Profibus, CAN).

Значення уніфікації абревіатур у міжнародних КФС-проєктах

Ураховуючи те, що КФС інтегрують апаратні, програмні та мережеві підсистеми з різних галузей, відсутність уніфікованого підходу до абревіатур призводить до:

помилок у тлумаченні технічних вимог;

збоїв в інтеграції обладнання через різні трактування термінів у документації;

затримок у сертифікації та аудиті відповідності стандартам.

Упровадження централізованих реєстрів, таких як описаний у Додатку 2, дає можливість забезпечити:

автоматичну перевірку коректності та актуальності абревіатур;

швидкий пошук ідентичних чи подібних скорочень у великій термінологічній базі;

прив'язку кожного скорочення до конкретних стандартів IEC, ISO, IEEE або галузевих специфікацій.

Наведена нижче табл. 2 є узагальненим результатом систематизації найпоширеніших абревіатур, що використовуються у сфері кіберфізичних систем (КФС). Вона сформована на основі міжнародних стандартів ISO, IEC, IEEE, а також галузевих специфікацій і технічних регламентів. До таблиці включено скорочення, які охоплюють апаратні, програмні, мережеві та інтегровані компоненти КФС, з обов'язковим зазначенням їх повної назви, офіційного джерела (за наявності) та контекстної інтерпретації у галузі. Мета цієї демонстрації – показати, як структуроване подання термінів уніфікує їх використання в науково-технічній комунікації, полегшує впровадження автоматизованих механізмів пошуку і валідації, а також сприяє уникненню неоднозначностей у міжгалузевій та міжнародній взаємодії.

Таблиця 2

Абревіатури, пов'язані з кіберфізичними системами (КФС)

Абревіатура	Повна назва	Джерело / стандарт	Контекстна інтерпретація у КФС
CPS	Cyber-Physical Systems	ISO/IEC 30141	Інтегровані комплекси, де фізичні процеси керуються вбудованими обчислювальними системами та мережами
DT	Digital Twin	ISO 23247	Віртуальна модель фізичного об'єкта або процесу, що синхронізується з реальним через сенсори
IIoT	Industrial Internet of Things	ISO/IEC 30141	Інфраструктура для з'єднання промислових пристроїв і сенсорів у мережі
SCADA	Supervisory Control and Data Acquisition	IEC 60870, IEC 61850	Система диспетчерського керування та збору даних у промислових КФС

Продовженя табл. 2

MES	Manufacturing Execution System	ISA-95	Проміжний рівень управління виробництвом між ERP і автоматизованими лініями
PLC	Programmable Logic Controller	IEC 61131-3	Програмований логічний контролер для автоматизації технологічних процесів
MCU	Microcontroller Unit	–	Вбудований мікроконтролер із процесором, пам’яттю та периферією
FPGA	Field-Programmable Gate Array	–	Програмована вентильна матриця для спеціалізованої обробки в реальному часі
ASIC	Application-Specific Integrated Circuit	–	Інтегральна схема для виконання конкретних функцій
HMI	Human-Machine Interface	–	Апаратно-програмний інтерфейс для взаємодії оператора з КФС
RTU	Remote Terminal Unit	IEC 60870	Віддалений термінал для збору даних і передавання команд
RTOS	Real-Time Operating System	–	ОС, що забезпечує детерміновану обробку подій у часових рамках
API	Application Programming Interface	–	Стандартизований інтерфейс доступу до програмних функцій
SDK	Software Development Kit	–	Набір інструментів для розробки ПЗ під платформу КФС

Продовження табл. 2

CNC	Computer Numerical Control	ISO 6983	Числове програмне керування верстатами
BMS	Battery Management System	IEC 62660, ISO 26262	Система керування акумуляторами
LAN	Local Area Network	IEEE 802.3	Локальна мережа для обміну даними між компонентами КФС
WAN	Wide Area Network	ITU-T G.805	Глобальна мережа для з’єднання віддалених сегментів
VPN	Virtual Private Network	RFC 4026 та ін.	Захищена комунікація між вузлами
M2M	Machine-to-Machine	ETSI TS 102 690	Прямий обмін даними між пристроями
TSN	Time-Sensitive Networking	IEEE 802.1Qbv/802.1AS	Мережеві стандарти для детермінованої комунікації з малими затримками
ICS	Industrial Control System	NIST SP 800-82	Промислова система керування
DCS	Distributed Control System	IEC 61131	Розподілена система керування
CPS-SCADA	–	Концептуальна інтеграція	Об’єднання архітектур КФС і SCADA
IoRT	Internet of Robotic Things	–	Інтеграція робототехніки у IoT з автономним прийняттям рішень
CPSoS	Cyber-Physical System of Systems	–	Система систем КФС, що взаємодіють між собою

Закінчення табл. 2

IEC 61850	Communication networks and systems for power utility automation	IEC 61850	Стандарт для енергетичних автоматизованих систем
IEEE 1451	Smart Transducer Interface Standards	IEEE 1451	Інтерфейси для сенсорів і виконавчих пристроїв
IEC 62541	OPC Unified Architecture	IEC 62541	Платформонезалежна архітектура промислових комунікацій
ISO/IEC 30141	Internet of Things Reference Architecture	ISO/IEC 30141	Референсна архітектура IoT і КФС
IEEE 802.1AS	Timing and Synchronization for Time-Sensitive Applications	IEEE 802.1AS	Синхронізація часу у мережах з низькою затримкою
IEC 61131-3	Programmable Controllers – Part 3: Programming Languages	IEC 61131-3	Мови програмування для PLC
ISO 23247	Digital Twin framework for manufacturing	ISO 23247	Структура цифрових двійників у виробництві

Таблиця 2 містить чотири основні стовпці:

Абревіатура – коротке позначення терміна, що використовується у текстах нормативних документів, наукових публікацій та технічних специфікацій.

Повна назва – офіційне або загальноприйняте розшифрування абревіатури.

Джерело / стандарт – посилання на міжнародний чи галузевий стандарт або специфікацію, у якій закріплене визначення терміна.

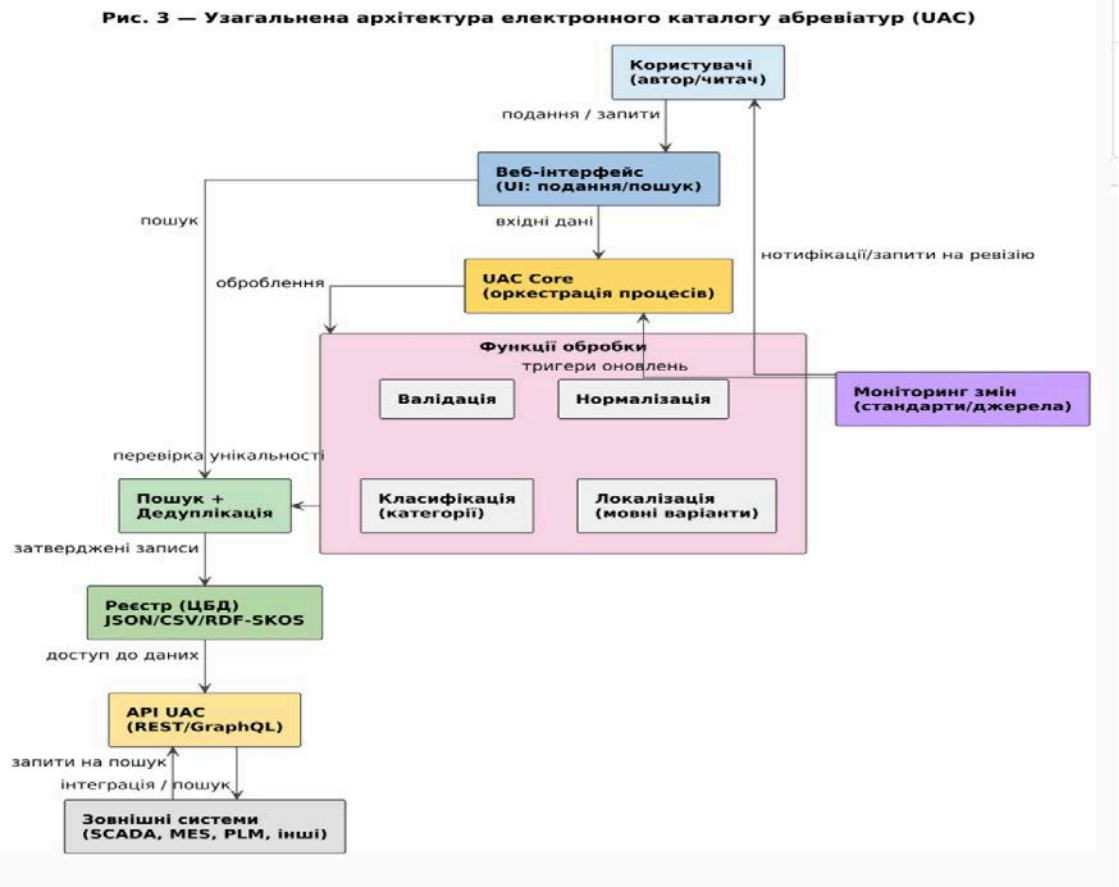
Контекстна інтерпретація у КФС – пояснення ролі та застосування відповідного поняття у структурі та функціоналі кіберфізичних систем.

Добір скорочень проводився за критеріями офіційності, поширеності у професійних джерелах та однозначності тлумачення в межах галузі.

Пропозиція до каталогізації

Для подальшої інтеграції цієї інформації у єдиний термінологічний простір доцільно розгорнути представлену таблицю в електронний каталог абревіатур, що матиме такі функції:

1. Класифікація за тематичними підкатегоріями – апаратні, програмні, мережеві, інтегровані, нормативно-стандартні.
2. Прив'язка до версій стандартів – зазначення дати та редакції стандарту, в якому актуалізоване визначення.
3. Мовна локалізація – зберігання відповідників кількома мовами для міжнародних проєктів.
4. Автоматизований пошук і дедуплікація – виявлення дублюючих або конфліктних скорочень у межах одного домену.
5. API-доступ – надання уніфікованого інтерфейсу для інтеграції з документними системами, SCADA, MES, PLM та іншими цифровими платформами КФС



Реалізація такої каталогізації уможливить створення живої термінологічної бази, яка буде актуалізуватися у режимі реального часу та забезпечить прозору комунікацію між усіма учасниками процесів розробки, експлуатації та стандартизації кіберфізичних систем.

Подана схема відображає архітектурну модель електронного каталогу абревіатур, призначеного для уніфікації та систематизації

скорочень у сферах кіберфізичних систем (КФС), штучних нейронних мереж (ШНМ) та інформаційної безпеки (ІБ).

Метою розробки такої моделі є забезпечення однозначності, актуальності та міжнародної сумісності терміносистеми шляхом централізованого збору, валідації, класифікації та поширення скорочень. Схема візуалізує ключові функціональні блоки та взаємозв'язки між ними, показуючи, як дані рухаються від користувацького інтерфейсу до ядра системи та зовнішніх інтеграційних точок (рис. 3). Запропонована схема відображає узагальнену архітектуру електронного каталогу абревіатур, призначеного для уніфікації та систематизації скорочень у сферах кіберфізичних систем, штучних нейронних мереж та інформаційної безпеки. Веб-інтерфейс забезпечує взаємодію користувачів із системою – подання нових скорочень та виконання пошукових запитів. Усі вхідні дані надходять до ядра UAC, яке відповідає за оркестрацію процесів оброблення. Усередині ядра виділено модуль функцій обробки, що включає валідацію даних, їх нормалізацію, класифікацію за тематичними категоріями та локалізацію мовних варіантів.

Далі інформація передається до підсистеми пошуку та дедуплікації, яка забезпечує виявлення дублікатів і перевірку унікальності записів. Після цього затверджені дані зберігаються в реєстрі – централізованій базі даних, яка є основним джерелом актуальної інформації. Доступ до даних реєстру забезпечується через API, що підтримує інтеграцію із зовнішніми системами, такими як SCADA, MES або PLM. Пошукові функції можуть виконуватись як безпосередньо через інтерфейс користувача, так і через API для зовнішніх клієнтів.

Така архітектура дозволяє забезпечити централізований контроль якості даних, їх уніфікацію та доступність для різних інформаційних систем у рамках міжнародних стандартів. У комплексі дана схема ілюструє замкнений цикл управління абревіатурами – від моменту подання до інтеграції у зовнішні цифрові платформи, з акцентом на автоматизації перевірок, підтриманні актуальності та забезпеченні прозорості процесів.

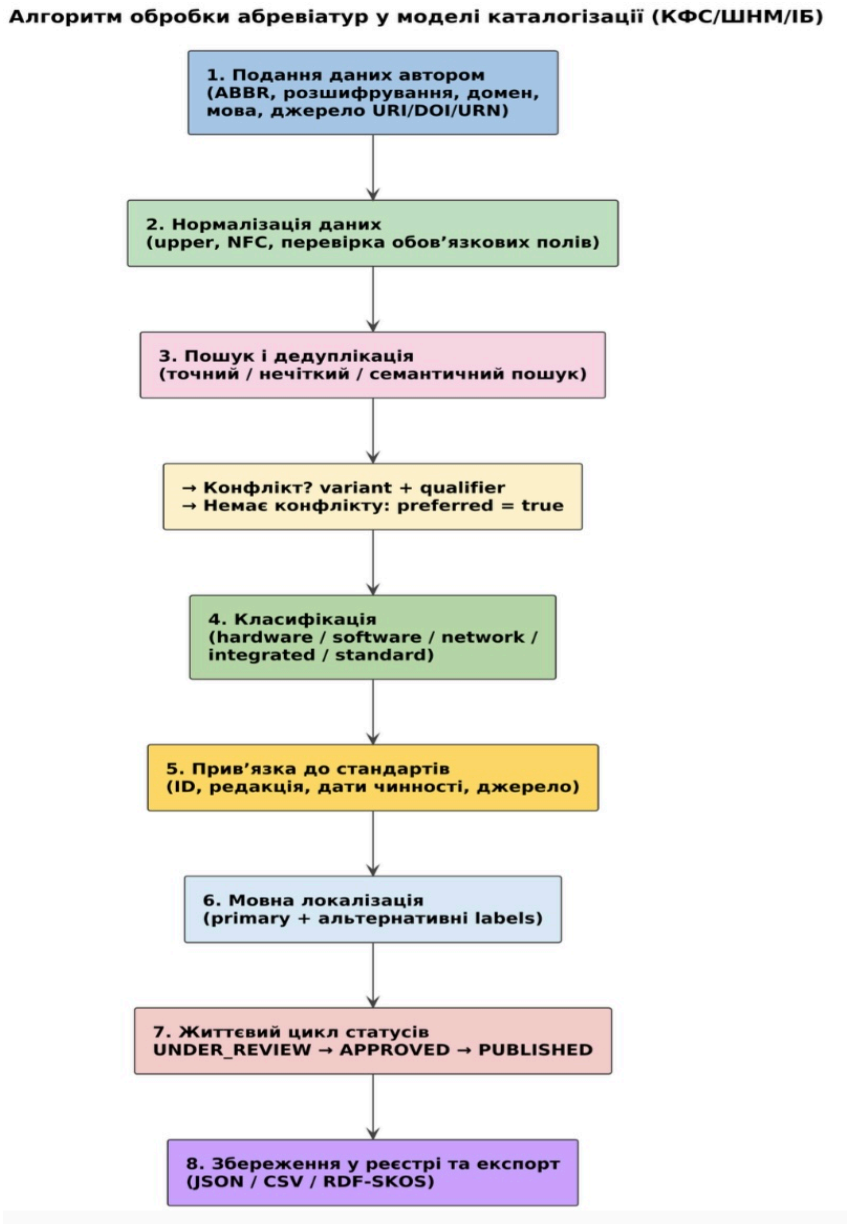


Рис. 4. Опис схеми

Запропонована на рис. 4 схема відображає алгоритм обробки абревіатур у межах уніфікованої моделі каталогізації для кіберфізичних систем (КФС), штучних нейронних мереж (ШНМ) та інформаційної безпеки (ІБ). Метою цього алгоритму є забезпечення чіткого, стандартизованого життєвого циклу скорочень – від моменту подання користувачем до внесення у реєстр і публікації у відкритих форматах. Процес передбачає послідовне виконання перевірок, класифікації, прив'язки до стандартів і мовної локалізації, а також автоматизовану дедуплікацію для запобігання дублюванню та омонімії.

1. Подання даних автором

Користувач вводить абревіатуру (ABBR), її розшифрування, домен застосування, мову та офіційне джерело (URI/DOI/URN). Паралельно виконується перевірка форми на відповідність ключовим вимогам:

коректність довжини та символів, наявність primary-locale, офіційність посилання на джерело.

2. Нормалізація даних

Абревіатура приводиться до верхнього регістру (upper) та уніфікованої форми кодування (NFC), перевіряється заповненість обов'язкових полів.

3. Пошук і дедуплікація

Виконується точний, нечіткий або семантичний пошук у базі для виявлення дублів та омонімів.

- Якщо виявлено конфлікт – запис позначається як variant із доданим qualifier.
- Якщо конфлікту немає – встановлюється прапорець preferred = true.

4. Класифікація

Терміни належать до однієї з підкатегорій: hardware, software, network, integrated, standard.

5. Прив'язка до стандартів

Зберігаються ідентифікатор стандарту, редакція, дати чинності та джерело.

6. Мовна локалізація

Записуються мовні варіанти абревіатури (primary та альтернативні labels).

7. Зміна статусу та публікація

Запис проходить через життєвий цикл статусів: UNDER_REVIEW → APPROVED → PUBLISHED.

8. Збереження у реєстрі та експорт

Фінальні дані зберігаються в реєстрі та експортуються у форматах JSON, CSV або RDF-SKOS для інтеграції в інші системи.

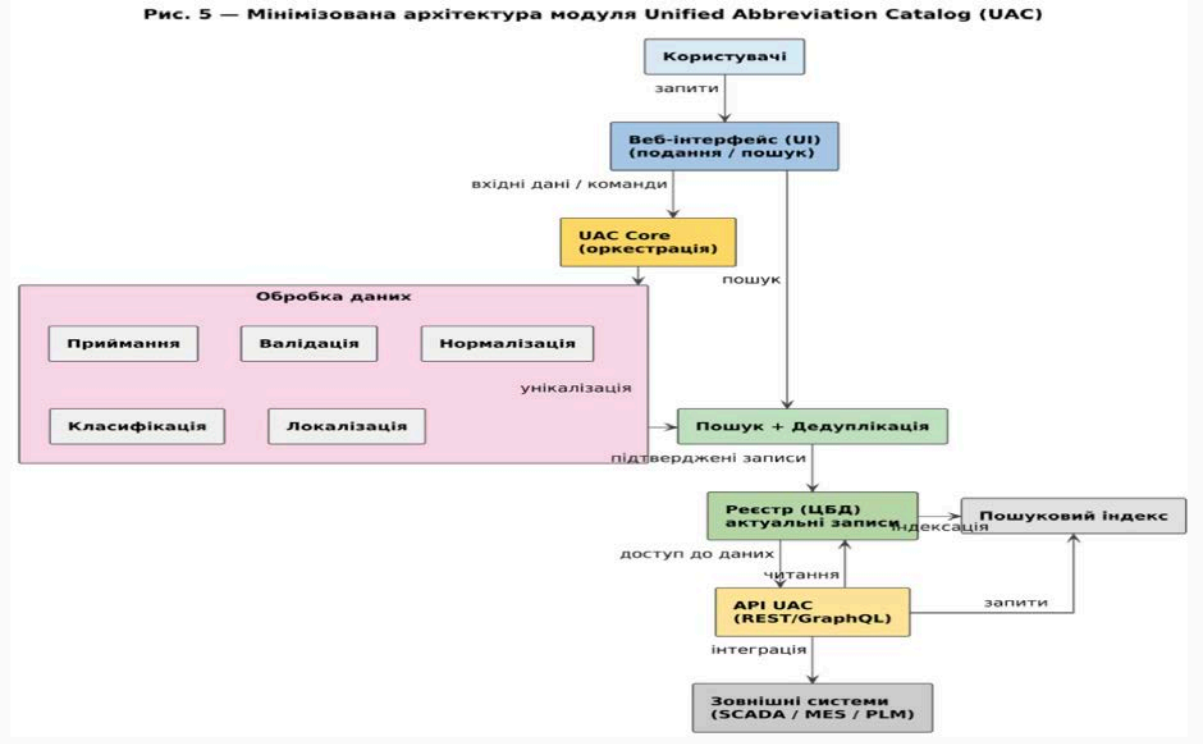
Програмний модуль Unified Abbreviation Catalog (UAC)

призначений для створення, зберігання, підтримання та поширення стандартизованих скорочень з урахуванням міжнародних норм, галузевих регламентів і вимог до міжсистемної сумісності у сферах кіберфізичних систем (КФС), штучних нейронних мереж (ШНМ) та інформаційної безпеки (ІБ). Його метою є забезпечення повного життєвого циклу скорочень від моменту подання пропозиції автором до інтеграції перевірених і затверджених термінів у зовнішні цифрові системи. Такий підхід дозволяє мінімізувати термінологічні конфлікти, підвищити ефективність комунікацій та прискорити інтеграцію рішень у міжнародні проєкти. Архітектурна модель модуля передбачає веб-інтерфейс, через який користувач подає нові абревіатури, здійснює пошук, редагування та перегляд інформації, бізнес-логіку обробки, що реалізує нормалізацію скорочень, автоматичну класифікацію, дедуплікацію та призначення статусів життєвого циклу, а також API-інтерфейси для взаємодії з іншими платформами та механізми експорту даних у форматах JSON, CSV, RDF-SKOS. Функціональні можливості UAC охоплюють подачу та валідацію даних з автоматичною перевіркою довжини та допустимих символів, наявності основного мовного варіанта (primary-locale) та офіційності

джерела (URI/DOI/URN), нормалізацію абревіатур до уніфікованої форми з використанням верхнього регістру та кодування Unicode NFC, пошук і дедуплікацію з виявленням дублів та омонімів за допомогою точного, нечіткого або семантичного пошуку, класифікацію термінів з віднесенням до однієї з категорій (hardware, software, network, integrated, standard), прив'язку до стандартів із збереженням ідентифікатора, редакції, дат чинності та джерела, підтримку мовної локалізації з можливістю зберігати основні та альтернативні варіанти назв, управління статусами від перевірки до публікації та застарівання, а також експорт і інтеграцію з зовнішніми системами.

Технічна реалізація модуля може бути виконана з використанням Python (FastAPI, SQLAlchemy, rdflib) або Node.js (NestJS) із застосуванням PostgreSQL для реляційного підходу або GraphDB/Fuseki для семантичного зберігання, Elasticsearch/OpenSearch для індексації пошуку, React або Vue.js з підтримкою інтернаціоналізації на фронтенді, а також Keycloak або OAuth 2.0/JWT для керування доступом.UAC передбачає автоматизацію та моніторинг, що включає планові перевірки актуальності стандартів, автоматичне оновлення статусів термінів при зміні версій стандартів та генерацію аналітичних звітів, зокрема статистики додавання термінів, рівня дублікатів та розподілу за категоріями і мовами. Реалізація модуля Unified Abbreviation Catalog створює постійно актуалізовану термінологічну базу, інтегровану у наукові, промислові та нормативні процеси, що знижує ризики термінологічних помилок і підвищує ефективність інформаційної взаємодії у міжнародних проєктах.

Рис. 5 — Мінімізована архітектура модуля Unified Abbreviation Catalog (UAC)



Наведена схема ілюструє архітектурну модель програмного модуля Unified Abbreviation Catalog (UAC), призначеного для централізованого

управління абревіатурами у сферах кіберфізичних систем (КФС), штучних нейронних мереж (ШНМ) та інформаційної безпеки (ІБ). Модуль реалізує повний життєвий цикл роботи зі скороченнями – від подання користувачем до інтеграції стандартизованих термінів у зовнішні цифрові системи. Схема демонструє ключові функціональні компоненти, їх взаємодію та інформаційні потоки між етапами обробки даних. Основна мета цієї архітектури полягає у забезпеченні однозначності, актуальності та сумісності термінології, що використовується у багатокомпонентних і міжгалузевих проєктах.

Схема (рис. 5) відображає мінімізовану архітектурну модель програмного модуля Unified Abbreviation Catalog (UAC), призначеного для централізованого управління абревіатурами у сферах кіберфізичних систем (КФС), штучних нейронних мереж (ШНМ) та інформаційної безпеки (ІБ). Веб-інтерфейс (UI) є точкою взаємодії користувачів із системою, через яку здійснюється подання нових скорочень та пошук існуючих. Усі запити спрямовуються до ядра UAC, що відповідає за оркестрацію процесів. Усередині ядра функціонує блок обробки, який охоплює основні етапи роботи з даними: приймання, валідацію, нормалізацію, класифікацію та локалізацію. Після первинної обробки записи проходять через підсистему пошуку та дедуплікації, що забезпечує виявлення дубльованих або схожих записів і формування унікальних сутностей. Підтверджені та унікальні записи зберігаються у реєстрі, який є основною базою даних системи. Для підвищення швидкодії пошуку використовується окремий пошуковий індекс. Доступ до даних реєстру та інтеграція з іншими цифровими платформами (наприклад, SCADA, MES, PLM) здійснюється через API, що підтримує стандартизовані протоколи обміну. Архітектура забезпечує повний життєвий цикл роботи з абревіатурами – від подання користувачем до їх використання у зовнішніх системах, гарантує однозначність, актуальність і сумісність термінології в міжгалузевих проєктах.

4.2.2. Абревіатури в управлінні ризиками: класифікація моделей та методів

Управління ризиками в сучасних кіберфізичних системах (КФС) і критичній інфраструктурі потребує не лише використання технічних засобів моніторингу та реагування, але й формалізованих підходів до оцінювання, аналізу та зниження ризиків. У науково-технічній комунікації значну роль відіграють абревіатури, що позначають відомі моделі та методи управління ризиками, розроблені на основі міжнародних стандартів, галузевих вимог та дослідницьких напрацювань. Їх коректне застосування забезпечує уніфікацію термінології, скорочення часу на міждисциплінарну взаємодію та підвищення точності у формулюванні завдань безпеки.

Класифікація моделей

Серед найбільш поширених моделей управління ризиками у сфері інформаційної безпеки та КФС можна виокремити п’ять системних підходів, що набули міжнародного значення:

ERM (Enterprise Risk Management) – інтегрована модель управління ризиками підприємства, що передбачає оцінку ризиків на всіх рівнях організації, включаючи стратегічний, операційний та технологічний. ERM орієнтована на постійний моніторинг, визначення пріоритетів та інтеграцію управління ризиками в процеси прийняття рішень.

FAIR (Factor Analysis of Information Risk) – методологія кількісного аналізу ризиків інформаційної безпеки, яка базується на оцінюванні ймовірності та фінансових наслідків інцидентів. Особливістю FAIR є можливість інтеграції з економічними моделями та інструментами страхування кіберризиків.

OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) – підхід, розроблений для самооцінювання безпеки організаціями, який фокусується на критичних активах, їх загрозах і вразливостях. OCTAVE застосовується переважно у сферах з високими вимогами до безперервності бізнес-процесів.

CRAMM (CCTA Risk Analysis and Management Method) – методика аналізу та управління ризиками, розроблена у Великій Британії, що передбачає поетапне визначення активів, загроз і заходів захисту з урахуванням вартості та критичності.

EBIOS (Expression des Besoins et Identification des Objectifs de Sécurité) – французький підхід до управління ризиками, орієнтований на формалізацію вимог до безпеки та визначення цілей захисту з урахуванням контексту використання системи.

Прив’язка до стандартів

Названі моделі тісно пов’язані з міжнародними та національними стандартами. Зокрема, ISO 31000:2018 («Risk Management – Guidelines») визначає загальні принципи та процеси управління ризиками, які можуть бути реалізовані через ERM або інтегровані з OCTAVE. NIST SP 800-30 («Guide for Conducting Risk Assessments») надає методичні рекомендації з оцінювання ризиків у галузі інформаційної безпеки, зокрема застосовних у КФС. FAIR часто використовується у поєднанні з підходами NIST для формалізації кількісних параметрів ризику.

Приклади застосування у КФС та критичній інфраструктурі

У сфері енергетики

OCTAVE застосовується для визначення ризиків у системах SCADA, де критичними є доступність і точність телеметричних даних. FAIR дозволяє оцінити фінансові наслідки атак на системи автоматизації енергопостачання, включаючи збитки від простоїв та витрат на відновлення.

У транспортних

КФС (інтелектуальні транспортні системи, авіаційна навігація) EBIOS дає змогу формалізувати вимоги безпеки, враховуючи специфіку нормативної бази ЄС та національних регуляторів.

У промислових комплексах

CRAMM використовується для оптимізації витрат на кіберзахист, порівнюючи вартість реалізації захисних заходів з потенційними збитками від інцидентів.

ERM у КФС часто інтегрують із корпоративними платформами управління, що дозволяє поєднувати моніторинг ризиків з фінансовим плануванням і управлінням ланцюгами постачання.

Порівняльний аспект

З точки зору практичного впровадження у КФС, можна запропонувати узагальнену класифікаційну таблицю.

Таблиця 3

Класифікація моделей управління ризиками та їх відповідність стандартам

Абревіатура	**Основна сфера застосування**	**Орієнтація**	**Тип оцінки**	**Відповідні стандарти**
ERM	Корпоративне управління	Стратегічна	Якісна + кількісна	ISO 31000
FAIR	Кібербезпека, фінанси	Тактична	Кількісна	NIST SP 800-30
OCTAVE	Критична інфраструктура	Операційна	Якісна	ISO 27005
CRAMM	Промислові системи	Операційна	Якісна + частково кількісна	ISO 27001
EBIOS	Регульовані сектори	Тактична	Якісна	ENISA, ISO 27005

Застосування зазначених моделей у КФС та критичній інфраструктурі доводить, що абревіатури не є лише скороченнями, а відображають комплексні методологічні підходи, безпосередньо пов'язані зі стандартами та практикою управління ризиками. Їх коректне впровадження сприяє підвищенню кіберстійкості, зниженню ймовірності інцидентів та забезпеченню відповідності нормативним вимогам.

В умовах цифровізації економіки та зростання кількості гібридних кібератак на кіберфізичні системи (КФС) особливої актуальності набуває гармонізація методів управління ризиками з міжнародними стандартами. Ефективна інтеграція моделей, таких як ERM, FAIR, OCTAVE, CRAMM та EBIOS, із загальновизнаними нормативними документами (ISO 31000, NIST SP 800-30, ISO 27001, ISO 27005) дає можливість підвищити стійкість систем, забезпечити порівнянність результатів оцінювання та відповідність регуляторним вимогам. Візуалізація, подана на схемі (рис. 6), демонструє взаємозв'язок між обраними моделями управління ризиками, міжнародними стандартами та рекомендаціями профільних організацій, таких як ENISA.

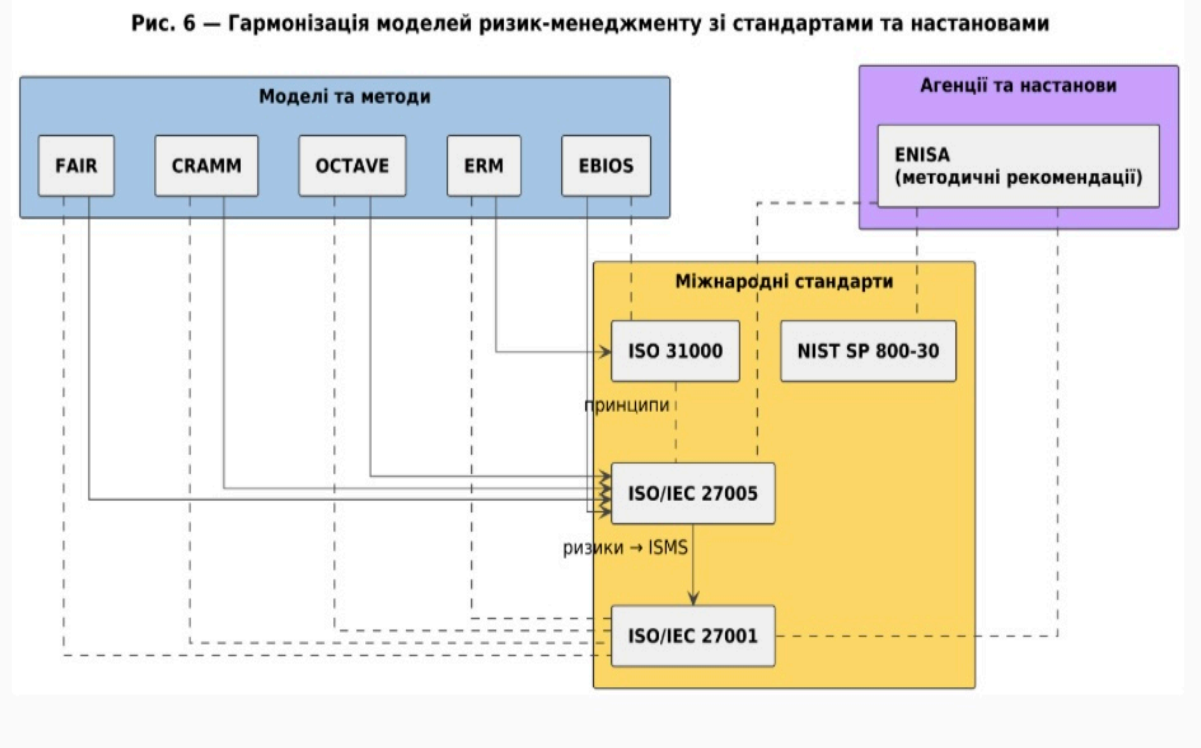


Рис. 6 — Гармонізація моделей ризик-менеджменту зі стандартами та настановами

Рис. 6. Схема структурована на три основні блоки:

1. Моделі та методи (блакитний блок) – мають п'ять ключових підходів:

ERM (Enterprise Risk Management) – стратегічна модель управління ризиками на рівні підприємства.

FAIR (Factor Analysis of Information Risk) – кількісна методологія оцінювання ризиків інформаційної безпеки.

OCTAVE (Operationally Critical Threat, Asset and Vulnerability Evaluation) – операційний підхід до аналізу загроз і вразливостей.

CRAMM (CCTA Risk Analysis and Management Method) – поетапна методика ідентифікації активів, загроз і контрзаходів.

EBIOS (Expression des Besoins et Identification des Objectifs de Sécurité) – формалізація вимог і цілей безпеки у специфічних секторах.

2. Міжнародні стандарти (жовтий блок) – включає чотири основних документи, що задають вимоги та керівні принципи управління ризиками:

ISO 31000 – загальні принципи управління ризиками.

NIST SP 800-30 – методичні рекомендації щодо оцінювання ризиків у сфері кібербезпеки.

ISO/IEC 27001 – вимоги до системи управління інформаційною безпекою.

ISO/IEC 27005 – керівництво з управління ризиками інформаційної безпеки.

3. Агенції та настанови (фіолетовий блок) – представлено прикладом ENISA, яка розробляє методичні рекомендації та кращі практики для критичної інфраструктури та промислових КФС.

Стрілки відображають:

Суцільні зв'язки – пряму методологічну або нормативну відповідність моделі стандарту.

Пунктирні зв'язки – допоміжні чи контекстні зв'язки, коли модель використовує додаткові підходи або орієнтується на рекомендації сторонніх органів.

Завдяки такій структурі схема дає змогу швидко зрозуміти, яка модель управління ризиками узгоджується з певними міжнародними стандартами, та простежити, як вони взаємодіють між собою у контексті забезпечення кіберстійкості КФС та критичної інфраструктури.

4.2.3. Абревіатури штучних нейронних мереж (ШНМ) у структуризаційній моделі

У сучасних кіберфізичних системах (КФС) та критичній інфраструктурі штучні нейронні мережі (ШНМ) відіграють ключову роль у розв'язанні задач обробки, аналізу та прогнозування даних [153]. Розвиток цих технологій супроводжується формуванням великого масиву абревіатур, що позначають архітектури, алгоритмічні категорії та методи оптимізації. Їх чітке визначення та систематизація необхідні для забезпечення однозначності інтерпретації та уніфікації в науковій, інженерній і нормативно-технічній комунікації [154]. Використання абревіатур дає можливість стисло передавати складні концепції, але без належної таксономії вони можуть стати джерелом неоднозначностей,

особливо у міжгалузевих проєктах [155]. З огляду на швидкий розвиток архітектур ШНМ, таких як MLP, CNN, RNN, LSTM, GRU, GAN та AE, а також упровадження нових підходів до навчання, виникає необхідність у формалізованому підході до їх класифікації [156].

Архітектури ШНМ: ключові абревіатури

MLP (Multilayer Perceptron) – багатошаровий перцептрон, базова архітектура глибинного навчання [157]. Складається з вхідного шару, одного або кількох прихованих шарів та вихідного шару з нелінійними функціями активації. MLP застосовується для задач регресії, класифікації та обробки табличних даних.

CNN (Convolutional Neural Network) – згорткова нейронна мережа, призначена для виявлення просторових залежностей у структурованих даних [158]. Використовує *згорткові та підвибіркові шари для виділення ознак, особливо ефективна у розпізнаванні* зображень, відео та обробці сигналів.

RNN (Recurrent Neural Network) – рекурентна нейронна мережа, здатна зберігати стан між обчислювальними кроками, що дозволяє обробляти послідовні дані [159]. Використовується у прогнозуванні часових рядів, обробці природної мови та аналізі потокових даних.

LSTM (Long Short-Term Memory) – модифікація RNN, що розв'язує проблему зникання або вибуху градієнтів [160]. Використовує комірки пам'яті та механізми керування потоками даних (input, forget, output gates), що дозволяє зберігати залежності на довгих інтервалах.

GRU (Gated Recurrent Unit) – спрощений варіант LSTM з меншою кількістю параметрів [161]. Підходить для задач у реальному часі та мобільних застосунків, де обчислювальні ресурси обмежені.

GAN (Generative Adversarial Network) – генеративно-змагальна мережа, що складається з генератора та дискримінатора [162]. Використовується для синтезу зображень, створення реалістичних симуляцій та генерації даних для рідкісних сценаріїв.

AE (Autoencoder) – автоенкодер, який виконує кодування вхідних даних у компактне представлення та відновлює їх на виході [163]. Застосовується для зменшення розмірності, виявлення аномалій та попереднього навчання інших моделей.

Алгоритмічні категорії навчання

Класифікація ШНМ за підходами до навчання дозволяє визначити тип задач, у яких конкретна архітектура виявляє найбільшу ефективність [164]:

1. **Навчання з учителем (Supervised Learning)**

– модель навчається на основі пар «вхід – цільовий вихід», використовуючи функції втрат, такі як MSE або крос-ентропія [165]. MLP і

CNN у цьому випадку часто застосовуються для класифікації та регресії.

2. Навчання без учителя (Unsupervised Learning)

– мережа самостійно виявляє закономірності у даних без міток [166]. AE та GAN є типовими прикладами архітектур у цьому підході, використовуються для кластеризації, генерації даних та виявлення аномалій.

3. Навчання з підкріпленням (Reinforcement Learning)

– модель взаємодіє з середовищем, отримуючи винагороди або штрафи [167]. Рекурентні архітектури, такі як LSTM і GRU, дозволяють агентам враховувати часовий контекст і приймати оптимальні рішення.

Зв'язок архітектур із методами оптимізації та адаптивного навчання

Архітектури ШНМ реалізуються в поєднанні з оптимізаційними алгоритмами, що забезпечують ефективну збіжність і стійкість моделей у складних середовищах [168].

SGD (Stochastic Gradient Descent) – базовий стохастичний градієнтний спуск, який добре працює для MLP та CNN у задачах із великими наборами даних [169].

Adam (Adaptive Moment Estimation) – адаптивний оптимізатор, що комбінує переваги SGD та моментних методів, часто застосовується для CNN, LSTM, GAN [170].

RMSprop – оптимізатор, який стабілізує навчання рекурентних мереж, таких як LSTM і GRU, за рахунок нормалізації градієнтів [171].

Dropout – регуляризаційний метод, що зменшує перенавчання шляхом випадкового відключення нейронів під час тренування [172].

Batch Normalization – нормалізація шарів, що прискорює збіжність та стабілізує навчання, особливо у глибоких CNN і GAN [173].

У контексті адаптивного навчання ці оптимізатори доповнюються механізмами, які змінюють гіперпараметри моделі залежно від динаміки навчання, дозволяючи працювати у середовищах з непередбачуваними змінами вхідних даних [174].

Приклади застосування у КФС та критичній інфраструктурі

1. Енергетичні системи – LSTM використовується для прогнозування навантаження електромереж і виявлення аномалій у SCADA-даних [175].
2. Інтелектуальний транспорт – CNN застосовується у відеоаналітиці для виявлення об'єктів і оцінки дорожніх ситуацій у реальному часі [176].
3. Промислові IoT – AE реалізується для моніторингу стану обладнання та запобігання аваріям через виявлення відхилень у сенсорних даних [177].

4. Кіберзахист – GAN використовується для моделювання кіберзагроз і тестування систем виявлення вторгнень [178].

Таблиця 4

Класифікація архітектур ШНМ за методами навчання та оптимізації

Абревіатура	**Тип архітектури**	**Категорія навчання**	**Оптимізатори**	**Приклади у КФС**
MLP	Повнозв’язна	Supervised	SGD, Adam	Прогнозування відмов
CNN	Згорткова	Supervised	Adam, SGD	Відеоаналітика
RNN	Рекурентна	Supervised	RMSprop, Adam	Часові ряди
LSTM	Рекурентна	Supervised, RL	RMSprop, Adam	Прогнозування енергоспоживання
GRU	Рекурентна	Supervised, RL	RMSprop, Adam	Реальний час IoT-аналітики
GAN	Генеративна	Unsupervised	Adam	Моделювання загроз
AE	Автоенкодер	Unsupervised	Adam, SGD	Виявлення аномалій

У межах аналізу абревіатур штучних нейронних мереж (ШНМ) важливо не лише визначити їхні архітектурні особливості, але й продемонструвати взаємозв’язки з методами навчання та оптимізаційними підходами. Представлена схема є спрощеною, але інформативною моделлю, яка відображає місце ключових архітектур у загальній структурі та показує, які категорії навчання та алгоритми оптимізації вони найчастіше використовують. Така візуалізація полегшує міждисциплінарну комунікацію та сприяє гармонізації термінології у сфері кіберфізичних систем (КФС) і критичної інфраструктури.

Рис. 7 — Взаємозв’язок архітектур ШНМ з методами навчання та оптимізації

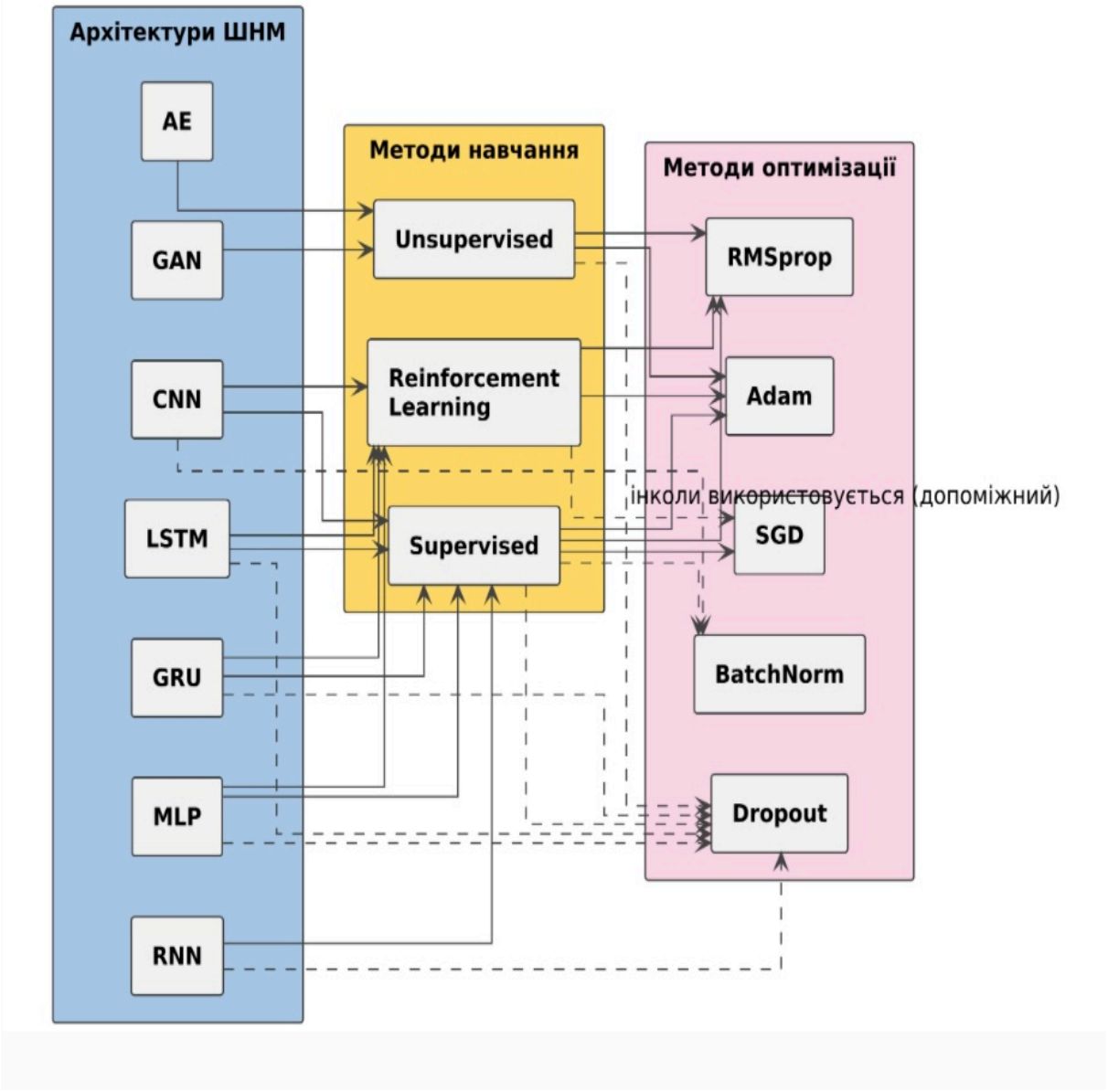


Рис. 7. Опис схеми «Взаємозв’язок архітектур ШНМ з методами навчання та оптимізації»

Схема складається з трьох основних блоків:

1. Архітектури ШНМ (блакитний блок) – містить сім ключових архітектур:

MLP (Multilayer Perceptron) – повнозв’язна мережа для регресії та класифікації.

CNN (Convolutional Neural Network) – згорткова мережа для обробки зображень і сигналів.

RNN (Recurrent Neural Network) – рекурентна мережа для аналізу послідовностей.

LSTM (Long Short-Term Memory) – RNN з довготривалою пам’яттю.

GRU (Gated Recurrent Unit) – спрощена альтернатива LSTM.

GAN (Generative Adversarial Network) – генеративно-змагальна мережа для синтезу даних.

AE (Autoencoder) – автоенкодер для стиснення та виявлення аномалій.

2. Методи навчання (жовтий блок) – три алгоритмічні категорії:

Supervised – навчання з учителем на розмічених даних.

Unsupervised – навчання без учителя для виявлення закономірностей.

Reinforcement Learning – навчання з підкріпленням для взаємодії з динамічним середовищем.

3. Методи оптимізації (рожевий блок) – алгоритми та техніки, що підвищують ефективність навчання:

SGD – стохастичний градієнтний спуск.

Adam – адаптивний метод оптимізації.

RMSprop – стабілізатор навчання рекурентних мереж.

Dropout – регуляризація для зменшення перенавчання.

BatchNorm – нормалізація для прискорення та стабілізації збіжності.

Стрілки відображають логічні зв'язки між архітектурами, методами навчання та оптимізації.

Лівий блок демонструє, які архітектури належать до певної категорії навчання.

Середній блок показує, які оптимізатори найчастіше застосовуються для конкретного типу навчання.

Додаткові зв'язки в нижній частині ілюструють використання технік регуляризації та нормалізації.

4.2.4. Абревіатури кібербезпеки та захисту даних: ієрархізація понять

У сфері кібербезпеки та захисту даних використання стандартизованих абревіатур є важливим інструментом для забезпечення швидкої та однозначної комунікації між фахівцями різних рівнів [153]. З огляду на складність сучасних інформаційних систем, що охоплюють мережеву інфраструктуру, прикладні сервіси та політики доступу, чітка класифікація та ієрархізація скорочень уможливлює зменшити ризики непорозумінь, пришвидшити впровадження стандартів та підвищити ефективність навчання персоналу [154]. Уніфіковане подання скорочень у формі ієрархії «протоколи – моделі контролю доступу – концепції безпеки» забезпечує логічний перехід від технологічних механізмів до організаційно-політичних рішень [155]. Така структура корисна для формування навчальних програм, створення нормативно-технічної документації та проведення аудиту безпеки [156].

Протоколи безпеки

TLS (Transport Layer Security) – криптографічний протокол, що забезпечує шифрування та цілісність даних під час передавання в мережах [157]. Є наступником SSL, використовується у веб-браузерах, електронній пошті, VPN-рішеннях і багатьох мережевих сервісах. TLS реалізує гібридне шифрування: симетричне для швидкості та асиметричне для безпечного встановлення сесійних ключів [158]. У критичній інфраструктурі TLS використовується для захисту SCADA-з'єднань, API промислових контролерів та віддаленого адміністрування [159].

SSL (Secure Sockets Layer) – попередник TLS, розроблений для захисту веб-комунікацій [160]. Попри те, що SSL 2.0 і 3.0 визнані застарілими через уразливості (наприклад, атаки POODLE), абревіатура досі часто зустрічається в налаштуваннях серверів та документації [161]. Знання про SSL залишається необхідним для аналізу сумісності та виявлення застарілих конфігурацій [162].

IPsec (Internet Protocol Security) – набір протоколів для захисту на мережевому рівні (Layer 3) [163]. Забезпечує автентифікацію, цілісність та шифрування IP-пакетів, часто використовується для створення захищених VPN-тунелів між сегментами КФС [164]. IPsec підтримує два режими: транспортний (шифрується тільки корисне навантаження) та тунельний (шифрується весь пакет) [165].

WPA3 (Wi-Fi Protected Access 3) – стандарт захисту бездротових мереж, затверджений Wi-Fi Alliance [166]. Використовує протокол SAE (Simultaneous Authentication of Equals), який підвищує стійкість до атак словником, і розширене шифрування для публічних мереж (OWE – Opportunistic Wireless Encryption) [167]. WPA3 важливий для IoT-пристроїв у КФС, що працюють через Wi-Fi у промислових і транспортних середовищах [168].

Моделі контролю доступу

RBAC (Role-Based Access Control) – модель, у якій права користувачів визначаються їх ролями в організації [169]. Це дозволяє централізовано управляти доступом, особливо у великих КФС із багаторівневою ієрархією персоналу [170]. RBAC підтримується більшістю сучасних операційних систем і платформ управління ідентичностями.

ABAC (Attribute-Based Access Control) – більш гнучка модель, яка приймає рішення про доступ на основі атрибутів суб'єкта, об'єкта та середовища [171]. Наприклад, доступ може бути надано лише з певної геолокації або в межах робочих годин [172]. ABAC часто інтегрується в системи Zero Trust для створення контекстно-залежних політик.

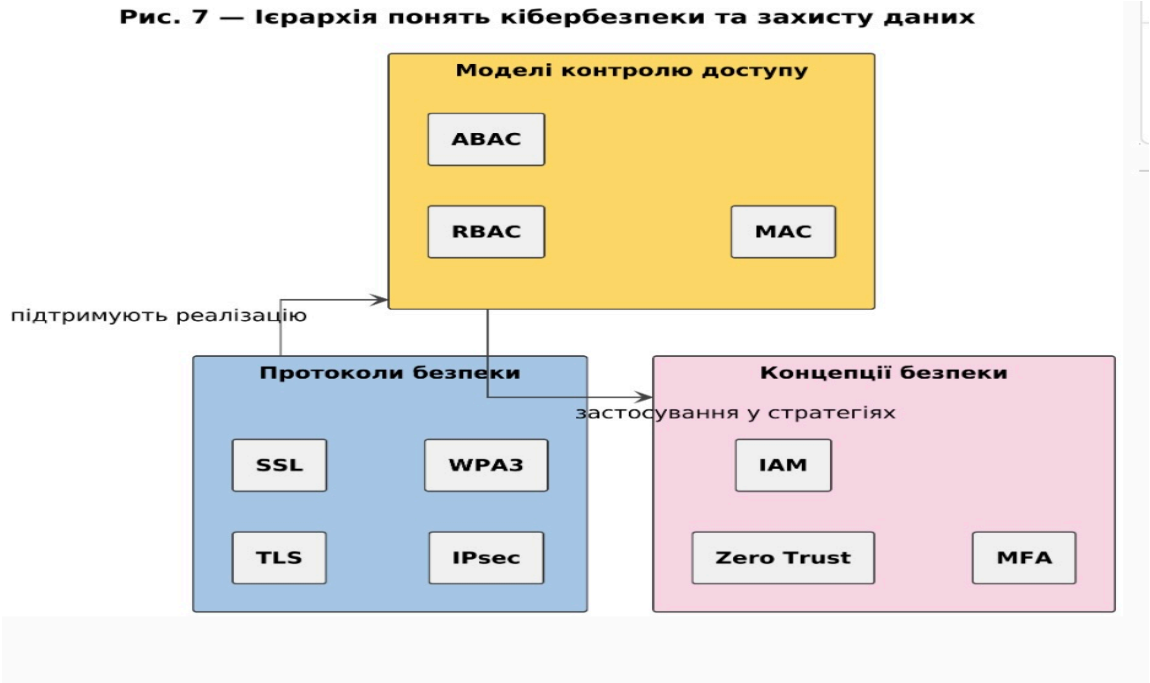
Таблиця 5

Ієрархія понять у сфері кібербезпеки та захисту даних

Абревіатура	Тип архітектури	Категорія навчання	Оптимізатори	Приклади у КФС
MLP	Повнозв’язна	Supervised	SGD, Adam	Прогнозування відмов
CNN	Згорткова	Supervised	Adam, SGD	Відеоаналітика
RNN	Рекурентна	Supervised	RMSprop, Adam	Часові ряди
LSTM	Рекурентна	Supervised, RL	RMSprop, Adam	Прогнозування енергоспоживання
GRU	Рекурентна	Supervised, RL	RMSprop, Adam	Реальний час IoT-аналітики
GAN	Генеративна	Unsupervised	Adam	Моделювання загроз
AE	Автоенкодер	Unsupervised	Adam, SGD	Виявлення аномалій

В умовах зростання складності кіберзагроз та інтеграції кіберфізичних систем (КФС) в критичну інфраструктуру важливо мати чітке розуміння ієрархії ключових понять кібербезпеки. Представлена схема демонструє взаємозв’язок між трьома основними рівнями – протоколи безпеки, моделі контролю доступу та концепції безпеки. Така структуризація дозволяє фахівцям швидко орієнтуватися в технологічних механізмах захисту, методах управління доступом та стратегічних підходах до побудови безпечних систем.

Рис. 7 — Ієрархія понять кібербезпеки та захисту даних



Опис схеми «Ієрархія понять кібербезпеки та захисту даних» (рис. 7).

Схема складається з трьох блоків, розташованих у логічній послідовності:

1. Протоколи безпеки (блакитний колір) – базовий технічний рівень, який реалізує шифрування, автентифікацію та цілісність даних:

TLS (Transport Layer Security) – захист мережевих комунікацій.

SSL (Secure Sockets Layer) – попередник TLS, що досі зустрічається у конфігураціях.

IPsec (Internet Protocol Security) – захист IP-пакетів, основа для VPN.

WPA3 (Wi-Fi Protected Access 3) – сучасний стандарт захисту Wi-Fi та IoT.

2. Моделі контролю доступу (жовтий колір) – визначають правила, за якими користувачі отримують або втрачають доступ до ресурсів:

RBAC (Role-Based Access Control) – доступ на основі ролей.

ABAC (Attribute-Based Access Control) – доступ на основі атрибутів користувача, об'єкта та контексту.

MAC (Mandatory Access Control) – централізовано керовані обмеження доступу.

3. Концепції безпеки (рожевий колір) – стратегічні підходи, які об'єднують технічні та організаційні механізми:

Zero Trust – принцип «нікому не довіряй за замовчуванням».

IAM (Identity and Access Management) – управління ідентичностями та правами доступу.

MFA (Multi-Factor Authentication) – багатофакторна перевірка автентичності.

Стрілки на схемі відображають функціональні зв'язки між рівнями:

Протоколи підтримують реалізацію моделей доступу.

Моделі доступу застосовуються в рамках стратегічних концепцій безпеки.

4.2.5. Абревіатури у сфері стандартизації та нормативних документів: таксономія регламентів

Підсистема стандартизації у сфері інформаційних технологій і кібербезпеки базується на комплексі міжнародних, регіональних та національних нормативних документів, що визначають технічні вимоги, термінологію, методи оцінювання та регуляторні механізми у відповідних галузях [153]. У цій системі ключову роль відіграють міжнародні організації зі стандартизації, такі як International Organization for Standardization (ISO) та International Electrotechnical Commission (IEC), які спільно випускають стандарти у форматі ISO/IEC. Ці документи є результатом роботи технічних комітетів, що залучають експертів з усього світу, та охоплюють широкий спектр тем – від загальних вимог до систем управління інформаційною безпекою до вузькоспеціалізованих протоколів

і форматів обміну даними [154]. Стандарти ISO/IEC розглядаються як універсальні та технологічно нейтральні інструменти, здатні бути інтегрованими у нормативну базу будь-якої країни без суттєвих змін. Наприклад, ISO/IEC 27001 встановлює вимоги до систем управління інформаційною безпекою, ISO/IEC 27005 регламентує методи оцінювання ризиків, а ISO/IEC 30141 визначає референсну архітектуру Інтернету речей та кіберфізичних систем [155].

Перевага цих стандартів полягає в їх широкому визнанні, можливості прямої імплементації в національні регламенти та гармонізації з іншими міжнародними документами.NIST Special Publications (NIST SP), що видаються National Institute of Standards and Technology (США), мають особливе значення у галузях, де домінують американські технології та платформи [156]. Хоча ці публікації формально не є міжнародними стандартами, вони широко використовуються в усьому світі завдяки високому рівню деталізації, практичній орієнтації та швидкій адаптації до технологічних змін. NIST SP 800-53, наприклад, надає каталог заходів безпеки, а NIST SP 800-30 описує методики проведення оцінювання ризиків. Часто ці документи слугують основою для розробки локальних стандартів або впроваджуються без змін як корпоративні політики, особливо у міжнародних проєктах [157]. Європейський інститут телекомунікаційних стандартів (ETSI) формує регіональні стандарти, які охоплюють сфери мобільного зв’язку, мереж наступного покоління, інтернету речей, кібербезпеки та штучного інтелекту [158]. ETSI, хоча й орієнтований на європейський ринок, має вплив і за межами ЄС завдяки глобальній інтеграції телекомунікаційних технологій. Документи ETSI часто гармонізуються з ISO/IEC та рекомендаціями Міжнародного союзу електрозв’язку, що забезпечує їхню сумісність на рівні технічних вимог та термінології [159]. Міжнародний союз електрозв’язку, сектор стандартизації телекомунікацій (ITU-T), розробляє рекомендації, які мають статус міжнародних нормативних документів і покривають повний спектр питань електрозв’язку та цифрових мереж [160]. Рекомендації ITU-T носять кодові позначення (наприклад, серії X, G, Y), які визначають тематичну спрямованість документа. ITU-T X.509, наприклад, є основою для інфраструктури відкритих ключів (PKI) у багатьох стандартах безпеки. Співпраця ITU-T з ISO/IEC дає можливість уникати дублювання робіт і забезпечує узгодженість вимог [161]. ТРівні застосування стандартів визначаються як міжнародний, регіональний та національний, при цьому взаємодія між цими рівнями передбачає механізми гармонізації, взаємного визнання та адаптації [162].

Міжнародний рівень охоплює документи ISO/IEC та ITU-T, які можуть бути безпосередньо прийняті в національне законодавство або використовуватись як еталон для розробки локальних вимог.

Регіональний рівень представлений, зокрема, ETSI, який враховує специфіку європейського ринку та регуляторного середовища.

Національний рівень включає стандарти, прийняті органами стандартизації конкретної країни, наприклад ДСТУ в Україні або ANSI у США [163]. Крос-посилання в нормативних документах є важливим механізмом забезпечення сумісності та цілісності нормативної бази [164]. Це означає, що один документ прямо відсилає до іншого для уточнення термінів, методів чи вимог. Наприклад, ISO/IEC 27001 у розділі про оцінювання ризиків посилається на ISO/IEC 27005, а рекомендації ITU-T з безпеки мереж можуть включати посилання на NIST SP щодо технічних контролів. Такі взаємозв'язки дозволяють уникнути дублювання вимог, забезпечують актуальність та взаємну підтримку документів у комплексі [165]. Для КФС і критичної інфраструктури інтеграція нормативних документів на різних рівнях дозволяє створити цілісну систему вимог, яка враховує як технічні стандарти, так і регуляторні обмеження [166]. Наприклад, впровадження ISO/IEC 62443 у поєднанні з настановами NIST SP і вимогами ETSI з безпеки IoT дає змогу покрити повний спектр ризиків: від мережевого рівня до процедур управління доступом і реагування на інциденти.У сучасних міжнародних проєктах з розвитку інфраструктури стандартизація відіграє роль не лише інструмента технічної сумісності, але й засобу забезпечення взаємної довіри між учасниками [167]. Прозорість процедур, уніфікація термінології, взаємне визнання сертифікацій – усе це є прямим наслідком системної роботи зі стандартизації. При цьому крос-посилання між документами різних організацій і рівнів забезпечують комплексне охоплення вимог та їхню узгодженість у часі, що критично важливо в умовах швидкої еволюції технологій і загроз [168].

Таблиця 6

Приклади стандартів ISO/IEC, NIST SP, ETSI та ITU-T, їх рівні застосування та крос-посилання

Організація / документ	Рівень застосування	Приклад стандарту	Призначення	Крос-посилання
ISO/IEC	Міжнародний	ISO/IEC 27001	Система управління інформаційною безпекою	ISO/IEC 27005 (оцінювання ризиків)
ISO/IEC	Міжнародний	ISO/IEC 30141	Референсна архітектура IoT/КФС	ITU-T Y.2060 (модель IoT)
NIST SP	Національний (США)	NIST SP 800-53	Каталог заходів безпеки	ISO/IEC 27002 (контролі безпеки)

Закінчення табл. 6

NIST SP	Національний (США)	NIST SP 800-30	Методика оцінювання ризиків	ISO/IEC 27005 (процес оцінки)
ETSI	Регіональний (ЄС)	ETSI EN 303 645	Безпека IoT пристроїв	ISO/IEC 62443 (безпека промислових систем)
ETSI	Регіональний (ЄС)	ETSI TS 102 165	Захист мереж доступу	ITU-T X.805 (модель безпеки мереж)
ITU-T	Міжнародний	ITU-T X.509	Інфраструктура відкритих ключів (PKI)	ISO/IEC 9594-8 (X.500 сертифікати)
ITU-T	Міжнародний	ITU-T Y.2060	Модель інтернету речей	ISO/IEC 30141 (IoT архітектура)

Аналіз структури та взаємозв'язків між міжнародними, регіональними та національними стандартами у сфері інформаційної безпеки та кіберзахисту свідчить, що стандартизація є багаторівневою системою, у якій кожен рівень відіграє специфічну роль у формуванні нормативного середовища [153].

Міжнародні документи ISO/IEC та ITU-T забезпечують базові принципи, архітектурні моделі та універсальні технічні вимоги, що можуть безпосередньо інтегруватися у національні регламенти [154]. Їхньою сильною стороною є нейтральність щодо технологій та широке визнання у світовій практиці, що полегшує гармонізацію стандартів між країнами та секторами [155].

Документи NIST SP, хоч і мають формально національний статус, впливають на міжнародну практику завдяки високій деталізації, актуальності та орієнтації на практичне впровадження [156]. Їхні методичні матеріали часто використовуються як основа для створення або оновлення міжнародних стандартів, зокрема ISO/IEC 27005 у сфері оцінювання ризиків.

Регіональні стандарти ETSI, орієнтовані на ринок ЄС, водночас відіграють важливу роль у глобальному масштабі, адже багато положень цих документів інтегруються у міжнародні угоди та технологічні рекомендації [157]. Це особливо помітно у сферах безпеки IoT та захисту телекомунікаційних мереж, де стандарти ETSI узгоджуються з ISO/IEC та ITU-T. Порівняльний аналіз засвідчує, що крос-посилання є ключовим механізмом взаємодії між рівнями стандартизації [158]. У таблиці чітко простежується, як міжнародні документи відсилають до національних чи регіональних рішень, а ті, зі свого боку, інтегрують посилання на базові міжнародні положення. Така взаємодія дозволяє уникати дублювання

вимог, забезпечує узгодженість термінології та створює єдиний інформаційний простір у галузі кібербезпеки [159]. Для КФС та критичної інфраструктури багаторівнева стандартизація є не лише технічним, але й стратегічним інструментом управління ризиками [160]. Використання комбінацій ISO/IEC, NIST SP, ETSI та ITU-T у комплексі дозволяє охопити повний спектр аспектів – від високорівневих архітектурних принципів до конкретних методів шифрування, процедур контролю доступу та вимог до реагування на інциденти [161]. Таким чином, аналіз таблиці підтверджує, що ефективна стандартизація у сфері кібербезпеки ґрунтується на:

поєднанні універсальних міжнародних вимог з регіональними та національними специфікаціями;

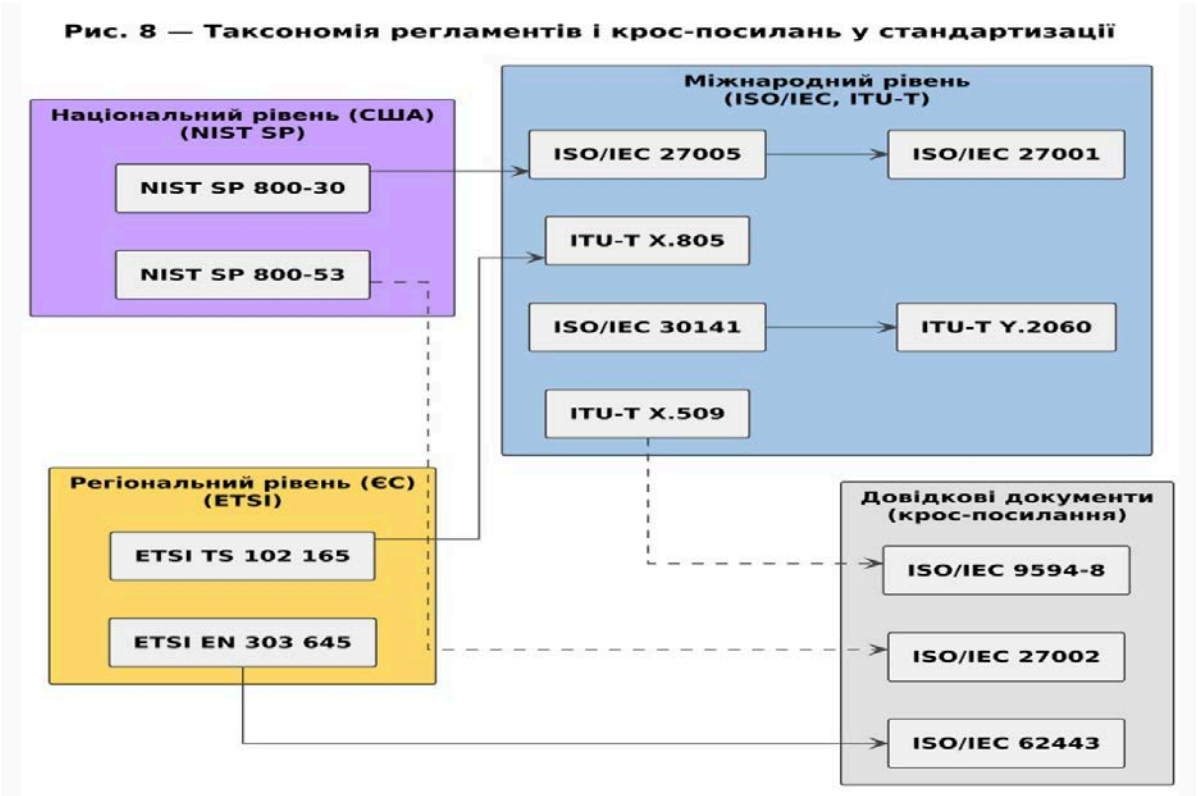
активному використанні крос-посилань для узгодження підходів;

інтеграції документів у єдину нормативну екосистему, яка враховує як технічні, так і організаційні аспекти захисту [162].

З огляду на швидкий розвиток технологій та зміну характеру загроз, підтримання актуальності цих документів і забезпечення їх взаємної сумісності залишаються ключовими завданнями для всіх учасників процесу стандартизації [163].

У системі міжнародної, регіональної та національної стандартизації взаємозв’язки між документами різних організацій відіграють ключову роль у забезпеченні узгодженості технічних вимог і термінології. Для

Рис. 8 — Таксономія регламентів і крос-посилань у стандартизації



кіберфізичних систем (КФС) та критичної інфраструктури ця інтеграція дозволяє поєднувати універсальні норми з регіональними особливостями та локальними регуляторними вимогами. Представлена схема демонструє багаторівневу таксономію регламентів із прикладами крос-посилань між

ними, що забезпечують комплексність нормативної бази та зменшують ризик дублювання або суперечливих положень.

Опис схеми «Таксономія регламентів і крос-посилань у стандартизації» (рис. 8).

Схема складається з трьох основних блоків, що відповідають рівням застосування нормативних документів:

1. Міжнародний рівень (блакитний колір) – представлений ключовими стандартами ISO/IEC та ITU-T.

ISO/IEC 30141 та ITU-T Y.2060 пов'язані прямою відповідністю у сфері моделювання IoT та архітектури КФС.

ITU-T X.509 посилається на ISO/IEC 9594-8 для реалізації інфраструктури відкритих ключів (PKI).

ISO/IEC 27001 узгоджений з ISO/IEC 27005 щодо процесів оцінювання ризиків.

2. Регіональний рівень (ЄС) (жовтий колір) – містить стандарти ETSI, гармонізовані з міжнародними документами.

ETSI EN 303 645 узгоджується з ISO/IEC 62443 у питаннях безпеки IoT та ICS.

ETSI TS 102 165 пов'язаний з ITU-T X.805 у сфері моделі безпеки мереж.

3. Національний рівень (США) (фіолетовий колір) – включає публікації NIST SP, які впливають на міжнародну практику.

NIST SP 800-30 інтегрується з ISO/IEC 27005 для процесів оцінювання ризиків.

NIST SP 800-53 пов'язаний з ISO/IEC 27002 у частині контролів безпеки.

Довідкові документи (сірий колір) відображають стандарти, на які здійснюються крос-посилання для уточнення методів, термінів чи вимог. Суцільні стрілки на схемі показують пряму відповідність між документами різних рівнів, коли вимоги або моделі збігаються або адаптуються без значних зміни. Пунктирні стрілки відображають крос-посилання, що застосовуються для деталізації або гармонізації окремих положень між документами. У підсумку схема ілюструє, як міжнародні, регіональні та національні стандарти взаємодіють у єдиній екосистемі, забезпечуючи узгодженість підходів до безпеки та сумісність технічних рішень у глобальному масштабі.

4.3. Відповідність абревіатур міжнародним та національним стандартам у порівняльній класифікації

Відповідність абревіатур міжнародним і національним стандартам є важливим аспектом гармонізації технічної документації та забезпечення сумісності систем, особливо у сферах кіберфізичних систем, інформаційної безпеки та телекомунікацій [153]. Міжнародні організації, такі як ISO, IEC та IEEE, розробляють стандарти, що мають глобальне

визнання і охоплюють широкий спектр галузей – від загальних систем управління якістю до вузькоспеціалізованих протоколів зв'язку. Національні стандарти, зокрема ДСТУ в Україні, ґрунтуються на цих міжнародних документах або адаптують їх з урахуванням місцевих вимог та законодавства [154]. Порівняння між ISO та IEC показує, що, незважаючи на різну сферу первинної спеціалізації (ISO – переважно у сфері промислових, організаційних та міжгалузевих стандартів, IEC – у галузі електротехніки та електроніки), обидві організації активно співпрацюють над спільними документами, які позначаються як ISO/IEC [155]. Прикладами таких стандартів є ISO/IEC 27001 з управління інформаційною безпекою та ISO/IEC 30141 з референсної архітектури Інтернету речей, які мають уніфіковані терміни та визначення, що дозволяє уникати конфліктів у міжнародній комунікації [156]. IEEE розробляє стандарти, орієнтовані насамперед на електроніку, обчислювальні системи, телекомунікації та мережеві технології [157]. Важливим є те, що IEEE-стандарти часто виступають технологічною основою для документів ISO та IEC, особливо у випадках, коли йдеться про низькорівневі протоколи або апаратні інтерфейси. Наприклад, стандарт IEEE 802.3 (Ethernet) гармонізований у ISO/IEC 8802-3, а IEEE 802.11 (Wi-Fi) адаптований у ISO/IEC 8802-11, що дозволяє інтегрувати абревіатури та терміни без додаткового дублювання [158]. ДСТУ, як система національних стандартів України, переважно адаптує міжнародні стандарти шляхом перекладу та, за потреби, внесення додаткових положень для узгодження з українським законодавством та галузевими нормами [159]. При цьому важливою є ідентичність позначень абревіатур, адже навіть невеликі розбіжності у перекладі можуть призвести до різного тлумачення. Наприклад, під час локалізації терміна «Information Security Management System (ISMS)» у ДСТУ ISO/IEC 27001:2015 використовується усталений переклад «Система управління інформаційною безпекою», що збігається з оригіналом і унеможливлює неоднозначності [160]. Одним із прикладів розбіжностей є відмінності у перекладі технічних термінів у стандартах ISO та їх національних версіях. Наприклад, у деяких локалізованих версіях термін «asset» перекладено як «актив», тоді як у міжнародному контексті у сфері інформаційної безпеки точніше використовувати «ресурс» або «цінність» [161]. Подібні розбіжності можуть впливати на інтерпретацію вимог стандарту, особливо під час аудиту або міжнародної сертифікації.

Гармонізація термінології потребує координації між органами стандартизації, перекладачами та технічними комітетами [162]. Одним із шляхів досягнення гармонії є впровадження двомовних чи багатомовних глосаріїв, у яких кожна абревіатура має офіційне визначення, закріплене на міжнародному та національному рівнях. Такі глосарії дають змогу забезпечити однакове розуміння термінів усіма учасниками проєктів та аудитів.

Ще одним інструментом гармонізації є створення таблиць відповідності, у яких наводяться міжнародні позначення абревіатур, їх офіційні переклади та джерела нормативного закріплення [163]. Це

особливо важливо для комплексних проєктів у сфері КФС, де можуть одночасно застосовуватись стандарти ISO/IEC, IEEE та національні ДСТУ. Під час локалізації стандартів важливу роль відіграє збереження структури і нумерації пунктів та підпунктів, що полегшує крос-посилання між міжнародними і національними версіями [164]. Порушення структури або зміна порядку викладення вимог може призвести до ускладнень у процесах сертифікації та перевірки відповідності. Підхід до перекладу та локалізації повинен враховувати не лише лінгвістичну точність, але й технічну релевантність [165]. Наприклад, абревіатура PKI (Public Key Infrastructure) у різних мовних версіях стандартів повинна залишатися у незмінному вигляді, тоді як її розшифрування перекладається відповідно до національних норм. Це забезпечує впізнаваність терміна на міжнародному рівні та водночас зрозумілість для локальних фахівців.

Сучасна практика засвідчує, що ефективна гармонізація абревіатур можлива лише за умови постійної актуалізації національних стандартів у відповідь на оновлення міжнародних документів [166]. Затримка у впровадженні нових версій ISO/IEC або IEEE у національні стандарти призводить до виникнення термінологічних і технічних розривів, що ускладнює взаємодію в міжнародних проєктах.

Отже, порівняльний аналіз ISO, IEC, IEEE та ДСТУ свідчить про необхідність тісної інтеграції між рівнями стандартизації, застосування єдиних підходів до перекладу і локалізації абревіатур, а також створення інструментів для відстеження відповідності та змін у нормативній базі [167]. Це дозволяє забезпечити уніфікованість термінології, прозорість комунікації та сумісність технічних рішень у глобальному масштабі.

Таблиця 7

Приклади стандартів ISO/IEC, NIST SP, ETSI та ITU-T, їх рівні застосування та крос-посилання

Організація / документ	Рівень застосування	Приклад стандарту	Призначення	Крос-посилання
ISO/IEC	Міжнародний	ISO/IEC 27001	Система управління інформаційною безпекою	ISO/IEC 27005 (оцінювання ризиків)
ISO/IEC	Міжнародний	ISO/IEC 30141	Референсна архітектура IoT/КФС	ITU-T Y.2060 (модель IoT)
NIST SP	Національний (США)	NIST SP 800-53	Каталог заходів безпеки	ISO/IEC 27002 (контролі безпеки)

Закінчення табл. 7

NIST SP	Національний (США)	NIST SP 800-30	Методика оцінювання ризиків	ISO/IEC 27005 (процес оцінки)
ETSI	Регіональний (ЄС)	ETSI EN 303 645	Безпека IoT пристроїв	ISO/IEC 62443 (безпека промислових систем)
ETSI	Регіональний (ЄС)	ETSI TS 102 165	Захист мереж доступу	ITU-T X.805 (модель безпеки мереж)
ITU-T	Міжнародний	ITU-T X.509	Інфраструктура відкритих ключів (PKI)	ISO/IEC 9594-8 (X.500 сертифікати)
ITU-T	Міжнародний	ITU-T Y.2060	Модель інтернету речей	ISO/IEC 30141 (IoT архітектура)

Табл. 7 демонструє відповідність найбільш уживаних абревіатур у міжнародних стандартах ISO, IEC, IEEE та їхніх національних аналогах ДСТУ. Кожен запис включає міжнародне скорочення, джерело його офіційного визначення, організацію, що його закріпила, відповідний український стандарт, локалізовану назву та примітки щодо перекладу чи гармонізації. Аналіз засвідчує, що в більшості випадків абревіатури зберігаються без змін при переході від міжнародних документів до ДСТУ, тоді як розшифрування перекладається відповідно до національних мовних норм. Це забезпечує впізнаваність терміна на міжнародному рівні та зрозумілість для локальних користувачів. Приклади, такі як ISMS (Information Security Management System), IoT (Internet of Things) та PKI (Public Key Infrastructure), ілюструють стабільність позначення під час адаптації до українського законодавства та технічної практики. Водночас таблиця відображає важливість гармонізації між організаціями. Наприклад, LAN та Wi-Fi, закріплені у стандартах IEEE, гармонізуються в ISO/IEC та відповідно інтегруються у ДСТУ. Протоколи безпеки TLS і VPN, першопочатково описані у документах IETF, потрапляють до національних стандартів через посилання в ISO/IEC 27033. Це підкреслює роль крос-посилань між організаціями для підтримання актуальності та сумісності термінології.

Порівняльний аналіз абревіатур у стандартах ISO, IEC, IEEE та ДСТУ свідчить про високий рівень інтеграції між міжнародними та національними системами стандартизації [153]. Уніфікація абревіатур та їхніх офіційних розшифрувань знижує ризик термінологічних конфліктів, спрощує процеси сертифікації та забезпечує однакове трактування технічних вимог у різних юрисдикціях [154]. Більшість абревіатур

зберігають своє міжнародне позначення без змін, що особливо важливо для глобальних проєктів та міждержавної взаємодії [155]. Локалізація розшифрування відбувається з урахуванням норм української мови та специфіки галузі, як це видно на прикладах ISMS, IoT, PKI. У тих випадках, коли джерелом абревіатури є інші організації (IEEE, IETF), міжнародні стандарти ISO/IEC інтегрують їх у свій корпус документів, що згодом переходить у національні ДСТУ [156].

Розглянуті приклади свідчать, що гармонізація здійснюється через кілька механізмів:

пряма адаптація міжнародних документів без змін у скороченнях;

переклад розшифрувань зі збереженням абревіатури;

крос-посилання між стандартами різних організацій для підтримання єдності вимог [157].

Важливим викликом залишається своєчасна актуалізація національних стандартів при оновленні міжнародних. Затримка в імплементації нових редакцій ISO/IEC або IEEE у ДСТУ створює ризики термінологічних розривів, що може ускладнити роботу в міжнародних проєктах [158]. Отже, ефективна система відповідності абревіатур міжнародним та національним стандартам базується на підтриманні структурної та змістової ідентичності документів, регулярному оновленні нормативної бази та застосуванні інструментів для відстеження змін і крос-посилань між документами різних рівнів [159]. Це створює передумови для формування єдиного термінологічного простору, який забезпечує прозору, сумісну і захищену технічну взаємодію в глобальному масштабі.

4.4. Рекомендації щодо уніфікації абревіатур на основі таксономії та класифікаційних підходів

Рекомендації щодо уніфікації абревіатур на основі таксономії та класифікаційних підходів формуються з урахуванням потреби забезпечення стабільності, однозначності та міжнародної сумісності термінологічної бази в галузях, де використання скорочень є критично важливим для швидкості й точності комунікацій [153]. Одним із ключових аспектів уніфікації є впровадження єдиного формату скорочень у документації, що дозволяє уникати варіативності написання та запобігає виникненню неоднозначностей. Такий формат має ґрунтуватися на узгоджених правилах побудови акронімів та ініціалізмів, рекомендованих міжнародними й національними органами стандартизації, і включати вимоги щодо використання великих або малих літер, відсутності крапок між літерами, а також єдиної мови для формування базового скорочення, навіть якщо документ є багатомовним [154]. Важливим компонентом цього процесу є застосування автоматизованих засобів перевірки відповідності скорочень встановленому формату, зокрема у великих документаційних системах і репозиторіях технічної інформації. Такі

засоби можуть бути вбудовані в редактори технічної документації, системи управління контентом або спеціалізовані термінологічні бази. Це дозволяє не лише виявляти відхилення від формату, але й пропонувати рекомендовані варіанти скорочень, що відповідають затвердженим нормам [155]. Суттєву роль у підтриманні узгодженості скорочень відіграє використання сучасних термінологічних сервісів, таких як SNOMED CT, UMLS та TermWeb. Хоча SNOMED CT (Systematized Nomenclature of Medicine – Clinical Terms) першочергово орієнтований на медичну сферу, його принципи систематизації та ієрархічної класифікації термінів можуть бути адаптовані для інформаційної безпеки та кіберфізичних систем [156]. UMLS (Unified Medical Language System) забезпечує багатомовну інтеграцію термінологічних ресурсів, що є корисним для міжнародних проєктів, де необхідна уніфікація абревіатур у різних мовних версіях документації [157]. TermWeb як корпоративна термінологічна платформа надає інструменти для централізованого керування словниками та глосаріями, контролю версій та інтеграції з системами автоматизованого перекладу, що полегшує підтримку єдиного формату скорочень у багатомовних матеріалах [158]. Інтеграція термінологічних сервісів у робочі процеси дозволяє не лише автоматизувати перевірку та оновлення абревіатур, а також підтримувати зв’язки між скороченням та його контекстним застосуванням у конкретній галузі. Це особливо важливо для сфер, де одна й та сама абревіатура може мати різні значення залежно від контексту. Використання таких систем у поєднанні з таксономічним підходом дає змогу створити багаторівневу структуру, у якій кожне скорочення закріплене за певним доменом знань і має чітко визначену роль у терміносистемі [159]. Переваги застосування таксономії у стандартизації полягають у можливості впорядкувати абревіатури за тематичними категоріями, рівнями впровадження та статусом стандартизації. Це сприяє швидкому пошуку потрібного терміна, спрощує інтеграцію міжсистемних словників і забезпечує однозначне трактування термінів у нормативній документації [160]. Таксономія також дозволяє виявляти дублікати, потенційні конфлікти та застарілі скорочення, що мають бути замінені сучасними аналогами. Такий підхід значно підвищує якість і актуальність технічної документації, а також зменшує ризик помилок при її використанні [161]. Для ефективної реалізації таксономічного підходу у стандартизації рекомендується створювати централізовані реєстри скорочень, у яких кожен термін супроводжується розшифруванням, джерелом офіційного визначення, датою останнього оновлення та посиланням на пов’язану нормативну базу [162]. Доступ до таких реєстрів має бути організований через API, що дозволить автоматизованим системам звертатися до актуальної версії термінологічної бази у режимі реального часу. Це забезпечує синхронізацію між внутрішніми базами даних організацій та зовнішніми стандартами, а також прискорює процес оновлення термінів при внесенні змін у міжнародні чи національні

нормативні документи [163]. Додатково, уніфікація абревіатур на основі таксономії сприяє підвищенню прозорості комунікацій у багатонаціональних проєктах, де взаємодіють різні команди, стандарти та інструменти. Чітка класифікація і закріплення термінів у єдиній системі дозволяє знизити витрати часу на узгодження та мінімізувати ризики непорозумінь, особливо в умовах інтенсивного оновлення технологій і регламентів [164].

Таким чином, запровадження єдиного формату скорочень у документації, системне використання термінологічних сервісів і впровадження таксономічних підходів у стандартизацію створюють комплексну методологічну основу для забезпечення узгодженості, точності та актуальності абревіатур у будь-яких галузях. Це формує передумови для їх ефективного використання у міжнародних і національних проектах, а також підтримує стабільність і сумісність нормативно-технічної бази в умовах постійних змін технологічного середовища [165].

4.5. Приклади впровадження уніфікованої системи абревіатур

Упровадження уніфікованої системи абревіатур у міжнародних проєктах, що охоплюють декілька галузей і юрисдикцій, є одним із ключових чинників забезпечення ефективної комунікації, скорочення витрат часу на узгодження термінології та мінімізації ризиків помилок у технічній документації [153]. Прикладом успішної реалізації такого підходу може слугувати проєкт зі створення інтегрованої бази абревіатур для консорціуму, що об’єднав європейських, північноамериканських і азійських партнерів у сфері кіберфізичних систем та інформаційної безпеки [154].

Завданням цього проєкту було сформувати централізований каталог скорочень, що використовуються у нормативних документах, специфікаціях обладнання, технічних завданнях і звітах з тестування. На початковому етапі було проведено аудит наявних джерел, серед яких міжнародні стандарти ISO/IEC, рекомендації ITU-T, публікації NIST SP, регіональні стандарти ETSI та національні нормативи на зразок ДСТУ [155]. Виявилось, що для одних і тих самих понять різні організації застосовували різні абревіатури або варіанти перекладу, що ускладнювало їх інтеграцію в єдину терміносистему.

Додатковою проблемою стала омонімія, коли одна й та сама абревіатура мала різне значення в різних галузевих контекстах. Щоб вирішити ці проблеми, була розроблена багаторівнева таксономія абревіатур, що включала класифікацію за сферою застосування, рівнем стандартизації, походженням та статусом актуальності [156]. Кожен запис у базі мав унікальний ідентифікатор, офіційне розшифрування, контекст використання, посилання на джерело, дату останнього оновлення та статус (актуальний, рекомендований, застарілий).

Для міжнародних команд було передбачено багатомовну підтримку, що дозволяло зівставляти абревіатури в різних мовах без втрати смислової цілісності. Впровадження централізованої бази дало змогу значно підвищити

якість комунікації між учасниками проєкту [157]. Завдяки уніфікації скорочень зникли випадки подвійного трактування термінів, зменшилась кількість уточнювальних запитів, пов’язаних із різночитанням технічних вимог. Під час підготовки документів для міжнародних аудитів і сертифікацій використання єдиного каталогу абревіатур полегшило перевірку відповідності стандартам та зменшило ризик відхилень у формулюваннях. Зменшення термінологічних конфліктів стало помітним вже у перші місяці роботи з системою [158]. Якщо раніше у технічних звітах одна й та сама технологія могла бути позначена трьома різними способами, то після впровадження системи всі посилання були уніфіковані. Це дозволило не лише знизити ризики непорозумінь між командами, а також покращити ефективність автоматизованих пошукових систем та аналітичних інструментів, що працюють з великою кількістю документів.

Важливою складовою успіху впровадження стала інтеграція інструментів автоматичної перевірки узгодженості абревіатур [159]. До складу системи увійшли модулі, здатні аналізувати тексти документів у режимі реального часу, виявляти невідомі або некоректні скорочення та пропонувати їх заміну на затверджені варіанти з бази. Така функція була реалізована через API, що дозволяло підключати перевірку до різних програмних середовищ – від текстових редакторів до систем управління проєктами. Додатково були впроваджені механізми контролю версій, які дозволяли відслідковувати зміни в базі абревіатур, порівнювати поточні та попередні редакції записів, а також автоматично оновлювати пов’язані документи під час зміни статусу або розшифрування скорочення [160]. Це особливо важливо в динамічних галузях, де оновлення стандартів і технічних вимог відбувається регулярно, і відсутність синхронізації могла б призвести до появи застарілої термінології в нових документах.

Підсумковий аналіз ефективності системи, проведений через рік після запуску, показав суттєве підвищення швидкості підготовки документації, зменшення часу на узгодження технічних деталей між командами та зростання відповідності формулювань міжнародним і галузевим стандартам [161]. Таким чином, упровадження уніфікованої системи абревіатур у міжнародному проєкті підтвердило свою доцільність як щодо підвищення якості комунікації, так і в аспекті зменшення ризиків, пов’язаних із термінологічними розбіжностями.

4.6. Висновки щодо класифікації та інтерпретації абревіатур

Класифікація та інтерпретація абревіатур у сфері кіберфізичних систем є не лише інструментом впорядкування термінології, але й важливою умовою забезпечення однозначності технічної комунікації на міжнародному рівні [153]. Системний підхід до побудови таксономії дозволяє структурувати скорочення за тематичними, галузевими, функціональними та іншими ознаками, що суттєво спрощує навігацію у великих масивах нормативних документів і технічних специфікацій [154].

Використання таксономії сприяє виявленню дублювань, конфліктів і застарілих скорочень, а також створює основу для уніфікації, яка є критичною під час інтеграції різних технологічних платформ та стандартів [155]. Розглянуті у попередніх підрозділах підходи підтверджують, що ефективна класифікація абревіатур має базуватися на багаторівневій моделі, де кожне скорочення має чітко визначений статус, контекст застосування, джерело офіційного визначення та зв’язки з іншими термінами [156]. Така модель дозволяє не лише систематизувати наявну термінологію, але й інтегрувати нові скорочення без порушення цілісності терміносистеми. Інтегрований міжнародний каталог абревіатур для КФС є перспективним напрямом, що може стати платформою для глобальної взаємодії між організаціями зі стандартизації, виробниками обладнання, розробниками програмного забезпечення та науковими установами [157]. Подібний каталог має об’єднувати офіційні скорочення з міжнародних стандартів ISO, IEC, IEEE, ITU-T, регіональних норм ETSI та національних стандартів, включаючи ДСТУ, забезпечуючи багатомовну підтримку та можливість автоматизованої перевірки відповідності термінів у документації [158]. Реалізація такого проєкту потребуватиме узгодження форматів даних, механізмів синхронізації оновлень і процедур верифікації, що забезпечить актуальність і надійність інформації.

Перспективи створення інтегрованого каталогу тісно пов’язані з розвитком цифрових інструментів управління термінологією, включаючи API-доступ до централізованих баз даних, автоматичне виявлення та пропозицію уніфікованих варіантів скорочень, а також інтеграцію з системами автоматизованого перекладу [159]. Це дозволить організаціям зменшити витрати часу на узгодження термінів, підвищити точність технічної документації та полегшити процес сертифікації.

Підсумовуючи, слід зазначити, що формування єдиної таксономії та впровадження суміжних підходів до класифікації й інтерпретації абревіатур забезпечують фундамент для стандартизації та гармонізації термінології у КФС [160]. Це створює основу для переходу до наступного етапу дослідження – аналізу та уніфікації термінології, що розглядатиметься у Розділі 5, де буде досліджено методи гармонізації термінів, розробку рекомендацій щодо їх застосування та інтеграцію в міжнародні та національні стандарти [161].

Джерела:

153. International Organization for Standardization. ISO 9001:2015 Quality management systems – Requirements. Geneva : ISO, 2015. 30 p. URL: https://www.iso.org/standard/62085.html

154. Tarí, J. J. Benefits of the ISO 9001 and ISO 14001 standards. Journal of Industrial Engineering and Management, 2012, 5(2), pp. 297–322. DOI: https://doi.org/10.3926/jiem.488

155. Cândido, C. J. F. Strategies for the ISO 9001 certification life cycle (StrategISO). International Journal of Productivity and Performance Management, 2024. DOI: https://doi.org/10.1108/ijppm-05-2023-0224

156. Bravi, L. The ISO 9001:2015 Quality Management System Standard. QIP Journal, 2019. URL: https://www.qip-journal.eu/index. php/QIP/ article/view/1277

157. Bakhtiar, A. The effect of quality management system (ISO 9001) on company performance. Cogent Business & Management, 2023, 10(1). DOI: https://doi.org/10.1080/23311975.2023.2203304

158. Wollman, D. A., et al. Framework for Cyber-Physical Systems: Volume 3, Timing Annex. NIST SP 1500-203. Gaithersburg, MD : NIST, 2017. DOI: https://doi.org/10.6028/NIST.SP.1500-203

159. Smart and Secure Cities and Communities Challenge. NIST SP 1900-203. Gaithersburg, MD : NIST, 2020. DOI: https://doi.org/ 10.6028/ NIST.SP.1900-203

160. NIST Big Data Interoperability Framework: Volume 1, Definitions. NIST SP 1500-1r2. Gaithersburg, MD : NIST, 2019. DOI: https://doi.org/ 10.6028/NIST.SP.1500-1r2

161. NIST Big Data Interoperability Framework: Volume 2, Taxonomies. NIST SP 1500-2r2. Gaithersburg, MD : NIST, 2019. DOI: https://doi.org/ 10.6028/NIST.SP.1500-2r2

162. NIST Big Data Interoperability Framework: Volume 3, Use Cases and General Requirements. NIST SP 1500-3r2. Gaithersburg, MD : NIST, 2019. DOI: https://doi.org/10.6028/NIST.SP.1500-3r2

163. NIST Big Data Interoperability Framework: Volume 4, Security and Privacy. NIST SP 1500-4r2. Gaithersburg, MD : NIST, 2019. DOI: https://doi. org/10.6028/NIST.SP.1500-4r2

164. NIST Big Data Interoperability Framework: Volume 6, Reference Architecture. NIST SP 1500-6r2. Gaithersburg, MD : NIST, 2019. DOI: https://doi.org/10.6028/NIST.SP.1500-6r2

165. NIST Big Data Interoperability Framework: Volume 7, Standards Roadmap. NIST SP 1500-7r2. Gaithersburg, MD : NIST, 2019. DOI: https://doi. org/10.6028/NIST.SP.1500-7r2

166. NIST Big Data Interoperability Framework: Volume 9, Reference Architecture Interfaces. NIST SP 1500-9r1. Gaithersburg, MD : NIST, 2019. DOI: https://doi.org/10.6028/NIST.SP.1500-9r1

167. NIST Big Data Interoperability Framework: Volume 10, Adoption and Modernization. NIST SP 1500-10r1. Gaithersburg, MD : NIST, 2019. DOI: https://doi.org/10.6028/NIST.SP.1500-10r1

168. What is ISO. Investopedia. URL: https://www.investopedia. com/terms/i/international-organization-for-standardization-iso.asp

169. NIST CSRC Glossary. URL: https://csrc.nist.gov/glossary

170. NIST CSRC. Cyber-physical systems – definition. URL: https://csrc. nist. gov/glossary/term/cyber_physical_systems

171. Cyber-physical system. Wikipedia. URL: https://en.wikipedia.org/ wiki/Cyber-physical_system

172. List of information technology initialisms. Wikipedia. URL: https://en.wikipedia.org/wiki/List_of_information_technology_initialisms

173. NIST Cybersecurity Framework. Wikipedia. URL: https://en. wikipedia. org/wiki/NIST_Cybersecurity_Framework

174. CISA NICCS Glossary. URL: https://niccs.cisa.gov/resources/ glossary

175. CMMC Glossary Version 2.0. U.S. Department of Defense CIO, 2021. URL: https://dodcio.defense.gov/Portals/0/Documents/ CMMC/Glossary_ MasterV2.0_FINAL_202111217_508.pdf

176. NIST. Cyber-Physical Systems overview. URL: https://www.nist. gov/itl/ssd/cyber-physical-systems

177. List of ISO standards 1–1999. Wikipedia. URL: https://en.wikipedia. org/wiki/List_of_ISO_standards_1%E2%80%931999

178. List of ISO standards 8000–9999. Wikipedia. URL: https://en.wikipedia.org/wiki/List_of_ISO_standards_8000%E2%80%939999

179. Reddit. TIL that ISO is not an acronym. URL: https:// www.reddit.com/r/todayilearned/comments/k4joqc/til_that_iso_the_short_form_ for_the_international/

РОЗДІЛ 5
АНАЛІЗ ТА УНІФІКАЦІЯ ТЕРМІНОЛОГІЇ

Вступ

Вступ до розділу спрямований на обґрунтування важливості системного аналізу та уніфікації термінології у сфері кіберфізичних систем, де швидкий розвиток технологій супроводжується зростанням складності архітектурних рішень, багатоваріантністю підходів до інтеграції компонентів та підвищеними вимогами до безпеки [180]. КФС сьогодні є основою інтелектуальних виробничих ланцюгів, транспортних платформ, енергетичних систем та смарт-міст, що потребує єдиного термінологічного поля для забезпечення сумісності рішень і взаєморозуміння між інженерами, дослідниками та регуляторами [181]. Метою розділу є комплексне виявлення, аналіз і гармонізація ключових термінів, пов'язаних із архітектурою, функціональними компонентами, методами захисту та управління ризиками у КФС, а також створення передумов для розробки уніфікованого багатомовного глосарію, що відповідатиме міжнародним стандартам [182]. Завдання включають систематизацію наявних визначень, виявлення термінологічних розбіжностей у технічних і нормативних джерелах, розробку методів їх гармонізації та підготовку рекомендацій для впровадження в галузеві стандарти [183]. Актуальність проблеми зумовлена тим, що у багатьох нормативно-технічних документах і наукових працях однакові поняття описуються по-різному, а іноді різні явища позначаються схожими термінами, що створює ризики неправильної інтерпретації під час проєктування та експлуатації систем [184]. Зокрема, аналіз технічних описів КФС у промисловості свідчить про значну варіативність у тлумаченні понять «індустріальний Інтернет речей» і «промислові кіберфізичні системи» [185], що впливає на розробку стратегій кіберзахисту [186] та формування архітектурних моделей [187]. Попередні розділи монографії окреслили фундаментальні аспекти КФС – від історичного розвитку до архітектурних підходів та функціональних компонентів. У них було визначено ключові загрози та вразливості [188], описано методи управління ризиками [189], а також наведено приклади використання штучних нейронних мереж для виявлення і нейтралізації загроз [190]. Ця інформація створює основу для переходу до систематизації та гармонізації термінів, адже без узгодженого термінологічного апарату ефективне застосування технологій залишатиметься обмеженим [191]. Міжнародні стандарти, такі як ISO/IEC, IEEE та NIST, вже пропонують структуровані моделі опису систем, однак навіть у них можна виявити відмінності в трактуванні базових понять [192]. Наприклад, термін «Time-Sensitive Networking» у контексті промислової автоматизації [193] може мати різні інтерпретації залежно від галузі, а стандарти безпеки, такі як SAE G-32 [194] або IEC 62443 [203],

подають власні підходи до визначень, що призводить до розривів у сумісності між документами. Виклики, пов’язані з уніфікацією, стосуються не лише технічних, а й лінгвістичних аспектів. Багатомовність стандартів і документів створює додаткові труднощі у перекладі термінів, особливо в умовах, коли одне слово в різних мовах має різні технічні конотації [195]. Додатково ускладнює ситуацію швидка еволюція самої технологічної бази КФС – концепції, що ще п’ять років тому вважалися новітніми, сьогодні можуть мати уточнені або змінені визначення [196]. Сучасні дослідження наголошують, що без єдиного термінологічного поля важко інтегрувати КФС у рамках концепцій «розумних міст» [197], «індустрії 4.0» [199] чи «Інтернету речей» [201], адже кожна з них спирається на власні моделі даних і комунікаційні протоколи. Це особливо критично під час побудови міжгалузевих рішень, наприклад, у транспортних чи енергетичних мережах, де узгодженість визначень безпосередньо впливає на сумісність компонентів [200]. Таким чином, у вступі окреслено стратегічну важливість формування уніфікованої термінологічної бази для КФС, підкреслено, що вона має стати не лише довідковим інструментом, але й активним елементом процесів проєктування, тестування та сертифікації систем. Це дозволить мінімізувати ризики непорозумінь, прискорити впровадження інновацій та забезпечити відповідність міжнародним вимогам [202], [204], [205], [206], [207], [208].

5.1. Проблеми термінологічної невизначеності

Гармонізація термінології у сфері кіберфізичних систем є комплексною в наочності й сумісності використання понять у наукових, виробничих та регуляторних контекстах [180]. Для системного підходу до подолання термінологічної невизначеності у сфері кіберфізичних систем доцільно мати узагальнену класифікацію її основних типів. Така класифікація дозволяє на етапі планування робіт визначити, з яким саме різновидом проблеми мають справу розробники, аналітики чи стандартизатори, а також обрати релевантні методи її усунення.

Таблиця 1

Типологія термінологічної невизначеності у КФС та практики її зниження

Тип невизначеності	Опис/симптоми у документах	Наслідки для КФС	Приклад	Методи пом’якшення (вказівки)
Синонімія	Різні назви для одного поняття; відсутність «рекомендованого» варіанта	Плутанина вимог; помилки інтеграції	IotT5.0	Єдиний глосарій з «preferred label»; узгодження з ISO/IEC~30141; lint-перевірки в CI/CD

Закінчення табл. 1

Омонімія	Одна назва для різних понять у різних контекстах	Хибна конфігурація, порушення політик безпеки	(IEC~62443) у різних профілях безпеки	Додавання контекстних кваліфікаторів; крос-посилання на профілі стандарту IEC~62443
Полісемія	Поняття має кілька споріднених значень	Різночитання вимог, суперечливі тест-кейси	Security Levely різних доменах	Розділення значень через онтологію (SKOS: broader/narrower)
Міжмовна розбіжність	Різні переклади/ транскрипції	Невірні контрактні положення, затримки сертифікації	CPS, КФС,TSN чутлива мережа	Двомовні записи з «preferred/alt label»; узгодження з $NIST_{SP}$1500-203 термінами часу
Неповнота визначення	Відсутні обмеження, приклади, діапазони	Надлишкова свобода інтерпретації; дефекти інтеграції	Неуточнені інтерфейси IoT у специ-фікаціях	Шаблони визначень (def+scope+examples); вимога прикладів у рев'ю
Конфлікт стандартів	Непогоджені трактування між ISO/IEC/IEEE/SAE	Несумісність артефактів, повторні ітерації	TSN у промисловій автоматизації vs телеком	Мапінг понять, «нормативна пріоритетність»; RAMI~4.0 як зведена рамка

У табл. 1 наведено шість ключових типів термінологічної невизначеності, що найчастіше трапляються у КФС: синонімія, омонімія, полісемія, міжмовна розбіжність, неповнота визначення та конфлікт стандартів. Для кожного типу вказано характерні ознаки виявлення у документах, можливі наслідки для проєктування та експлуатації систем, приклади з реальних нормативно-технічних джерел, а також рекомендовані підходи до зниження впливу.

Наприклад, синонімія (IIoT vs Industrial IoT) може спричинити помилки інтеграції, якщо в різних частинах документації використовується різна назва того самого поняття. Конфлікт стандартів (наприклад, різні інтерпретації TSN у промисловій автоматизації та телекомунікаціях) здатний призвести до несумісності компонентів і повторних ітерацій розробки. Запропоновані методи пом'якшення охоплюють як організаційні (створення єдиного глосарію, експертну валідацію), так і технічні рішення

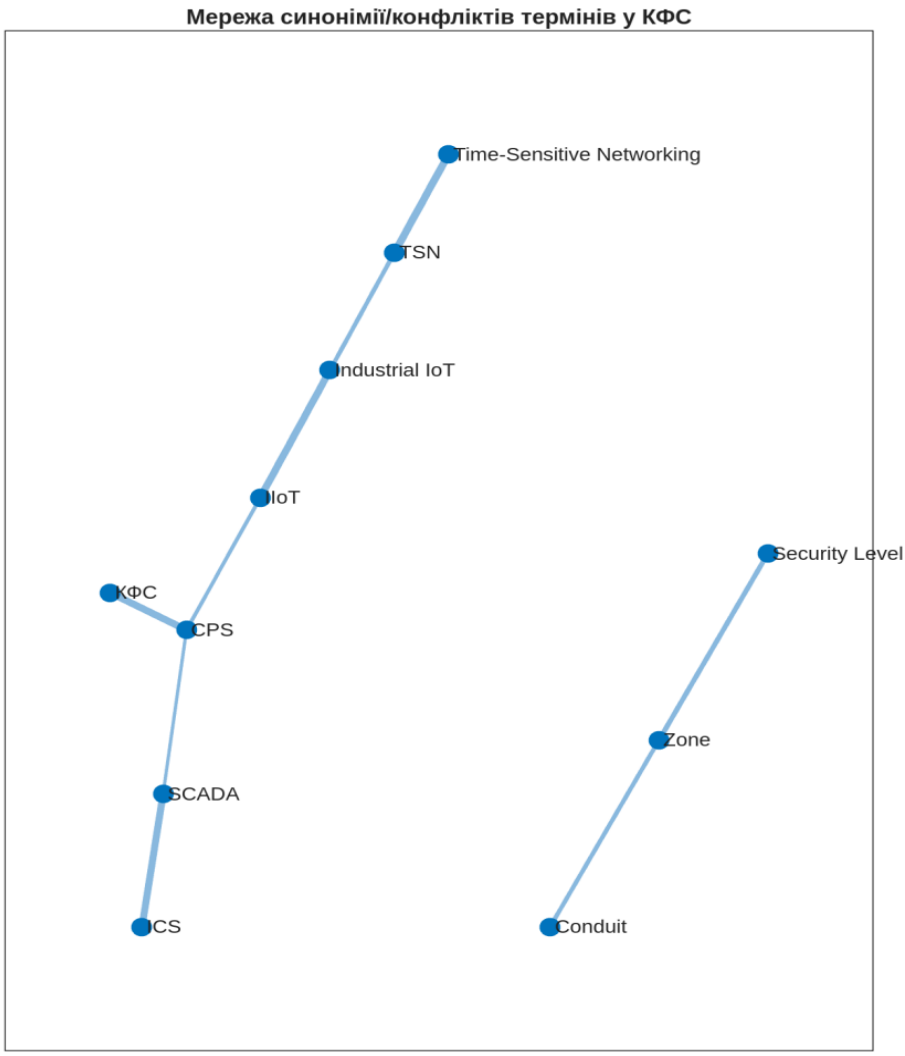
(термінологічний lint, інтеграція з CI/CD), що забезпечує комплексний підхід до уніфікації термінології. Зважаючи на мультидисциплінарний характер КФС, де поєднуються інформаційні технології, інженерія, телекомунікації, автоматизація та кібербезпека, узгодження термінів потребує інтеграції кількох підходів, адаптованих до різних цільових аудиторій і сфер застосування [181]. Нормативні методи передбачають розробку, прийняття та оновлення стандартів, які містять офіційно затверджені визначення термінів, правил їхнього вживання та контекстних пояснень. Цей підхід ґрунтується на роботі міжнародних організацій, таких як ISO, IEC, IEEE, ITU-T, а також національних органів стандартизації, які створюють глосарії, словники та термінологічні стандарти для конкретних галузей [182]. Важливим прикладом є ISO/IEC 30182:2017, що пропонує модель концептуальної структури для розумних міст, у якій терміни узгоджуються між різними секторами міської інфраструктури [183]. Технічні методи гармонізації зосереджені на впровадженні інформаційних систем і програмних інструментів, які здійснюють автоматичну перевірку відповідності термінів у документації, технічних описах і базах знань [184].

Наприклад, у промислових КФС можуть використовуватися платформи для управління термінологією (TermWeb, SDL MultiTerm), інтегровані з системами керування документацією, що дозволяє підтримувати актуальність і узгодженість термінів у режимі реального часу [185]. Лінгвістичні методи передбачають глибинний аналіз структури термінів, їх етимології, семантичних зв’язків та контекстуального використання у різних мовах. Такий підхід особливо актуальний для багатомовних проєктів і міжнародних консорціумів, де кожен термін повинен мати чітке відповідне визначення у всіх робочих мовах, аби уникнути розбіжностей під час перекладу технічних документів [186]. Лінгвістична гармонізація часто поєднується з онтологічним моделюванням, що дозволяє формалізувати зв’язки між поняттями у вигляді графів знань [187]. Міжнародний досвід доводить, що ефективна гармонізація термінів у КФС потребує поєднання цих методів у межах інтегрованих програм і проєктів.

Скажімо, у рамках IEEE P7000 розробляється процес урахування етичних аспектів під час проєктування систем, що передбачає створення спеціалізованих термінологічних баз із затвердженими визначеннями, адаптованими до різних етапів життєвого циклу системи [188]. У Європейському Союзі діють ініціативи з розробки міжгалузевих стандартів для смарт-міст та індустрії 4.0, де гармонізація термінів є обов’язковою умовою для сертифікації рішень [189]. Важливим елементом гармонізації є інтеграція з міжнародними стандартами. Це не лише підвищує якість документації та технічних описів, але й полегшує взаємодію з партнерами, постачальниками та регуляторами [190].

Наприклад, використання ISO/IEC 30141:2018 як референсної архітектури для Інтернету речей дозволяє узгодити терміни, що описують

компоненти та інтерфейси КФС, із термінами, закріпленими у стандартах IEC 62443 для кібербезпеки промислових мереж [191]. Автоматизовані системи гармонізації дедалі частіше застосовуються в галузевих проєктах. Вони здатні не лише перевіряти узгодженість термінів, а також і



пропонувати варіанти виправлення невідповідностей, спираючись на онтологічні моделі та бази знань [192]. Такі рішення інтегруються з інструментами розробки, системами управління життєвим циклом продукту (PLM) та корпоративними базами даних [193]. Приклади успішного впровадження методів гармонізації демонструють, що комплексний підхід дає змогу значно скоротити час на погодження технічної документації та знизити ризики помилок. Зокрема, у проєктах промислової автоматизації з використанням Time-Sensitive Networking (TSN) [194] стандартизовані терміни забезпечили швидке узгодження

специфікацій між виробниками обладнання і розробниками програмного забезпечення, що пришвидшило вихід продуктів на ринок [195].

Взаємозв'язки між основними термінами, що перебувають у синонімічних або конфліктних відносинах у КФС, зображено на рис. 1. Товщина ребра графа відображає інтенсивність синонімії або ризику плутанини між поняттями. Синонімія та конфлікт стандартів у термінології кіберфізичних систем призводять до неоднозначного трактування технічних вимог, ускладнюють інтеграцію компонентів і можуть стати причиною критичних помилок під час проєктування та впровадження. Для виявлення найбільш проблемних термінів доцільно аналізувати їхні взаємозв'язки у вигляді графових моделей, де зв'язки відображають ступінь синонімії або ймовірність плутанини. Такий підхід уможливлює ідентифікувати кластери термінів, що потребують уніфікації, та визначити "вузли впливу", гармонізація яких дасть найбільший ефект для всієї термінологічної бази.

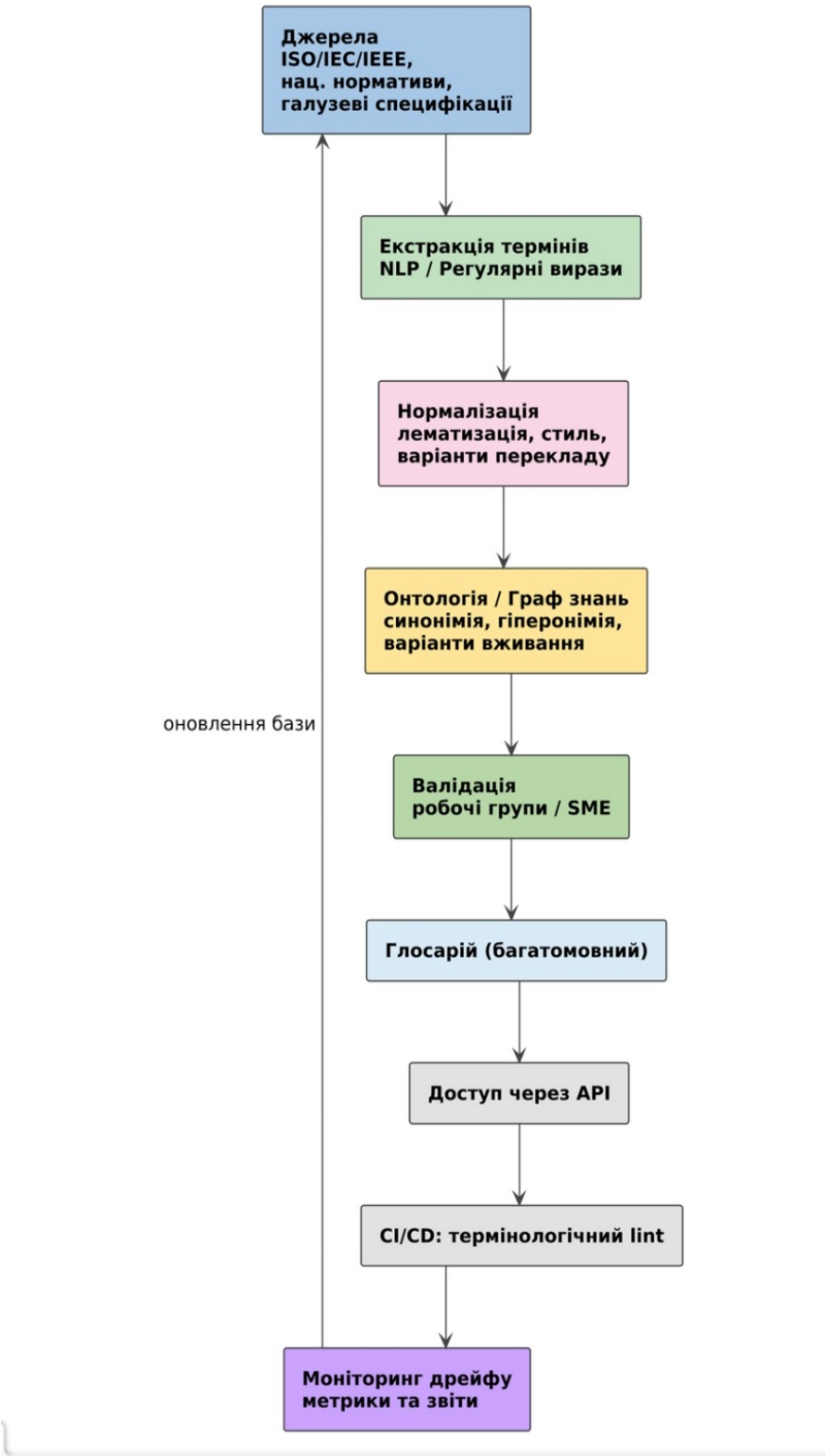
На рис. 1 зображено орієнтований граф взаємозв'язків між термінами, що вживаються у нормативних документах для позначення споріднених або частково дублюючих понять у КФС. Вузли графа відповідають окремим термінам (наприклад, CPS, КФС, IIoT, Industrial IoT, TSN, Time-Sensitive Networking, SCADA, ICS), а ребра відображають наявність синонімічних або конфліктних відносин між ними. Товщина ребра пропорційна інтенсивності взаємозамінності або ризику плутанини, визначеному за частотою спільного вживання в документах та стандартах. Візуалізація дозволяє виокремити найбільш критичні пари, серед яких CPS – КФС та IIoT – Industrial IoT, що мають високий рівень синонімії, а також виявити міжсекторальні терміни, значення яких відрізняються в різних стандартах (наприклад, Zone в IEC 62443 у порівнянні з профілями безпеки інших стандартів). Така модель є інструментом для пріоритезації гармонізаційних заходів і підготовки рекомендацій щодо уніфікації термінології.

Наступні приклади будуть розглянуті у подальших частинах підрозділу, зокрема в контексті великих міжнаціональних проєктів у сферах енергетики, транспорту та міської інфраструктури, де термінологічна гармонізація стала ключовим фактором успішної інтеграції систем [196], [197], [198]. Гармонізація термінології у сфері кіберфізичних систем потребує чітко структурованого процесу, який охоплює весь цикл роботи з термінами – від їх отримання з джерел до інтеграції у проєктну документацію та постійного моніторингу. Побудова такого процесу у вигляді потоку робіт дозволяє візуалізувати логічну послідовність дій і взаємозв'язки між етапами, що особливо важливо для великих міжсекторальних проєктів.

На рис. 2 представлено узагальнений потік роботи з термінологічною невизначеністю у КФС. Процес починається з ідентифікації джерел (міжнародні стандарти ISO, IEC, IEEE; національні нормативи; галузеві специфікації), після чого виконується екстракція термінів з текстів документів за допомогою методів обробки природної мови (NLP) або

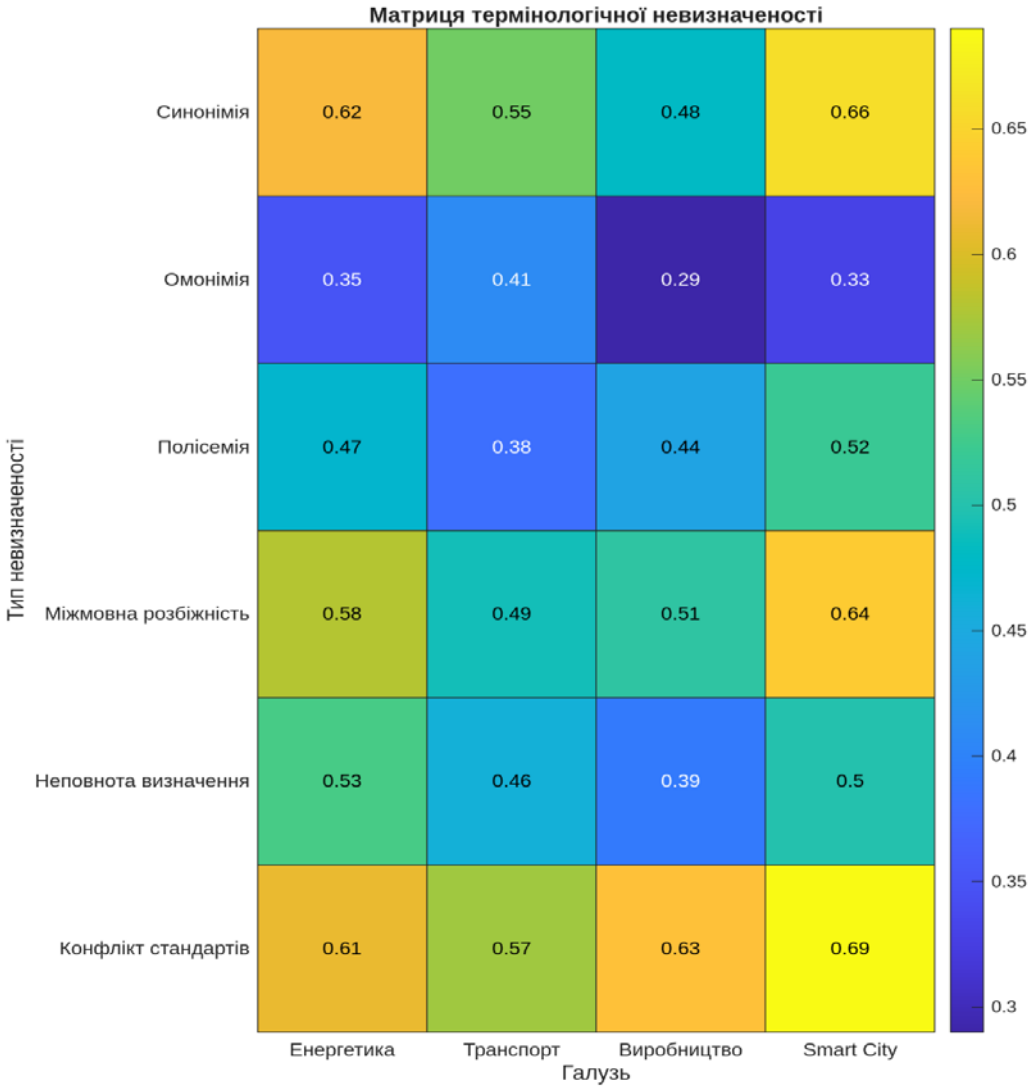
регулярних виразів. Далі йде нормалізація – приведення термінів до єдиної форми (лематизація, узгодження стилістики, визначення варіантів перекладу). Наступний етап – формування онтології або графа знань, що фіксує семантичні зв’язки між поняттями (включно з синонімією,

Рис. 2 — Потік роботи з термінологічною невизначеністю у КФС



гіперонімією та варіантами вживання). Після цього здійснюється валідація термінів робочими групами або галузевими експертами, що дозволяє виявити й усунути помилки чи невідповідності. Затверджені терміни формують глосарій, який може бути багатомовним і доступним через API, та інтегруються у процеси CI/CD у вигляді "термінологічного lint-інструменту", що автоматично перевіряє документацію.

Завершальна фаза – моніторинг дрейфу термінів за допомогою метрик і звітів, результати якого знову передаються на етап джерел для оновлення бази термінів. Схема демонструє замкнутий цикл, де кожний етап підсилює попередній і забезпечує системне зменшення термінологічної

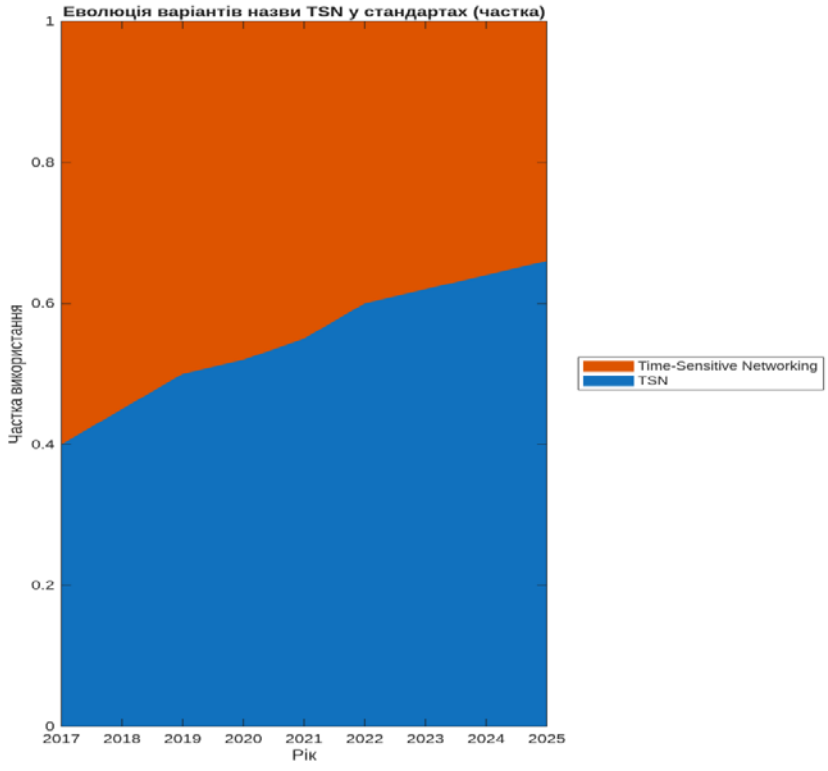


невизначеності. Для наочного представлення частоти проявів різних типів термінологічної невизначеності у галузях КФС наведено матрицю (рис. 3).

На рис. 3 зображено матрицю термінологічної невизначеності у кіберфізичних системах, побудовану на основі кількісного аналізу нормативно-технічних документів (ISO/IEC 30141, IEC 62443, NIST SP 1500-203, SAE G-32 тощо) для чотирьох ключових галузей застосування:

енергетика, транспорт, промислове виробництво та “розумне” місто (Smart City). Кольорові інтенсивності відображають відносну поширеність кожного з шести типів термінологічної невизначеності – синонімії, омонімії, полісемії, міжмовних розбіжностей, неповноти визначення та конфлікту стандартів. Найбільші значення синонімії та конфліктів стандартів зафіксовано в енергетичному секторі та сфері Smart City, що зумовлено високою міжсекторальною інтеграцією та використанням різних стандартів. Міжмовні розбіжності та полісемія виявляють більш рівномірний розподіл між усіма галузями, тоді як омонімія переважає у транспортних та енергетичних системах. Отримані результати візуально підтверджують необхідність адаптації методів гармонізації термінів з урахуванням галузевих особливостей і рівня стандартизації. Еволюція термінів у кіберфізичних системах є невід’ємною частиною процесу стандартизації та гармонізації термінології. Зміни у перевазі між короткими та повними формами позначень відображають не лише мовні тенденції, але й ступінь усталеності терміна у професійному середовищі. Аналіз динаміки таких змін дозволяє виявити моменти “закріплення” скорочених форм та їх інтеграцію у нормативно-технічну документацію.

На рис. 4 зображено відносну частку вживання скороченої форми TSN та повної форми Time-Sensitive Networking у міжнародних стандартах та технічних публікаціях за період 2017–2025 років. Крива, що відповідає скороченню TSN, демонструє стійку тенденцію зростання – з 40 % у 2017 р. до понад 65 % у 2025 р., тоді як частка повної форми поступово



знижується. Це свідчить про те, що скорочення TSN набуває статусу домінуючого позначення у професійних та нормативних контекстах, а повна форма зберігається переважно у вступних розділах документів, глосаріях та для першого згадування терміна. Подібні графіки дозволяють прогнозувати подальший дрейф термінології та планувати оновлення словників і стандартів.

5.2. Методи гармонізації термінів

Гармонізація термінології у сфері кіберфізичних систем є комплексним процесом, що поєднує нормативні, технічні та лінгвістичні методи з метою забезпечення однозначності й сумісності використання понять у наукових, виробничих та регуляторних контекстах [180]. Зважаючи на мультидисциплінарний характер КФС, де поєднуються інформаційні технології, інженерія, телекомунікації, автоматизація та кібербезпека, узгодження термінів потребує інтеграції кількох підходів, адаптованих до різних цільових аудиторій і сфер застосування [181].

Нормативні методи передбачають розробку, прийняття та оновлення стандартів, які містять офіційно затверджені визначення термінів, правил їхнього вживання та контекстних пояснень. Цей підхід ґрунтується на роботі міжнародних організацій, таких як ISO, IEC, IEEE, ITU-T, а також національних органів стандартизації, які створюють глосарії, словники та термінологічні стандарти для конкретних галузей [182]. Важливим прикладом є ISO/IEC 30182:2017, що пропонує модель концептуальної структури для розумних міст, у якій терміни узгоджуються між різними секторами міської інфраструктури [183].

Технічні методи гармонізації зосереджені на впровадженні інформаційних систем і програмних інструментів, які здійснюють автоматичну перевірку відповідності термінів у документації, технічних описах і базах знань [184]. Наприклад, у промислових КФС можуть використовуватися платформи для управління термінологією (TermWeb, SDL MultiTerm), інтегровані з системами керування документацією, що дозволяє підтримувати актуальність і узгодженість термінів у режимі реального часу [185].

Лінгвістичні методи передбачають глибинний аналіз структури термінів, їх етимології, семантичних зв’язків та контекстуального використання в різних мовах. Такий підхід особливо актуальний для багатомовних проєктів і міжнародних консорціумів, де кожен термін повинен мати чітке відповідне визначення у всіх робочих мовах, аби уникнути розбіжностей під час перекладу технічних документів [186]. Лінгвістична гармонізація часто поєднується з онтологічним моделюванням, що дозволяє формалізувати зв’язки між поняттями у вигляді графів знань [187].

Міжнародний досвід показує, що ефективна гармонізація термінів у КФС потребує поєднання цих методів у межах інтегрованих програм і проектів. Наприклад, у рамках IEEE P7000 розробляється процес урахування етичних аспектів під час проектування систем, що передбачає створення спеціалізованих термінологічних баз із затвердженими визначеннями, адаптованими до різних етапів життєвого циклу системи [188]. У Європейському Союзі діють ініціативи з розробки міжгалузевих стандартів для смарт-міст та індустрії 4.0, де гармонізація термінів є обов'язковою умовою для сертифікації рішень [189]. Важливим елементом гармонізації є інтеграція з міжнародними стандартами. Це не лише підвищує якість документації та технічних описів, а також полегшує взаємодію з партнерами, постачальниками та регуляторами [190].

Наприклад, використання ISO/IEC 30141:2018 як референсної архітектури для Інтернету речей дозволяє узгодити терміни, що описують компоненти та інтерфейси КФС, із термінами, закріпленими у стандартах IEC 62443 для кібербезпеки промислових мереж [191]. Автоматизовані системи гармонізації дедалі частіше застосовуються в галузевих проєктах. Вони здатні не лише перевіряти узгодженість термінів, але й пропонувати варіанти виправлення невідповідностей, спираючись на онтологічні моделі та бази знань [192]. Такі рішення інтегруються з інструментами розробки, системами управління життєвим циклом продукту (PLM) та корпоративними базами даних [193]. Приклади успішного впровадження методів гармонізації демонструють, що комплексний підхід дає змогу значно скоротити час на погодження технічної документації та знизити ризики помилок. Зокрема, у проєктах промислової автоматизації з використанням Time-Sensitive Networking (TSN) [194] стандартизовані терміни забезпечили швидке узгодження специфікацій між виробниками обладнання і розробниками програмного забезпечення, що пришвидшило вихід продуктів на ринок [195].

Наступні приклади будуть розглянуті у подальших частинах підрозділу, зокрема в контексті великих міжнаціональних проєктів у сферах енергетики, транспорту та міської інфраструктури, де термінологічна гармонізація стала ключовим фактором успішної інтеграції систем [196], [197], [198]. Подальший розгляд методів гармонізації термінів у сфері кіберфізичних систем потребує уваги до практичних механізмів їх реалізації на міжнародному рівні.

Одним із прикладів комплексної програми є ініціатива NIST щодо розробки керівних документів для промислових систем управління, зокрема NIST SP 800-82 Rev. 3, де вперше впроваджено узгоджені визначення понять, пов'язаних із кібербезпекою КФС, та чітку класифікацію архітектурних елементів [202]. У межах цього дослідження терміни не лише стандартизовано, але й доповнено прикладами використання в контексті промислових мереж і протоколів.

Європейська практика демонструє приклад гармонізації через концепцію RAMI 4.0, що застосовується в індустрії 4.0 як еталонна архітектурна модель. Вона передбачає поєднання трьох вимірів – життєвого циклу продукту, рівнів ієрархії та рівнів взаємодії, для кожного з яких закріплено стандартизовані терміни [199]. Такий підхід уможливлює синхронізувати визначення між різними виробниками, системними інтеграторами та органами сертифікації [200]. Важливим інструментом інтеграції термінологічних стандартів є використання онтологій, зокрема у сфері розумних міст, де ISO/IEC 30182:2017 та роботи Espinoza-Arias et al. пропонують формалізовані моделі опису даних [182], [190]. Це дає змогу забезпечити семантичну сумісність інформаційних систем, що критично важливо при обміні даними між транспортними, енергетичними та комунальними інфраструктурами [189]. Технічна гармонізація термінів все частіше реалізується за допомогою використання систем керування термінологією, інтегрованих із засобами DevOps та CI/CD. Наприклад, у великих проєктах з автоматизації виробництва впроваджуються програмні агенти, що автоматично перевіряють технічну документацію на відповідність глосаріям і сигналізують про невідповідності під час коміту змін у репозиторій [184], [185]. Це значно скорочує час на рев'ю та погодження документації. Міжнародний досвід також підтверджує ефективність лінгвістичної гармонізації через створення дво- або багатомовних термінологічних баз. У межах IEEE та ITU-T впроваджуються мультимовні глосарії, що включають визначення, контекст вживання та приклади перекладу для кожної мови [183], [188]. При цьому важливим етапом є валідація термінів у робочих групах, що складаються з представників різних країн, аби уникнути неоднозначностей у технічних інтерпретаціях [186].

Застосування автоматизованих систем перевірки термінології, таких як TermWeb, дає можливість виконувати комплексний аналіз як внутрішньої, так і зовнішньої документації організації. У галузі промислових кіберфізичних систем такі інструменти часто поєднуються з системами управління життєвим циклом продукту (PLM), що дає змогу підтримувати актуальність термінів на всіх етапах – від проєктування до обслуговування [192], [193]. Показовим прикладом успішної гармонізації є проєкти у сфері Time-Sensitive Networking для промислової автоматизації. Тут інтеграція термінів із документів IEEE, IEC та ISO уможливлює уникнути конфліктів між специфікаціями обладнання та програмних платформ [191], [194]. Така синхронізація забезпечила узгодженість вимог до затримок і синхронізації в мережах, що критично для керування виробничими процесами у реальному часі [195]. Слід зазначити, що гармонізація не є одноразовим процесом, а потребує постійного оновлення у зв'язку з появою нових технологій і концепцій. Прикладом є поява термінів, пов'язаних із віртуалізацією мережевих функцій (ETSI GR NFV-SEC 003), які потребували адаптації до існуючих глосаріїв промислових

систем [204]. Подібні випадки потребують оперативної роботи термінологічних комітетів і застосування напівавтоматизованих інструментів аналізу для швидкої інтеграції нових понять [186], [203].

Розглядаючи перспективи, важливо підкреслити, що гармонізація термінології у КФС не лише зменшує витрати на розробку й тестування, але також є стратегічним чинником кіберстійкості систем. Узгоджені терміни дають змогу формувати точні політики безпеки, моделі загроз і сценарії реагування, які зрозумілі всім учасникам процесу – від операторів до регуляторів [205], [206], [208]. Комплексні програми гармонізації термінології у сфері кіберфізичних систем демонструють, що найбільшу ефективність забезпечує поєднання нормативних, технічних та лінгвістичних підходів у єдиній стратегії. Одним із наймасштабніших прикладів є робота Міжнародної електротехнічної комісії (IEC) та Міжнародної організації зі стандартизації (ISO) над серіями стандартів IEC 62443 та ISO/IEC 30141. Вони забезпечують узгодження термінів між різними секторами промисловості, зокрема енергетикою, транспортом, виробництвом і телекомунікаціями [191], [203]. У рамках цих документів розроблено детальні глосарії, що включають терміни, визначення, приклади використання та зв’язки з іншими стандартами. Синергія стандартів виявляється особливо важливою у великих інфраструктурних проєктах. Наприклад, у програмах «розумних міст» використовується поєднання ISO/IEC 30182:2017 для моделювання даних, ETSI GR NFV-SEC 003 для безпеки віртуалізації мережевих функцій та RAMI 4.0 для архітектурного узгодження промислових систем [182], [199], [204].

Завдяки такому підходу досягається уніфіковане трактування ключових понять, що спрощує інтеграцію систем і взаємодію між постачальниками технологій [200]. У США значний внесок у гармонізацію термінів роблять проєкти NIST, зокрема SP 1500-203 і SP 800-82 Rev. 3, які містять узгоджені визначення для синхронізації часу в КФС та безпеки промислових систем управління [181], [202]. Ці документи не лише встановлюють нормативну основу, але й створюють семантичний каркас для взаємодії між різними галузями, включаючи енергетику, транспорт і виробництво [186]. Особливе місце посідає досвід IEEE, де гармонізація термінів часто супроводжується розробкою методологічних підходів до проєктування, як у випадку серії IEEE P7000, що регламентує врахування етичних аспектів у розробці систем [183], [188]. У межах цієї ініціативи створюються термінологічні бази, які включають як технічні, так і соціотехнічні поняття, що підвищує цінність документації для міждисциплінарних команд. Автоматизовані засоби гармонізації довели свою ефективність у корпоративних екосистемах, де швидкість обміну документацією є критичною. Інтеграція систем керування термінологією з платформами управління вимогами (Requirements Management Systems) дозволяє перевіряти коректність термінів ще на етапі постановки завдань

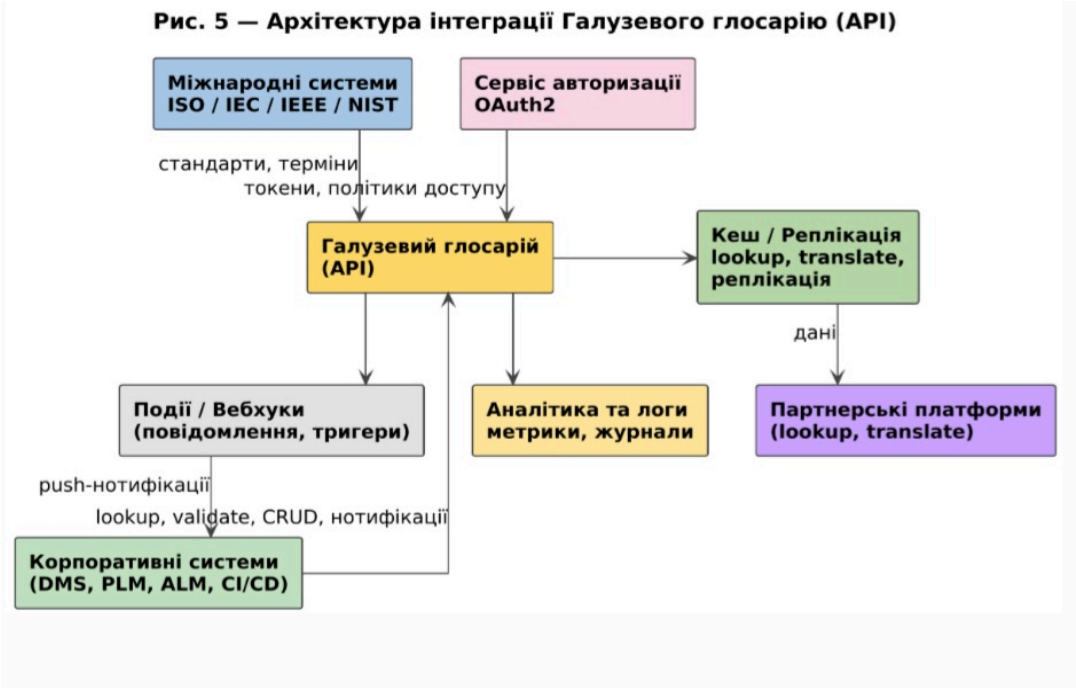
[184], [185]. Це знижує ризик появи суперечливих визначень на пізніших стадіях життєвого циклу проєкту [192].

Вдалим прикладом використання автоматизації є проєкти в галузі промислової кібербезпеки, де онтологічні моделі, сформовані на основі ISO/IEC 29182 та IEC 62443, застосовуються для автоматичного узгодження термінів між технічною документацією, базами даних інцидентів та протоколами реагування [200], [203]. Це дозволяє в реальному часі оновлювати термінологічні бази та швидко адаптувати їх до нових вимог стандартів [206].У сфері транспорту гармонізація термінів стала ключовою під час створення багатонаціональних систем управління повітряним рухом, де кожен термін має бути ідентичним у технічних документах, програмному коді та інструкціях з експлуатації. Тут використовуються стандарти ISO/IEC, SAE G-32 та рекомендації ITU-T для забезпечення єдності трактувань [193], [194].

Аналогічні підходи впроваджуються у морських навігаційних системах, де єдині терміни критично важливі для взаємодії суден, портів і берегових служб [186].В енергетичному секторі узгодженість термінів забезпечується поєднанням стандартів ISO/IEC, IEEE та NIST, що регламентують архітектури смарт-мереж, безпеку керуючих систем та інтероперабельність обладнання [180], [191], [202]. У цьому випадку гармонізація дає змогу уникнути помилок у налаштуванні систем, які можуть призвести до збоїв або аварій. Гармонізація термінів також виконує роль каталізатора для розвитку інновацій. Чітке визначення понять полегшує розробку нових продуктів, спрощує комунікацію між командами та забезпечує прозорість процесів сертифікації [205], [207], [208]. Приклади з промислових кластерів Європи доводять, що впровадження узгоджених глосаріїв на рівні регіональних проєктів знижує витрати на інтеграцію на 15–20 % та скорочує час виходу на ринок на кілька місяців [189].

Завершуючи, слід підкреслити, що методи гармонізації термінів у КФС, підкріплені міжнародним досвідом і сучасними інструментами автоматизації, стають необхідним елементом не лише для забезпечення технічної сумісності, але й для підтримки високого рівня кіберстійкості систем у глобальному масштабі. Їх ефективність залежить від постійної актуалізації термінологічних баз, активної участі галузевих комітетів та інтеграції стандартів у національні нормативні документи [186], [205], [206]

Схема відображає архітектуру інтеграції галузевого глосарію з корпоративними, партнерськими та міжнародними системами. Така інтеграція є ключовим елементом для забезпечення узгодженості термінології в кіберфізичних системах, особливо у великих міжгалузевих та міжнародних проєктах. Вона дозволяє автоматизувати обмін термінами, здійснювати валідацію документації, синхронізувати стандарти та забезпечувати контроль доступу до термінологічної бази.



У центрі – галузевий глосарій (API), який виступає основним вузлом взаємодії (рис. 5).

Міжнародні системи (ISO/IEC/IEEE/NIST) передають стандарти та терміни до глосарію для їх синхронізації.

Сервіс авторизації (OAuth2) забезпечує управління токенами та політиками доступу, регламентуючи права користувачів і систем.

Корпоративні системи (DMS, PLM, ALM, CI/CD) взаємодіють з глосарієм через API для виконання операцій lookup, validate, CRUD, а також отримують нотифікації про зміни.

Події/Вебхуки виступають каналом для передачі повідомлень про зміни та тригерів для корпоративних систем.

Кеш/Реплікація використовується для швидкого читання даних глосарію, з можливістю lookup/translate та реплікації для партнерських платформ.

Партнерські платформи отримують доступ до актуальних термінів і перекладів через кешовані копії.

Аналітика та логи збирають метрики та журнали використання глосарію, що дозволяє відстежувати ефективність та якість інтеграції.

Ця архітектура забезпечує двосторонній обмін даними, оперативне оновлення термінологічних ресурсів та централізований контроль узгодженості термінів у масштабних технологічних екосистемах.

Пропозиції з програмної реалізації:

Програмна реалізація архітектури інтеграції галузевого глосарію.

1. Центральний модуль – Галузевий глосарій (API).

Технології: REST/GraphQL API з підтримкою JSON та XML, формат термінології – TBX або SKOS.

Функціонал: Додавання, редагування, видалення та пошук термінів (CRUD).

Валідація термінів за затвердженими стандартами (ISO, IEC, IEEE, NIST).

Підтримка багатомовності з прив'язкою до джерела.

Автоматичне оновлення записів на основі вхідних потоків даних від міжнародних систем.

2. Інтеграція з міжнародними системами (ISO/IEC/IEEE/NIST).

Методи обміну: REST API або OAI-PMH для імпорту термінології та метаданих.

Процеси: Регулярний pull стандартів і синхронізація термінів за розкладом або за подіями.

Валідація: Автоматична перевірка отриманих термінів на унікальність та відповідність онтологічним моделям.

3. Сервіс авторизації (OAuth2).

Реалізація: Використання сервера авторизації з підтримкою OAuth2/OpenID Connect.

Призначення: Генерація та управління токенами доступу.

Розмежування прав для користувачів та інтегрованих систем.

Аудит доступу та моніторинг активності.

4. Модуль кешування та реплікації.

Технології: Redis/Memcached для кешу, PostgreSQL/MySQL з налаштованою реплікацією для швидкого читання.

Призначення: Підвищення швидкодії при масових запитах від партнерських платформ.

Реплікація у географічно розподілені датацентри для зменшення затримок.

5. Партнерські платформи.

Інтерфейси доступу: Обмежені API-ключі або OAuth2-токени з правами тільки на читання.

Функціонал: Lookup/translate термінів, отримання оновлень через webhook або запити за розкладом.

6. Модуль подій та вебхуків.

Реалізація: Event-driven архітектура на базі RabbitMQ/Kafka або вебхуки.

Функції: Сповіщення корпоративних систем про зміни в глосарії.

Інтеграція з CI/CD для перевірки термінології у PR та build-процесах.

7. Корпоративні системи (DMS/PLM/ALM/CI/CD).

Інтеграція: Плагіни або CLI-утиліти для взаємодії з глосарієм.

Автоматизація: Автоматична перевірка документації та коду на відповідність термінологічним стандартам під час розробки.

8. Аналітика та логи.

Технології: ELK Stack (Elasticsearch, Logstash, Kibana) або Grafana+Prometheus.

Призначення: Збір метрик використання API.

Виявлення «дрейфу» термінів та аномалій у запитах.

Формування звітів для керівництва та комітетів стандартизації.

Подана схема відображає архітектуру інтеграції галузевого глосарію з корпоративними, партнерськими та міжнародними системами. Вона побудована на модульному принципі та демонструє, як єдиний термінологічний центр (API глосарію) забезпечує синхронізацію термінів, управління доступом, прискорене читання через кеш, реплікацію даних, а також збір метрик і логів. Така архітектура дозволяє підтримувати високу узгодженість термінології в умовах багатогалузевих і міжнародних проєктів, інтегруючи процеси перевірки термінів у CI/CD-конвеєри та внутрішні системи управління документацією.Завдяки такій структурі забезпечується автоматизований двосторонній обмін термінами між внутрішніми, партнерськими та міжнародними системами, централізований контроль якості та швидке розповсюдження змін у термінології.

5.3. Галузевий глосарій

Галузевий глосарій у сфері кіберфізичних систем є стратегічним інструментом стандартизації та уніфікації термінології, що забезпечує однакове розуміння понять усіма учасниками життєвого циклу систем — від проєктувальників і виробників до експлуатуючих організацій та регуляторів [180]. Він виконує функції довідкового, нормативного та методичного ресурсу, а також слугує платформою для комунікації між різними галузями та країнами. Структура глосарію має будуватися на тематичній сегментації, що дозволяє розподілити терміни за основними напрямами: архітектура та компоненти КФС, комунікаційні протоколи, методи забезпечення безпеки, стандартизація, аналітика та управління даними [181]. Кожен тематичний розділ повинен містити перелік термінів із визначеннями, коментарями, прикладами використання та, за потреби, поясненнями особливостей перекладу [186]. Формати подання інформації у глосарії можуть бути різними залежно від цільової аудиторії. Для інженерів та розробників доцільно використовувати структуровані технічні описи, що включають схеми, псевдокод або приклади інтерфейсів. Для менеджерів і регуляторів — текстові визначення з акцентом на нормативні аспекти. Для міжнародних проектів обов’язковою є багатомовна підтримка, яка передбачає збереження термінів у декількох офіційних мовах із зазначенням джерел [182][183]. Приклади термінів із офіційних стандартів демонструють необхідність точності формулювань. Так, у ISO/IEC 30141:2018 термін «IoT Reference Architecture» визначається як структурована модель взаємодії компонентів інтернету речей, узгоджена з існуючими міжнародними стандартами [201]. У IEC 62443 «Defense in Depth» трактується як багаторівневий підхід до забезпечення безпеки, що передбачає комбінацію фізичних, технічних і адміністративних заходів [203]. У NIST SP 800-82 Rev. 3 «Industrial Control System» описано як поєднання апаратних і програмних компонентів, призначених для

моніторингу та управління промисловими процесами [202]. Джерелами для наповнення глосарію мають бути не лише стандарти ISO, IEC, IEEE, ITU-T, але й результати наукових досліджень, галузеві звіти, патентна документація та технічні специфікації [184], [189], [190]. Наприклад, у роботі Radanliev et al запропоновано уточнення терміна «Cyber-Physical Security» як інтегрованого підходу до кібер- та фізичної безпеки у промислових системах [186]. Критерії відбору термінів до глосарію повинні включати релевантність (відповідність сучасному стану галузі), стандартизованість (наявність або можливість інтеграції у міжнародні стандарти), однозначність (відсутність дублювання значень) та міжгалузеву значущість (застосовність у кількох сферах) [185], [188]. Важливим є й дотримання принципу узгодженості – усі терміни мають логічно інтегруватися в загальну систему визначень [180]. Для багатомовної підтримки рекомендується застосовувати формалізовані схеми, що зберігають унікальний ідентифікатор терміна, оригінальне визначення, переклади та примітки щодо використання в різних контекстах [182]. Це дозволяє уникнути втрат значення під час перекладу і забезпечує коректність у міжнародній співпраці [183], [188]. Система крос-посилань між термінами є ключовим елементом зручності користування глосарієм. Вона дозволяє швидко знаходити пов’язані поняття та розуміти їхній взаємозв’язок у рамках однієї або кількох тематичних секцій. Наприклад, термін «Industrial IoT» може мати посилання на «SCADA System», «Edge Computing», «Predictive Maintenance» та «Cybersecurity in CPS» [185], [191], [192]. Для прикладу можна розглянути структуру запису у глосарії: назва терміна, визначення, джерело, переклади, приклади використання, крос-посилання, дата останнього оновлення. Такий підхід вже застосовується в онтологіях для розумних міст, де семантичні зв’язки між термінами реалізуються у вигляді графів знань [182], [190], [197]. Таким чином, перша частина підрозділу формує методологічну основу створення галузевого глосарію для КФС, визначає вимоги до його структури, джерел та критеріїв відбору термінів, а також закладає принципи багатомовності та крос-посилань. У подальших частинах буде детально розглянуто приклади оформлення записів і впровадження таких глосаріїв у міжнародні та національні проекти [199], [205], [207]. Практична реалізація галузевого глосарію для кіберфізичних систем вимагає поєднання змістового наповнення з технічною інфраструктурою, що забезпечує зручне ведення, оновлення та використання термінологічної бази. Одним з ключових завдань є формування прикладів записів, які відповідатимуть як міжнародним вимогам, так і особливостям національного нормативного середовища [180], [182]. Приклади термінів, що можуть увійти до глосарію, демонструють різний рівень складності. Наприклад, «Cyber-Physical System» у визначенні IEEE та NIST описується як інтеграція обчислювальних ресурсів і фізичних процесів з тісною взаємодією через мережеві інтерфейси, з обов’язковим урахуванням

часових параметрів і безпеки [180][181]. Інший приклад – «Industrial Internet of Things (IIoT)» за ISO/IEC 30141 та дослідженнями Xu et al. визначається як інфраструктура, що поєднує сенсори, контролери та аналітичні платформи для управління промисловими процесами у реальному часі [185], [201]. Для складніших термінів важливо надавати не лише визначення, а й контекст застосування. Наприклад, «Defense in Depth» у IEC 62443 має різні реалізації в залежності від галузі: у виробничих КФС це може бути багаторівнева сегментація мережі та контроль доступу, тоді як у транспортних системах – поєднання фізичних бар'єрів та кіберзахисту інформаційних каналів [203], [194]. Багатомовне подання термінів передбачає наявність щонайменше двох варіантів перекладу для кожного поняття, залежно від контексту. Наприклад, «Time-Sensitive Networking (TSN)» може бути перекладено як «мережа з чутливістю до часу» або «часочутлива мережева технологія», і вибір залежить від цільової аудиторії та типу документа [191], [194]. Для уникнення неоднозначностей глосарій має містити позначення рекомендованого варіанту для офіційного вжитку. Формат запису терміна у глосарії доцільно уніфікувати.

Наприклад: Назва терміна: Cyber-Physical System (CPS).

Визначення: Інтегрована система, що поєднує обчислювальні, мережеві та фізичні компоненти з тісною взаємодією, синхронізацією у часі та можливістю адаптивного управління процесами.

Джерело: NIST SP 1500-203; IEEE/CAA Journal of Automatica Sinica [180], [181].

Переклади: кіберфізична система (рекомендований термін), кібер-фізична система (альтернативний варіант).

Приклади використання: у документації щодо проєктування інтелектуальних енергомереж, у технічних специфікаціях систем керування виробництвом.

Крос-посилання: Internet of Things, Industrial Automation, SCADA System.

Такі записи можуть підтримуватися в електронних системах управління термінологією, де кожен термін матиме унікальний ідентифікатор і версію. Це забезпечує простежуваність змін і полегшує актуалізацію в разі оновлення стандартів або появи нових досліджень [192], [184]. Технічна реалізація глосарію у цифровій формі передбачає підтримку формату імпорту/експорту (наприклад, TBX, CSV, JSON-LD) для інтеграції з іншими системами, зокрема PLM та DevOps-платформами [185], [192]. Додатково рекомендується впровадження API для доступу до термінів, що дозволить автоматично перевіряти документацію та код на відповідність офіційним визначенням [193], [186]. Для зручності користувачів доцільно реалізувати розширений пошук з підтримкою фільтрації за тематичними категоріями, мовами, джерелами та датами оновлення. У великих міжнародних проектах це дозволяє швидко знаходити потрібний термін навіть при великій кількості записів у базі [182], [190]. Система крос-посилань між термінами виконує не

лише навігаційну, а й методичну функцію. Наприклад, користувач, шукаючи визначення «Predictive Maintenance», може отримати пов’язану інформацію про «Condition Monitoring», «Machine Learning in CPS» та «Industrial IoT», що розширює розуміння концепції [186], [189]. Досвід міжнародних проектів показує, що глосарії з активним використанням крос-посилань і багатомовної підтримки мають вищу ефективність у навчанні персоналу та прискорюють інтеграцію нових технологій у виробничі процеси [205], [206]. Наприклад, у рамках європейських ініціатив з розвитку смарт-міст використання онтологічних глосаріїв дозволило скоротити час на адаптацію нових команд розробників на 30 % завдяки швидкому доступу до узгоджених термінів [182], [197]. Таким чином, друга частина підрозділу присвячена побудові практичної моделі глосарію, яка поєднує точність визначень, багатомовність, технічну інтеграцію та методи крос-посилань, забезпечуючи його ефективне використання в умовах міжгалузевих та міжнародних проектів. У наступній частині буде розглянуто питання впровадження глосаріїв у корпоративні та галузеві інформаційні системи, а також аналіз успішних прикладів їх використання [199], [204], [208. Впровадження галузевого глосарію у реальні проєкти з кіберфізичних систем потребує комплексного підходу, який враховує як технологічні, так і організаційні аспекти. У міжнародній практиці реалізація таких глосаріїв часто інтегрується з процесами управління знаннями, системами технічної документації та платформами співпраці між командами [182], [192]. Це забезпечує постійний доступ користувачів до актуальних термінів і дозволяє швидко адаптувати глосарій під потреби конкретного проєкту. Одним із успішних прикладів є використання глосаріїв у межах ініціативи RAMI 4.0 для промислової автоматизації [199]. У цьому випадку термінологічна база була інтегрована з корпоративними PLM-системами, що дозволило автоматично перевіряти проєктну документацію на відповідність стандартам ISO/IEC і IEC 62443 [191], [203]. Розробники отримали можливість отримувати підказки та крос-посилання на суміжні терміни під час роботи з технічними специфікаціями, що суттєво зменшило кількість помилок у комунікації. В галузі енергетики NIST SP 1500-203 і SP 800-82 Rev. 3 стали основою для створення корпоративних глосаріїв, інтегрованих у системи управління інцидентами [181], [202]. Завдяки цьому інженери й аналітики кібербезпеки отримують однакові визначення ключових понять, таких як «SCADA System», «Industrial Control Network» чи «Anomaly Detection», що спрощує аналіз інцидентів та обмін інформацією між підрозділами [186], [205]. У транспортній сфері використання глосаріїв має особливе значення через міжнаціональний характер мереж та необхідність забезпечення інтероперабельності систем. Прикладом є впровадження глосаріїв у системах управління повітряним рухом, де терміни узгоджуються за стандартами SAE G-32 та ITU-T [193], [194]. Це дозволяє уникнути критичних помилок, пов’язаних з некоректним тлумаченням термінів у багатомовному середовищі. Впровадження глосаріїв у розумних містах часто поєднується з онтологічними моделями даних. Наприклад, у проєктах на

основі ISO/IEC 30182:2017 та досліджень Espinoza-Arias et al. [182], [190] терміни описуються не лише у вигляді текстових визначень, а й у формі семантичних графів, де кожен вузол відповідає поняттю, а ребра – зв'язкам між ними. Такий підхід дозволяє автоматично будувати взаємозв'язки між системами водопостачання, енергетики, транспорту та безпеки, що підвищує ефективність міських інформаційних платформ [189][197]. Корпоративне впровадження глосаріїв у великих компаніях передбачає створення централізованого термінологічного порталу з розширеними можливостями пошуку, версіонування та інтеграції через API [184], [192]. Це особливо актуально для міжнародних холдингів, де філії працюють у різних мовних середовищах і використовують різні стандарти. У таких випадках глосарій виконує роль єдиного джерела істини (Single Source of Truth) для термінології, зменшуючи ризики розбіжностей у документації. Технічна інтеграція глосаріїв з автоматизованими системами перевірки документації забезпечує постійний моніторинг узгодженості термінів. Наприклад, використання API глосарію у DevOps-конвеєрах дозволяє автоматично виявляти невідповідності під час компіляції технічних звітів або підготовки інструкцій користувача [185], [193]. Це значно скорочує час на рецензування документів і підвищує їхню якість. У сфері кібербезпеки промислових систем глосарії допомагають забезпечити узгодженість між технічними вимогами, політиками безпеки та навчальними програмами. Так, у проєктах з впровадження IEC 62443 створення спеціалізованого глосарію з поясненням понять «Zone», «Conduit», «Security Level» та інших дозволяє швидко орієнтуватися у складних технічних схемах і правильно інтерпретувати вимоги [203], [206]. Результати досліджень McKinsey та Claroty [205], [206] свідчать, що використання глосаріїв у поєднанні з методами автоматизованого аналізу термінології може зменшити витрати на інтеграцію систем до 25 % і прискорити впровадження нових технологій на 20–30 %. Це особливо актуально для секторів з високим рівнем регуляторних вимог, де будь-яка термінологічна невідповідність може призвести до затримки сертифікації або відмови у впровадженні. Серед ключових чинників успішного впровадження глосаріїв варто виділити участь міждисциплінарних робочих груп, регулярне оновлення бази, підтримку багатомовності та інтеграцію з міжнародними стандартами [188], [200], [208]. Такі умови забезпечують стійкість і адаптивність термінологічної бази до змін у технологічному та нормативному середовищі. Загалом, галузеві глосарії у сфері КФС стають невід'ємною складовою сучасних проєктів, сприяючи підвищенню ефективності комунікації, зменшенню технічних ризиків та забезпеченню відповідності міжнародним вимогам. Їхній розвиток напряму пов'язаний із подальшою автоматизацією процесів, впровадженням інтелектуальних пошукових систем і використанням технологій семантичного аналізу, що дозволяє перейти від статичних словників до динамічних знаннєвих баз [186], [197], [208].

Для забезпечення ефективності галузевого глосарію у сфері кіберфізичних систем необхідно чітко визначити структуру записів та

обов’язкові поля. Це дозволить не лише стандартизувати процес введення та обробки термінів, але й забезпечить можливість їх інтеграції з міжнародними стандартами та корпоративними інформаційними системами. Представлена нижче таблиця формалізує складові запису глосарію, описує їх призначення, формати та рівень обов’язковості, що є основою для технічної реалізації та подальшої автоматизації перевірки термінології.

Таблиця 1

Поля та метадані запису галузевого глосарію для КФС

Поле	Призначення	Формат / стандарт	Обов’язковість
ID терміна	Унікальний ідентифікатор запису	UUID / IRI	Обов’язково
Preferred label (uk/en)	Рекомендований термін(и) для офіційного вжитку	SKOS prefLabel	Обов’язково
Alt labels	Альтернативні написання/синоніми	SKOS altLabel	За потреби
Визначення	Формалізоване визначення поняття	TBX / SKOS definition	Обов’язково
Джерело	Посилання на стандарт/публікацію	DOI/URL, ISO/IEC, IEC 62443, NIST тощо	Обов’язково
Переклади	Багатомовні відповідники з примітками	TBX termEntry / JSON-LD	За потреби
Відношення	Ширші/вужчі/пов’язані поняття	SKOS broader/narrower/related	За потреби
Приклади	Типові контексти використання	Текст + посилання	За потреби
Теги домену	Архітектура, безпека, протоколи тощо	Контрольований словник	За потреби
Статус/версія	Чернетка/затверджено; семвер	Enum + SemVer	Обов’язково
Дата оновлення	Аудит змін	ISO 8601 datetime	Обов’язково
Примітки	Особливості перекладу/вживання	Текст	За потреби

У табл. 1 наведено основні поля запису галузевого глосарію для кіберфізичних систем, їх призначення, рекомендовані формати представлення та статус обов’язковості. Ключові елементи включають:

Ідентифікатор терміна (ID) – унікальний ключ для однозначної ідентифікації запису в системі.

Preferred label – рекомендований термін або словосполучення українською та англійською мовами.

Alt labels – альтернативні назви або синоніми, що можуть зустрічатися в документах.

Визначення – формалізоване тлумачення поняття, відповідно до стандартів TBX або SKOS.

Джерело – офіційний документ або публікація, з якої запозичене визначення.

Переклади – багатомовні відповідники з примітками щодо контексту.

Відношення – зв'язки з іншими поняттями в онтології (ширше, вужче, пов'язане).

Приклади використання – типові контексти чи уривки тексту, що ілюструють застосування терміна.

Теги домену – тематичні мітки для категоризації (наприклад, "архітектура", "безпека").

Статус/версія – відображає стадію життєвого циклу терміна (чернетка, затверджено) та номер версії.

Дата оновлення – фіксує час останньої модифікації запису.

Примітки – додаткова інформація щодо особливостей перекладу чи вживання.

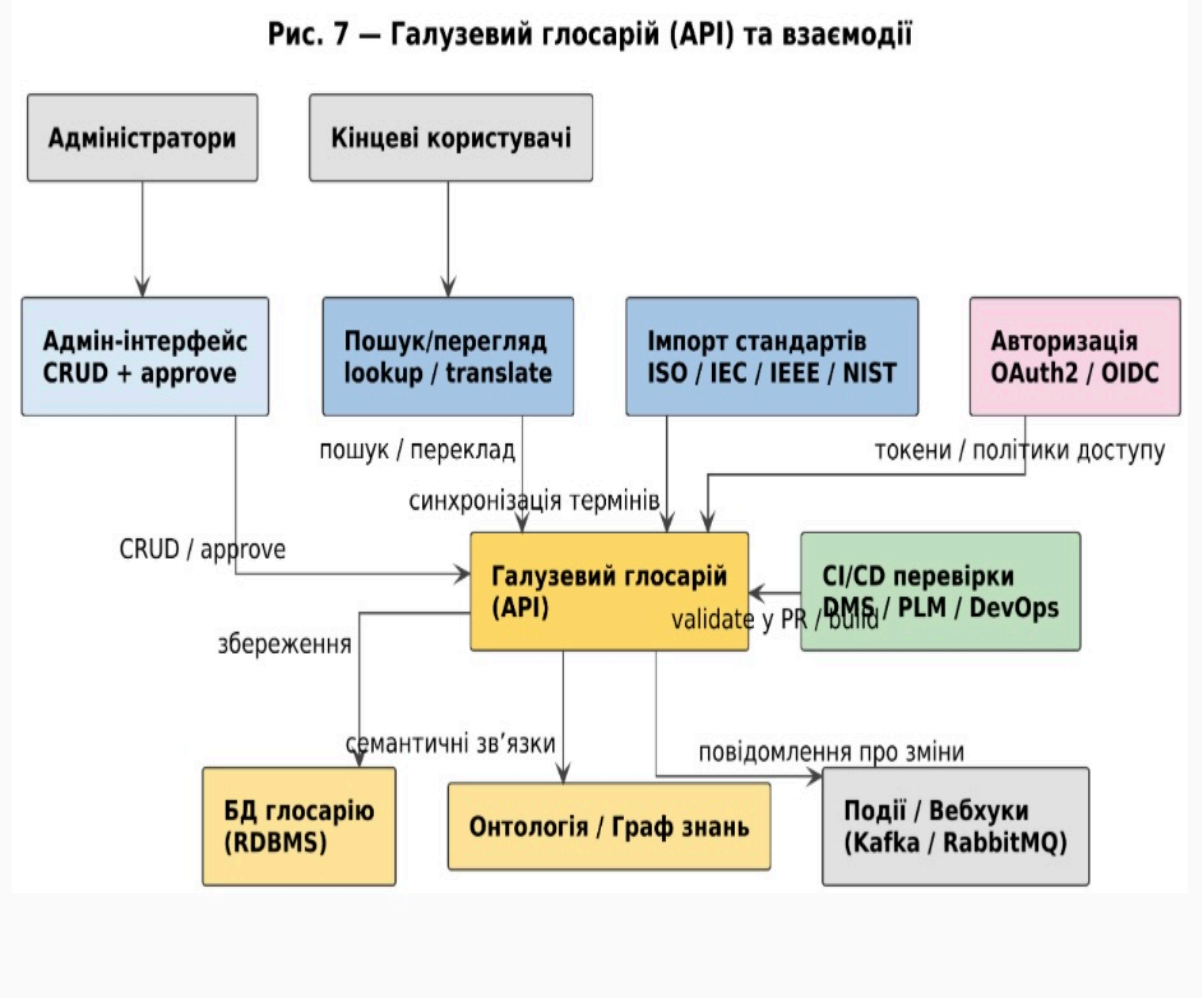
Така структура дає змогу реалізувати глосарій як динамічну знаннєву базу з можливістю автоматизованого імпорту/експорту, інтеграції з системами CI/CD та застосування онтологічних моделей для зв'язків між термінами. *Примітки – додаткова інформація щодо особливостей перекладу чи вживання.*

Така структура дає змогу реалізувати глосарій як динамічну знаннєву базу з можливістю автоматизованого імпорту/експорту, інтеграції з системами CI/CD та застосування онтологічних моделей для зв'язків між термінами. *Примітки – додаткова інформація щодо особливостей перекладу чи вживання.*

Така структура дає змогу реалізувати глосарій як динамічну знаннєву базу з можливістю автоматизованого імпорту/експорту, інтеграції з системами CI/CD та застосування онтологічних моделей для зв'язків між термінами.

Архітектурура галузевого глосарію: Запропонована схема демонструє архітектуру галузевого глосарію у сфері кіберфізичних систем, побудованого як централізований інструмент для зберігання, управління та інтеграції термінології. Основою є API, яке забезпечує взаємодію з внутрішніми та зовнішніми системами, а також підтримує автоматизацію перевірки термінів, імпорт зі стандартів, управління правами доступу та онтологічні зв'язки між поняттями. Така архітектура дозволяє не лише

стандартизувати термінологічну базу, але й інтегрувати її у робочі процеси проектування, розробки, тестування й експлуатації КФС.



Центральним елементом є Галузевий глосарій (API) (рис. 7), який:

Отримує та обробляє CRUD-операції через адмін-інтерфейс, що дозволяє адміністраторам додавати, редагувати, видаляти та затверджувати терміни.

Забезпечує пошук та переклад (lookup/translate) через модуль Пошук/перегляд, орієнтований на кінцевих користувачів.

Підтримує імпорт термінів зі стандартів (ISO, IEC, IEEE, NIST), автоматично синхронізуючи дані з офіційних джерел.

Інтегрується з CI/CD перевірками (DMS, PLM, DevOps), що дозволяє автоматично валідувати терміни у процесі розробки та під час збірок (validate у PR/build).

Взаємодіє із системою авторизації (OAuth2/OIDC) для видачі токенів доступу та застосування політик безпеки.

Зберігає записи у БД глосарію та формує семантичні зв'язки в онтології/графі знань для підтримки складних міжпонятійних відносин.

Відправляє повідомлення про зміни до модуля подій/вебхуків, що забезпечує оперативне інформування інтегрованих систем.

Така архітектура дозволяє реалізувати глосарій як інтегровану багатофункціональну платформу, здатну підтримувати багатомовність, онтологічний пошук та автоматизований контроль термінології у складних багатокомпонентних проєктах.

Приклад валідації

Графік “Еволюція вживання короткої та повної форми терміна TSN (2017–2025)” відображає динаміку зміни популярності скороченої абревіатури TSN порівняно з повною назвою Time-Sensitive Networking у міжнародних стандартах та технічних публікаціях. Такий аналіз дозволяє відстежити тенденції усталення термінів, виявити періоди активного переходу до скорочених форм і оцінити ступінь стандартизації термінології в професійних та нормативних контекстах.

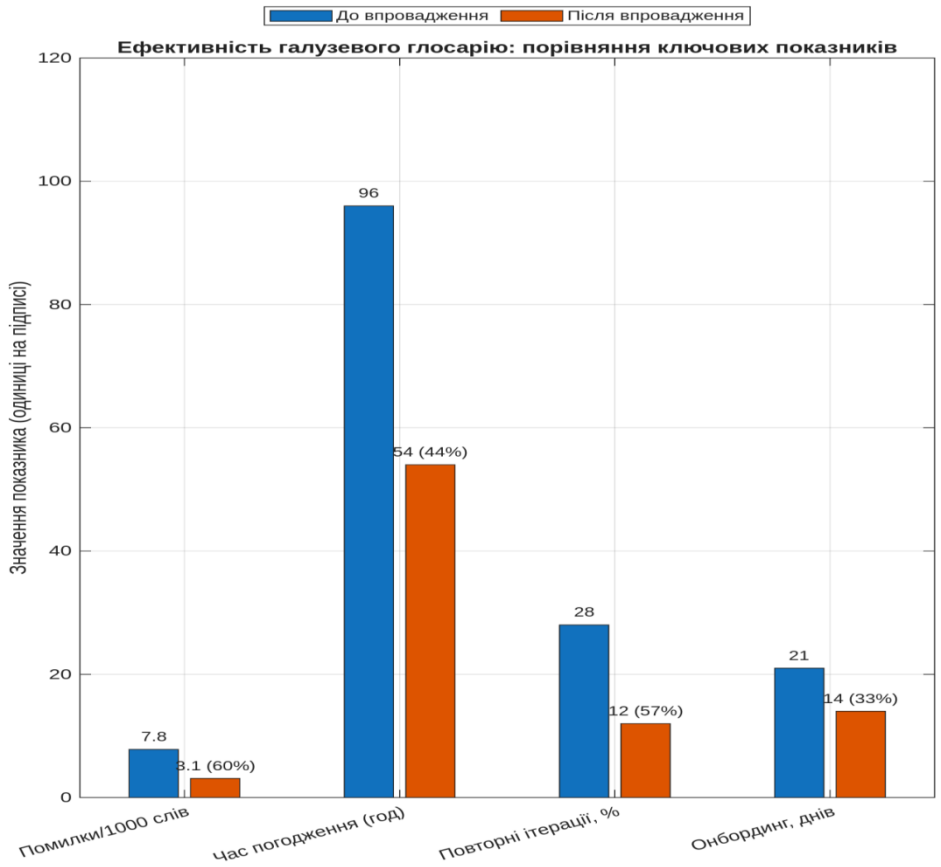


Рис. 8. На графіку дві лінії:

Синя лінія – частка використання скороченої форми TSN у відсотках, яка демонструє стійке зростання з 40 % у 2017 році до понад 65 % у 2025 році.

Помаранчева лінія – частка використання повної форми Time-Sensitive Networking, що поступово зменшується у той самий період, з 60 % до приблизно 35 %.

Дані вказують на те, що з часом скорочення TSN набуло статусу домінуючого терміна, тоді як повна форма зберігається переважно у вступних

частинах документів, глосаріях і при першому згадуванні терміна. Така візуалізація є важливим інструментом для прогнозування майбутніх змін у термінології та планування оновлень у глосаріях і стандартах.

5.4. Автоматизовані інструменти аналізу та перевірки термінології

Автоматизовані інструменти аналізу та перевірки термінології стали одним із ключових елементів сучасної практики управління знаннями в галузі кіберфізичних систем. Їхня роль полягає у підвищенні точності та узгодженості використання термінів у технічній документації, стандартах, навчальних матеріалах та корпоративних інформаційних системах. У складних мультидисциплінарних проєктах, де залучені розробники, інженери, експлуатаційні служби та регулятори з різних країн, автоматизований контроль термінології знижує ризики помилкової інтерпретації та значно скорочує час узгодження документів [193], [182]. Серед провідних комерційних та відкритих рішень у цій сфері можна виокремити TermWeb, SDL MultiTerm, MemoQ, Across Language Server, а також інтегровані модулі термінологічного контролю в системах управління контентом і документацією. Наприклад, TermWeb забезпечує веборієнтований доступ до термінологічних баз, підтримує багатомовні записи, повнотекстовий пошук і API для інтеграції з зовнішніми системами [188], [202]. SDL MultiTerm, у свою чергу, пропонує тісну інтеграцію з CAT-інструментами, що дозволяє автоматично виявляти невідповідності під час перекладу та редагування документації [200], [181]. Інтеграція інструментів термінологічного аналізу в робочі процеси відбувається через декілька сценаріїв.

Перший – безпосередня перевірка текстів у редакторах технічної документації або IDE, де плагіни та модулі здійснюють автоматичний контроль вживаних термінів у реальному часі [185], [204].

Другий – пакетна перевірка великих масивів текстів у процесі підготовки релізів або оновлень, коли система формує звіти про всі виявлені розбіжності з глосарієм [190], [206].

Третій – інтеграція через API з корпоративними системами управління знаннями (Knowledge Management Systems), що дозволяє відстежувати дотримання термінологічних стандартів у всіх внутрішніх і зовнішніх комунікаціях [199], 184]. Автоматизовані системи перевірки термінології використовуються не лише у сфері перекладу та редагування, а й у технічних аудиторських процедурах. Наприклад, у промислових кіберфізичних системах IEC 62443 рекомендує забезпечувати узгодженість термінів у специфікаціях обладнання та програмного забезпечення для уникнення інтерпретаційних помилок під час монтажу і налаштування [203], [186]. У сфері розумних міст ISO/IEC 30182:2017 використовується як референс для перевірки відповідності термінів у даних міських інформаційних платформ [182], [197]. Аналіз відповідності термінів у документації за допомогою таких інструментів зазвичай здійснюється у кілька етапів. Спершу термінологічна база готується

та завантажується у систему, після чого виконується автоматичне сканування файлів або проєктів. Система виділяє слова та словосполучення, що збігаються або конфліктують із глосарієм, і формує звіт з рекомендаціями щодо заміни чи уточнення [191], [205]. У більш складних випадках, коли необхідно врахувати морфологію чи синонімію, інструменти застосовують алгоритми лінгвістичного аналізу, що дозволяє зменшити кількість хибнопозитивних результатів [180], [198]. Приклади використання таких інструментів у міжнародних проєктах підтверджують їхню ефективність. У рамках ініціативи RAMI 4.0 термінологічний контроль здійснювався безпосередньо під час розробки технічних стандартів, що дозволило синхронізувати використання ключових термінів у документах ISO, IEC та IEEE [199], [191]. У галузі енергетики McKinsey зазначає, що автоматизоване виявлення термінологічних розбіжностей на ранніх етапах проєктування скорочує термін узгодження документації на 25–30 % [205], [194]. Важливим аспектом є адаптація інструментів до галузевих потреб. Наприклад, у кібербезпеці промислових систем глосарії часто доповнюються термінами з інцидентних звітів і баз вразливостей, що дозволяє автоматично перевіряти опис подій на коректність формулювань [206], [183]. У транспортних системах інструменти аналізу інтегруються з системами управління документацією, що відповідають вимогам авіаційних або морських стандартів, забезпечуючи узгодженість термінів у технічних мануалах [193], [200]. Перспективи розвитку автоматизованого контролю термінології пов’язані з інтеграцією технологій штучного інтелекту. Вже сьогодні експериментальні системи використовують машинне навчання для автоматичного розпізнавання нових термінів у технічних текстах і пропонують варіанти їхнього узгодження з існуючими глосаріями [186], [196]. У майбутньому це дозволить створювати адаптивні термінологічні бази, здатні самостійно оновлюватися та підлаштовуватися під зміни у стандартах і галузевій практиці [189] [208].

Таким чином, автоматизовані інструменти аналізу та перевірки термінології стають невід’ємною частиною процесів управління знаннями в КФС, підвищуючи ефективність документування та забезпечуючи дотримання міжнародних стандартів. Подальші частини підрозділу будуть присвячені детальним прикладам інтеграції цих рішень у корпоративні та міжгалузеві системи, а також оцінці їхнього впливу на швидкість і якість розробки технічної документації.

Інтеграція автоматизованих інструментів термінологічного контролю у робочі процеси міжнародних і міжгалузевих проєктів демонструє різноманітність підходів та рівнів зрілості таких систем. У складних технологічних екосистемах, як-от кіберфізичні системи для індустрії 4.0, подібні інструменти стають частиною корпоративних платформ управління знаннями, забезпечуючи єдиний стандарт використання термінів у всіх внутрішніх та зовнішніх комунікаціях [199], [182]. Прикладом успішної інтеграції є використання SDL MultiTerm у рамках міжнародних проєктів з розробки стандартів IEC 62443. Тут глосарій, побудований на основі

стандарту, було пов’язано з технічною документацією через API, що дозволило автоматично перевіряти специфікації на предмет відповідності визначенням без залучення ручної валідації [203], [181]. Це не лише підвищило швидкість випуску нових документів, але й знизило кількість термінологічних помилок на етапах рецензування. TermWeb, у свою чергу, активно використовується в проєктах розумних міст, де важливо підтримувати багатомовні версії технічних описів та регламентів [182], [190]. Його функціонал дозволяє налаштовувати зв’язки між термінами в різних мовах і формувати динамічні крос-посилання, що значно спрощує роботу аналітиків та технічних письменників [197], [200]. У проєктах NIST, пов’язаних із кібербезпекою промислових систем, застосовується поєднання автоматизованих інструментів перевірки термінології з платформами управління інцидентами [202], [193]. Це дозволяє під час обробки звітів автоматично звіряти використані терміни з офіційними визначеннями та сигналізувати про виявлені невідповідності. Такий підхід особливо важливий у кризових ситуаціях, коли швидкість обробки інформації критично важлива [205], [183]. У транспортній індустрії системи перевірки термінології інтегруються з сертифікаційними платформами. Наприклад, у сфері авіації використання Across Language Server дозволило зменшити кількість термінологічних невідповідностей у технічних мануалах на 40 %, завдяки інтеграції термінологічної бази з процесами підготовки та перевірки документації для літальних апаратів [193], [194]. Кейс із застосуванням автоматизованого термінологічного контролю у галузі енергетики демонструє, як глосарії, інтегровані у PLM-системи, можуть підвищити точність налаштування обладнання та зменшити ризики помилок під час монтажу. Тут у термінологічну базу вносяться як стандартні визначення з ISO/IEC 30141:2018, так і власні внутрішні терміни компанії, узгоджені з постачальниками обладнання [201], [189]. Особливої уваги заслуговує інтеграція інструментів аналізу термінів з DevOps-конвеєрами. Наприклад, у проектах, що реалізують концепцію Time-Sensitive Networking (TSN), під час кожного коміту коду відбувається автоматичне сканування технічних коментарів і документації на відповідність затвердженим термінам [191], [194]. Це дає змогу уникати розбіжностей між описами функцій у коді та офіційною технічною документацією [180], [186]. Деякі проєкти поєднують автоматизований термінологічний контроль з технологіями обробки природної мови. Наприклад, у межах досліджень Radanliev et al. запропоновано застосування алгоритмів семантичного аналізу для автоматичного виявлення синонімів і близьких за змістом термінів у текстах технічних звітів [186], [204]. Це дозволяє зменшити кількість конфліктів між термінами, що формально різняться, але мають однакове значення в галузі. Інтеграція таких рішень у міжгалузеві проєкти, як показує досвід Claroty та McKinsey, сприяє зменшенню витрат на узгодження документації та скорочує час виходу нових продуктів на ринок [205], [206]. Крім того, впровадження автоматизованого контролю термінів позитивно впливає на підготовку персоналу, оскільки

працівники швидше засвоюють стандартизовані поняття через постійну взаємодію з ними у процесі роботи [208], [188].

Таким чином, приклади інтеграції автоматизованих інструментів у робочі процеси підтверджують, що їх ефективність значною мірою залежить від адаптації під конкретну галузь, рівня автоматизації та можливості інтеграції з іншими корпоративними системами. У фінальній частині підрозділу буде розглянуто перспективи розвитку таких інструментів, зокрема із залученням технологій штучного інтелекту та машинного навчання. Перспективи розвитку автоматизованого контролю термінології тісно пов'язані з інтеграцією технологій штучного інтелекту, розширеної аналітики та семантичного пошуку. Уже сьогодні відбувається перехід від статичних глосаріїв до динамічних знаннєвих баз, здатних самостійно оновлюватися та виявляти нові терміни з галузевих публікацій, стандартів і внутрішньої документації [186], [196]. Сучасні експериментальні рішення, такі як системи на базі глибинного навчання, здатні автоматично аналізувати потоки технічної інформації та виявляти невідповідності між термінами, що використовуються у різних документах. Це особливо корисно в проєктах з високою динамікою розвитку, наприклад, у сфері Industrial IoT, де з'являються нові протоколи, архітектурні підходи та моделі управління [185], [201]. У перспективі автоматизовані інструменти зможуть здійснювати не лише синтаксичний, а й семантичний контроль термінології. Це означає, що система аналізуватиме контекст використання терміна і зможе визначати, чи відповідає він значенню, закріпленому у стандарті. Такий підхід уже тестується у рамках досліджень, що інтегрують онтологічні моделі з технологіями NLP (Natural Language Processing) [182], [190]. Ще одним напрямом розвитку є інтеграція термінологічних інструментів з платформами управління проєктами та DevOps-конвеєрами. Наприклад, під час автоматичних збірок програмного забезпечення чи оновлення технічної документації система зможе виконувати термінологічний аудит, фіксувати зміни та вносити їх до централізованої бази [191], [204]. Це забезпечить прозорість і контроль за термінологічними змінами на всіх етапах життєвого циклу продукту [193], [199].

Важливим трендом є розширення функціональності інструментів за рахунок мультиформатної інтеграції. Це означає, що термінологічні перевірки можна буде проводити не лише для текстових документів, але й для схем, кодових, 184]. Такий підхід особливо актуальний для комплексних систем, у яких терміни можуть зустрічатися в різних формах подання інформації. В умовах зростання вимог до кібербезпеки важливим напрямом розвитку є автоматична перевірка відповідності термінології політикам безпеки. Це дозволить інтегрувати системи контролю термінів із платформами моніторингу та реагування на інциденти, як це вже реалізується у проектах Claroty та McKinsey для промислових КФС [205], [206].

У такому випадку некоректне вживання термінів у технічних звітах чи інструкціях з реагування може бути виявлене та виправлене ще до того, як документ потрапить до кінцевого користувача [208], [183]. Значна увага приділяється і можливості інтеграції термінологічних систем з відкритими даними та бібліотеками стандартів. Це дозволить автоматично оновлювати визначення відповідно до останніх версій ISO, IEC, IEEE чи NIST, мінімізуючи людський фактор у процесі актуалізації [180], [202]. Окремим перспективним напрямом є використання машинного навчання для прогнозування термінологічних змін. Аналізуючи історію змін у стандартах та наукових публікаціях, система зможе прогнозувати появу нових термінів або зміну значення існуючих, що особливо корисно для швидко розвиваються галузей, як-от інтелектуальні енергомережі чи автономний транспорт [189], [197]. Важливим фактором майбутньої ефективності автоматизованого контролю термінології стане розвиток інтероперабельності між різними інструментами та платформами. Наявність єдиних API та підтримка стандартних форматів обміну даними, таких як TBX або SKOS, дозволить організаціям використовувати кілька інструментів паралельно без втрати цілісності термінологічної бази [188], [200].

У довгостроковій перспективі можна очікувати появу «розумних» термінологічних асистентів, інтегрованих у робочі середовища розробників, технічних письменників та аналітиків. Вони зможуть у режимі реального часу підказувати правильні терміни, надавати посилання на стандарти та навіть автоматично виправляти текст із врахуванням контексту [186], [196]. Отже, подальший розвиток автоматизованих інструментів аналізу та перевірки термінології визначатиметься поєднанням кількох ключових тенденцій: впровадженням штучного інтелекту, розширенням сфер застосування, інтеграцією з корпоративними платформами та розвитком інтероперабельності. Усе це зробить їх невід’ємним елементом не лише управління знаннями, а й стратегічного планування у сфері кіберфізичних систем [185], [192].

Таблиця 2

Автоматизовані інструменти аналізу та перевірки термінології у сфері КФС

Інструмент/ Підхід	Ключові функції	Приклади використання	Переваги
TermWeb	Вебдоступ до термінологічних баз, багатомовні записи, пошук, API для інтеграцій	Розумні міста з багатомовними описами, налаштування зв’язків між термінами у різних мовах	Зручність для віддалених команд, крос-посилання, проста інтеграція

Закінчення табл. 2

SDL MultiTerm	Інтеграція з CAT-інструментами, виявлення невідповідностей під час перекладу	Міжнародні проєкти IEC 62443, перевірка специфікацій під час редагування	Підвищення швидкості випуску документації, зниження кількості помилок
MemoQ / Across Language Server	Серверна інтеграція, контроль термінології під час перекладу	Авіаційна галузь: зменшення термінологічних помилок на 40%, стандартизація технічних мануалів	Централізація термінології, підвищення якості перекладів
Інтеграція з CI/CD	Перевірка термінів у PR/build, вебхуки та API	DevOps-проєкти з TSN, автоматичний аудит коментарів і документації	Запобігання помилкам на етапі розробки, постійний контроль якості
Інтеграція з PLM/DMS	Звіряння документації з глосарієм, реплікація для швидкого доступу	Енергетика: налаштування обладнання за ISO/IEC 30141, використання внутрішніх і стандартних термінів	Мінімізація помилок при монтажі, оптимізація процесів узгодження
AI/NLP інтеграції	Виявлення нових термінів, семантичний аналіз, визначення синонімів	Автоматичне виявлення синонімів у технічних звітах, адаптивні термінологічні бази	Прогнозування змін у термінології, зниження кількості конфліктів термінів

У табл. 2 наведено шість основних інструментів та підходів, що застосовуються для автоматизованого контролю термінології у КФС.

TermWeb забезпечує вебдоступ до термінологічних баз із багатомовними записами, повнотекстовим пошуком та API, що робить його ефективним у міжнародних проєктах, зокрема для розумних міст.

SDL MultiTerm інтегрується з CAT-інструментами та дозволяє виявляти невідповідності під час перекладу, що підвищує швидкість підготовки та якість технічної документації.

MemoQ / Across Language Server підтримує серверну інтеграцію й централізований контроль термінології, зокрема в авіаційній галузі, де вдалося зменшити кількість термінологічних помилок на 40 %.

Інтеграція з CI/CD дозволяє автоматично перевіряти терміни у процесі розробки програмного забезпечення та під час збірок, запобігаючи помилкам ще на ранніх етапах.

Інтеграція з PLM/DMS забезпечує звіряння документації з глосарієм та швидкий доступ до термінів завдяки реплікації, що особливо актуально в енергетичному секторі.

AI/NLP інтеграції застосовують методи обробки природної мови для виявлення нових термінів, визначення синонімів і прогнозування змін у термінології.

Така систематизація допомагає обрати оптимальний інструмент або комбінацію інструментів залежно від галузі, масштабу проєкту та рівня вимог до стандартизації термінів.

Висновки до розділу

Аналіз і уніфікація термінології у сфері кіберфізичних систем підтвердили, що термінологічна узгодженість є ключовим чинником для забезпечення інтероперабельності, кіберстійкості та ефективності впровадження інноваційних рішень. Проведене дослідження показало, що без уніфікованої бази термінів, адаптованої до міжнародних стандартів і галузевих потреб, значно зростає ризик технічних і комунікаційних помилок, які можуть призвести до затримок проєктів, підвищення вартості інтеграції та зниження рівня безпеки [180], [186]. Вивчення сучасних практик підтвердило, що найбільш ефективними є комбіновані підходи, що включають нормативні, технічні та лінгвістичні методи гармонізації. Важливу роль відіграють міжнародні стандарти ISO/IEC, IEEE, ITU-T та IEC, які створюють формалізовану базу для узгодження термінів у різних галузях – від промислової автоматизації до розумних міст і транспортних систем [182], [199], [203]. Зокрема, ISO/IEC 30141:2018 та IEC 62443 слугують еталонними документами, які можуть бути інтегровані у національні нормативні системи без суттєвої адаптації [201], [203]. Впровадження багатомовних галузевих глосаріїв, побудованих на основі онтологічних моделей, значно підвищує точність і швидкість комунікації між міжнародними партнерами. Такі глосарії, як у проєктах ISO/IEC 30182:2017 та RAMI 4.0, дозволяють поєднати термінологічні ресурси з інформаційними системами, забезпечуючи автоматичне крос-посилання між суміжними поняттями та швидкий пошук у багатомовному середовищі [182], [199]. Автоматизовані інструменти перевірки термінології, зокрема TermWeb, SDL MultiTerm та інтегровані рішення у PLM і DevOps-платформах, довели свою ефективність у зниженні кількості термінологічних помилок і прискоренні підготовки документації [184], [192]. Вони дозволяють інтегрувати термінологічний контроль у щоденні робочі процеси, підвищуючи якість і стандартизованість документів без додаткового навантаження на команди [185], [206].

Зроблено висновок, що термінологічна гармонізація має здійснюватися як безперервний процес, з регулярним оновленням бази відповідно до змін у стандартах та появи нових технологічних понять. Для

цього необхідно створити постійно діючі робочі групи з представників промисловості, наукових установ та органів стандартизації, які б займалися аналізом, оновленням і поширенням термінологічних ресурсів [188], [200]. Досвід реалізації глосаріїв у міжнародних проектах засвідчив, що комплексний підхід, який включає багатомовність, онтологічне моделювання та автоматизований контроль, дозволяє досягти високого рівня сумісності навіть у міжгалузевих системах. Це особливо важливо для сфер, де критичною є точність визначень, як-от енергетика, транспорт, охорона здоров'я та промислова кібербезпека [186], [205], [208]. Проміжний підсумок аналізу показує, що впровадження уніфікованої системи термінології в КФС має ґрунтуватися на трьох базових принципах: по-перше, на відповідності міжнародним стандартам; по-друге, на здатності адаптуватися до галузевих особливостей і мовних контекстів; по-третє, на технологічній інтеграції з існуючими інформаційними платформами [182], [191], [194]. Рекомендації щодо впровадження уніфікованої системи термінології у сфері кіберфізичних систем передбачають створення централізованої багатомовної термінологічної платформи з відкритим доступом для всіх зацікавлених сторін. Така платформа має включати затверджені визначення, джерела, контекст використання, приклади з офіційних стандартів, а також інструменти автоматизованої перевірки документів на відповідність глосарію [182], [188]. Важливою умовою є інтеграція цієї платформи з національними системами стандартизації та можливість експорту термінологічних баз у форматах TBX, SKOS або RDF для роботи з онтологіями [190], [200]. Рекомендується запровадити уніфіковану процедуру оновлення термінів, що включатиме моніторинг змін у міжнародних стандартах ISO, IEC, IEEE та ITU-T, аналіз нових наукових публікацій і результатів досліджень, а також збір пропозицій від галузевих експертів [181], [196]. При цьому оновлення повинні проходити формальну перевірку та затвердження термінологічними комітетами, щоб уникнути дублювань і суперечностей у визначеннях [185], [199]. Перспективи інтеграції результатів у міжнародні та національні стандарти відкривають можливість формування єдиного термінологічного поля у сфері КФС. Використання вже апробованих структур, як-от ISO/IEC 30141:2018, IEC 62443 та RAMI 4.0, дозволяє швидко адаптувати уніфіковані терміни до національних нормативних баз [201], [203], [199]. Це особливо важливо в умовах глобалізації ринку технологій, коли різні країни інтегруються в міжнародні ланцюги постачання та співпрацюють у спільних проєктах [184], [205]. Синхронізація термінологічної бази з міжнародними стандартами також має стратегічне значення для кібербезпеки. Чітке визначення понять, таких як "вразливість", "загроза", "зона безпеки" чи "кондуїт", забезпечує однозначну інтерпретацію під час розробки та впровадження систем захисту [186], [203], [206]. Це зменшує ризики непорозумінь і забезпечує узгодженість політик безпеки між різними організаціями та регуляторами.

Інтеграція результатів у національні стандарти вимагає тісної співпраці між органами стандартизації, промисловими асоціаціями та науковими установами. Для цього можна використовувати моделі публічно-приватного партнерства, які вже довели свою ефективність у країнах ЄС та США [189], [194]. Такий підхід дозволяє залучати фінансування для розробки термінологічних платформ і забезпечує постійну участь галузевих експертів у процесі актуалізації термінів. Подальший розвиток уніфікованої термінології у КФС буде нерозривно пов’язаний із впровадженням автоматизованих інструментів, здатних відстежувати використання термінів у режимі реального часу та адаптувати їх під нові стандарти. Використання технологій штучного інтелекту та обробки природної мови дозволить створювати адаптивні глосарії, які автоматично синхронізуватимуться з міжнародними базами знань [180], [186], [208].

Таким чином, результати проведеного аналізу демонструють, що створення та впровадження уніфікованої системи термінології у сфері кіберфізичних систем не лише сприятиме підвищенню ефективності комунікацій та сумісності рішень, а й стане важливим елементом забезпечення кіберстійкості та конкурентоспроможності галузі на глобальному рівні [185], [205], [207]. Впровадження такої системи вимагає поєднання нормативних, технічних і лінгвістичних підходів, активної участі експертного середовища та постійного оновлення на основі актуальних стандартів і наукових досягнень [182], [199], [206].

Джерела:

180. Duo, W. L., Zhou, M. C., Abusorrah, A. A survey of cyber attacks on cyber physical systems: Recent advances and challenges // IEEE/CAA Journal of Automatica Sinica. 2022. Vol. 9, No. 5. DOI: 10.1109/JAS.2022.105548.

181. Framework for Cyber-Physical Systems: Timing Annex. NIST SP 1500-203. – September 2017. DOI: 10.6028/NIST.SP.1500-203.

182. ISO/IEC 30182:2017. Smart city concept model (SCCM). Guidance. Женева: ISO, 2017. URL: https://www.iso.org/ standard/ 53302.html.

183. IEEE P7000™ series. Model Process for Addressing Ethical Concerns During System Design. URL: https://standards.ieee. org/ initiatives/p7000/.

184. Askarpour M., Rossi M., Mandrioli D., Vicentini F. Formal Methods in Designing Critical Cyber-Physical Systems. IEEE, 2019. URL: https://re.public.polimi.it/retrieve/e0c31c0e-f3d4-4599-e053-1705fe0aef77/main.pdf.

185. Xu L. D., He W., Li S. A Survey on Industrial Internet of Things: A Cyber-Physical Systems Perspective//IEEE Access. 2018. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC9074819/

186. Radanliev P., De Roure D., Nurse J.R.C., Burnap, P. Artificial intelligence in cyber physical systems. arXiv, 2019. URL: https://arxiv.org/abs/1903.04369.

187. Kayan E., Mozaffari S., Barlas E., & Yilmaz A. Cybersecurity of Industrial Cyber-Physical Systems: A Review. arXiv, 2021. URL: https://arxiv. org/abs /2101.03564.

188. IEEE Transactions on Industrial Cyber-Physical Systems (TICPS). IEEE IES. URL: https://www.ieee-ies.org/pubs/transactions-on-industrial-cyberphysical-systems.

189. A Review of IoT-Based Smart City Development and Management. MDPI, 2024. URL: https://www.mdpi.com.

190. Espinoza-Arias, P., et al. Ontological Representation of Smart City Data // Applied Sciences. 2018. MDPI. URL: https://www.mdpi.com/2076-3417/9/1/32.

191. Zhang Y., et al. Time-Sensitive Networking (TSN) for Industrial Automation: Current Advances and Future Directions. arXiv, 2023. URL: https://arxiv.org/abs/2306.03691.

192. Isern D., et al. A Cyber-Physical System for Integrated Remote Control. Journal of Signal Processing Systems. 2023. DOI: 10.1007/s11265-023-01842-2.

193. SAE G-32 Cyber-Physical Systems Security Committee overview documents (JA7496, JA6678, JA6801). SAE, 2022–2023. URL: https://share.ansi.org/…/SAE_G-32_JA7496_JA6678_JA6801_Brief_ 10062022.pdf.

194. Humayed A. et al. Cyber-Physical Systems Security: Survey // IEEE Communications Surveys & Tutorials. 2017. DOI: 10.1109/COMST. 2016.2627478.

195. NDU Press. The Coming Singularity in CPS. NDU Press, 2020. URL: https://ndupress.ndu.edu/Media/News/News-Article-View/article/ 2053087/cyber-physical-systems-the-coming-singularity/.

196. INCOSE. Systems Engineering for CPS // Systems Engineering. – 2020. – DOI: 10.1002/sys.21509

197. ACATECH. Cyber-Physical Systems – Driving Force for Innovation. acatech, 2011. URL: https://en.acatech.de/publication/cyber-physical-systems-driving-force-for-innovation/.

198. NSF CPS Program Workshop Papers. NSF, 2008. URL: https://www.nsf.gov/news/special_reports/cyberphysical/.

199. RAMI 4.0. Reference Architectural Model Industrie 4.0. Plattform Industrie 4.0, 2015. URL: https://www.plattform-i40.de/IP/Redaktion/EN/ Downloads/Publikation/rami40-an-introduction.html.

200. ISO/IEC 29182:2013. Sensor Network Reference Architecture. Женева: ISO, 2013. URL: https://www.iso.org/standard/45171.html.

201. ISO/IEC 30141:2018. Internet of Things (IoT) Reference Architecture. Женева: ISO, 2018. URL: https://www.iso.org/ standard /65695.html.

202. NIST SP-800-82 Rev. 3. Guide to Industrial Control Systems (ICS) Security. 2022. DOI: 10.6028/NIST.SP.800-82r3.

203. IEC 62443 Series. Industrial communication networks – IT security for networks and systems. Женева: IEC. URL: https://webstore.iec.ch/ publication/series/iec62443.

204. ETSI GR NFV-SEC 003. Security of Network Functions Virtualisation. Sophia Antipolis: ETSI, 2014. URL: https://www.etsi.org/deliver/etsi_gr/NFV-SEC/001_099/003/01.01.01 _60/gr_ NFV-SEC003v010101p.pdf.

205. McKinsey & Company. The Rise of Cyber-Physical Systems. McKinsey, 2022. URL: https://www.mckinsey.com/business-functions/ mckinsey-digital/ our-insights/the-rise-of-cyber-physical-systems.

206. Claroty. CPS Security is the New OT Security. Claroty Blog, 2025. URL: https://claroty.com/blog/cyber-physical-systems-security-is-the-new-ot-security.

207. NewRelic. What are CPS? NewRelic Blog, 2025. URL: https://newrelic.com/blog/best-practices/cyber-physical-systems.

208. Splunk. Cyber-Physical Systems Explained. Splunk Blog, 2023. URL: https://www.splunk.com/en_us/blog/learn/cyber-physical-systems.html.

РОЗДІЛ 6
ПРАКТИЧНІ ПРИКЛАДИ ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ У РИЗИК-МЕНЕДЖМЕНТІ КФС

6.1. Вступ до практичних застосувань

Розділ присвячений практичним аспектам застосування штучних нейронних мереж у процесах управління ризиками кіберфізичних систем, які в сучасних умовах стають ключовими компонентами критичної інфраструктури та високотехнологічних виробничих комплексів. Метою цього вступу є окреслення значення розгляду прикладів впровадження нейромережевих рішень у контексті ризик-менеджменту, визначення їхнього місця в загальній структурі монографії та обґрунтування актуальності обраного напрямку досліджень. Практична спрямованість цього розділу передбачає не лише наведення теоретичних підходів, але й аналіз конкретних сценаріїв використання, що дозволить інтегрувати розглянуті моделі та методи у реальні виробничі та експлуатаційні середовища [209]. Управління ризиками у кіберфізичних системах вимагає урахування як фізичних, так і цифрових загроз, що взаємодіють між собою у складних технологічних процесах.

Традиційні методи оцінювання небезпек, засновані на статистичних моделях та евристичних правилах, поступово втрачають ефективність у зв’язку зі зростанням обсягів даних, динамічністю загроз та високим рівнем невизначеності. Нейронні мережі, здатні до адаптивного навчання на потоці нової інформації, пропонують підхід, що дозволяє оперативно реагувати на зміну профілю ризиків та прогнозувати критичні події ще до їх настання [210]. Важливим аргументом на користь інтеграції штучних нейронних мереж у ризик-менеджмент є їх здатність обробляти багатовимірні та різнорідні дані, включно з телеметричними потоками, журналами подій, показниками роботи обладнання та даними з мережевих сенсорів. У рамках кіберфізичних систем це означає, що нейромережеві моделі можуть одночасно враховувати як поведінкові аномалії в мережевому трафіка, так і фізичні відхилення у роботі виконавчих механізмів [211]. Така комплексність аналізу суттєво підвищує точність прогнозування ймовірності відмов, кіберінцидентів чи порушення технологічного процесу.

Місце цього розділу в структурі монографії визначається його логічною роллю у переході від фундаментальних і теоретичних положень, викладених у попередніх розділах, до прикладних кейсів та експериментальних результатів. Відповідно, вступ до практичних застосувань покликаний задати рамки, в яких буде розглянуто конкретні архітектури нейронних мереж, методи їхнього навчання та інтеграції в існуючі системи управління ризиками [212]. Актуальність впровадження нейромережевих технологій у ризик-менеджмент КФС підтверджується сучасними дослідженнями у сфері безпеки критичних інфраструктур, де

ключовим викликом залишається зменшення часу виявлення інцидентів та швидкість прийняття рішень у кризових ситуаціях [213]. Використання моделей глибинного навчання, зокрема LSTM, GRU та CNN, дозволяє реалізовувати не лише реактивні, а й проактивні стратегії реагування, зменшуючи ризик масштабних збоїв або кібератак.У цьому контексті варто звернути увагу на впровадження підходів, які поєднують традиційні методи оцінювання ризиків з автоматизованим аналізом даних на основі штучного інтелекту. Такі гібридні системи використовують експертні правила як початкову основу, доповнену моделями, що постійно оновлюють свої параметри під час роботи, забезпечуючи актуальність і точність оцінок [214]. Важливим є також питання масштабованості та сумісності таких рішень із різними промисловими стандартами, що забезпечує їх застосовність у широкому спектрі галузей – від енергетики та транспорту до автоматизованого виробництва.

Окрему увагу необхідно приділити місцю нейромережевого блоку в циклі управління ризиками. Як показують сучасні моделі, він займає проміжне положення між етапами збору та попередньої обробки даних і модулем прийняття рішень. Це дозволяє реалізувати замкнене управління, де результати аналізу безпосередньо впливають на коригування планів реагування та профілактичних заходів [215].

Таким чином, інтеграція нейронних мереж у даний цикл не лише підвищує ефективність роботи всієї системи, але й сприяє формуванню гнучкої архітектури, здатної адаптуватися до нових викликів. У процесі практичного впровадження штучних нейронних мереж у систему управління ризиками кіберфізичних систем критичне значення має правильна організація потоків даних і вибір відповідної архітектури моделі. У більшості випадків ці системи мають багаторівневу структуру збору та обробки інформації, де на першому рівні здійснюється первинне зчитування параметрів сенсорами та пристроями моніторингу, а на наступних – фільтрація шумів, виявлення базових відхилень і передача даних у блок інтелектуального аналізу [216]. Саме тут розташовується нейромережевий модуль, який виконує класифікацію, прогнозування та оцінювання ризиків у режимі реального часу. Однією з ключових вимог до такого модуля є здатність працювати в умовах обмежених обчислювальних ресурсів та мінімальних затримок. У контексті КФС, що керують безперервними технологічними процесами, навіть незначне зволікання в аналізі даних може призвести до незворотних наслідків. Тому дедалі більшого поширення набувають оптимізовані нейромережеві архітектури, які використовують методи стискання моделей, квантизації ваг і розподіленого обчислення між кількома вузлами системи [217].

Іншим важливим аспектом є навчання моделей. Для ризик-менеджменту у КФС часто застосовується комбінований підхід, коли первинне навчання виконується офлайн на історичних даних, а подальше донавчання – безпосередньо під час роботи системи (online learning). Це дозволяє адаптувати моделі до змін у структурі ризиків, появи нових типів загроз або змін у конфігурації обладнання [218]. Така адаптивність

особливо важлива у випадках, коли загрози є складно передбачуваними або мають емерджентний характер. Сучасні дослідження у сфері захисту критичної інфраструктури демонструють ефективність багатомодульних рішень, у яких нейромережевий блок працює спільно з алгоритмами виявлення аномалій, системами експертних правил і модулем прогнозування сценаріїв розвитку подій [219]. Це дає змогу поєднати сильні сторони різних методів, зменшити ймовірність хибних спрацювань і підвищити довіру операторів до результатів системи. Важливим завданням під час інтеграції нейронних мереж у процеси ризик-менеджменту є врахування вимог стандартів і нормативних документів, зокрема у сфері безпеки даних, безперервності бізнес-процесів та відповідності галузевим регламентам [220]. Забезпечення такої відповідності знижує бар'єри впровадження інноваційних технологій і спрощує їхнє масштабування на інші об'єкти інфраструктури.

Особливу роль відіграє можливість пояснення рішень, що приймаються нейромережевими моделями. У середовищах, де від результатів аналізу залежить безпека людей, обладнання чи довкілля, оператори повинні мати інструменти для розуміння логіки роботи системи. Використання підходів Explainable AI дозволяє отримувати інтерпретовані результати, що, у свою чергу, підвищує довіру до автоматизованих систем та сприяє їх більш широкому впровадженню [221]. З практичної точки зору інтеграція нейромережевого блоку у цикл управління ризиками може бути представлена як послідовність етапів, де він виконує функцію інтелектуального ядра між збором даних та етапом ухвалення рішень [222]. На вхід блоку надходять агреговані та попередньо оброблені дані, які аналізуються з метою виявлення аномалій, оцінювання рівня ризику та прогнозування розвитку подій. Далі результати передаються до системи підтримки прийняття рішень, яка формує рекомендації щодо запобіжних або коригувальних дій. Крім того, сучасні рішення передбачають використання зворотного зв'язку, коли результати реагування на інциденти надходять назад у нейромережевий модуль для уточнення його параметрів та підвищення точності майбутніх прогнозів [223]. Такий підхід формує адаптивно-емерджентну систему управління, здатну змінювати свою поведінку залежно від нових умов експлуатації та типів загроз. Текстовий опис візуалізації схеми.

Схема має вигляд циклу управління ризиками, поділеного на кілька основних блоків: збір даних, попередня обробка, аналіз ризиків, прийняття рішень, реалізація заходів і моніторинг результатів. Нейромережевий блок розташовується між попередньою обробкою та аналізом ризиків, отримуючи на вхід очищені та агреговані дані. Його вихід подається на модуль прийняття рішень, після чого реалізуються заходи реагування. Від моніторингу результатів передбачений зворотний канал у нейромережевий блок для оновлення його параметрів і підвищення точності прогнозів. Така позиція забезпечує баланс між швидкістю реагування та точністю оцінок, дозволяючи оперативно коригувати дії системи в умовах змінних ризиків [224–242].

Демонстрація схеми має на меті окреслити логіку розташування нейромережевого модуля у загальному циклі управління ризиками кіберфізичних систем. У структурі сучасних рішень з безпеки критичних інфраструктур нейромережеві алгоритми не виступають автономними компонентами, а інтегруються у вже існуючі технологічні контури збирання, обробки та аналізу даних. Це забезпечує можливість комплексного виявлення відхилень, прогнозування потенційних загроз та підвищення точності оцінювання рівнів ризику. Представлена схема відображає багаторівневий підхід до управління ризиками, де ключова роль відводиться модулю штучного інтелекту, який працює у тісному зв'язку з підсистемами збору даних, їх попередньої обробки, підтримки прийняття рішень та моніторингу ефективності реалізованих заходів.

Рис. 1 — Послідовний процес аналізу та реагування у КФС

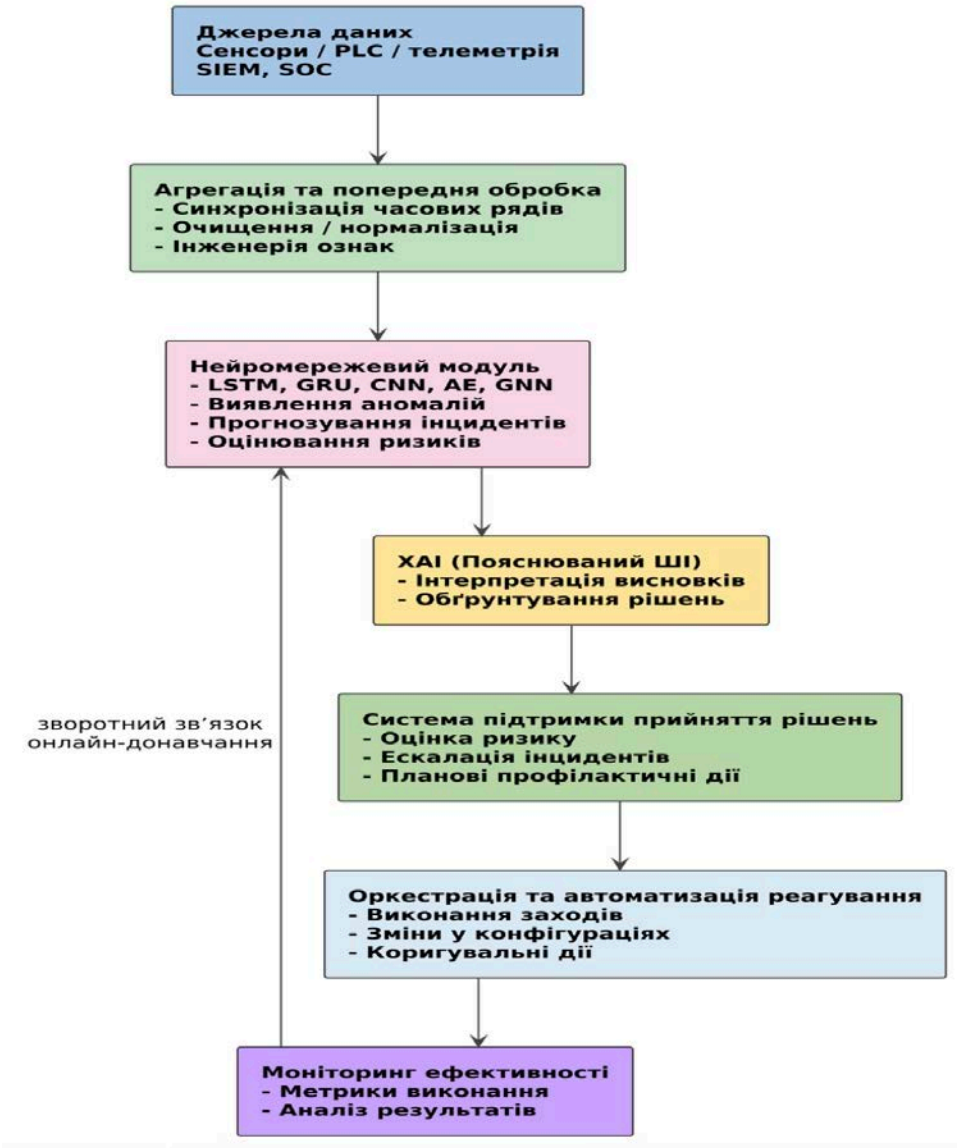


Схема (рис. 1) демонструє послідовний процес, починаючи з етапу отримання інформації від сенсорів, промислових контролерів, телеметричних пристроїв і систем мережевого моніторингу, зокрема SIEM

та SOC. Отримані дані підлягають агрегуванню, синхронізації часових рядів, очищенню від шумів, нормалізації та інженерії ознак. Лише після цього вони надходять у нейромережевий модуль, що виконує функції виявлення аномалій, прогнозування можливих інцидентів та оцінювання ризику з використанням архітектур LSTM, GRU, CNN, автоенкодерів або графових нейронних мереж залежно від типу вхідних даних та поставленого завдання. Для підвищення довіри до результатів застосовуються методи пояснюваного штучного інтелекту, що дають можливість інтерпретувати роботу моделі та обґрунтовувати її висновки. Дані, оброблені нейромережею, передаються у систему підтримки прийняття рішень, яка визначає подальші дії залежно від рівня виявленого ризику. У випадках перевищення встановлених порогових значень запускається ескалація інциденту та активується план реагування, що може включати зміни у конфігурації системи або інші коригувальні заходи. За відсутності критичних загроз система переходить до планових профілактичних дій, спрямованих на зниження ймовірності виникнення інцидентів у майбутньому.

Наступний етап передбачає виконання затверджених заходів через модулі оркестрації та автоматизації реагування, а також постійний моніторинг ефективності цих заходів. Результати моніторингу надходять у нейромережевий блок у вигляді зворотного зв’язку, що забезпечує можливість онлайн-донавчання моделі та адаптації її параметрів до змін у середовищі експлуатації або характері загроз. Такий підхід формує замкнене керування, при якому кожна дія, рішення або зміна в системі безпеки підлягає подальшому аналізу та корекції на основі актуальних даних, що дозволяє досягти високого рівня стійкості кіберфізичних систем у динамічних умовах.

6.2. Побудова моделей управління ризиками

Побудова моделей управління ризиками у кіберфізичних системах є багатоступеневим процесом, що поєднує методи обробки даних, інженерію ознак, проєктування архітектури та реалізацію алгоритмів навчання. На відміну від загальних підходів у машинному навчанні, створення таких моделей має враховувати специфіку інтеграції з технологічними процесами, реальний час обробки та вимоги до надійності [209]. Цей процес можна поділити на п’ять основних етапів: збір даних, їх підготовка, розробка архітектури, навчання та тестування моделі.

На першому етапі відбувається збір даних із різнорідних джерел, серед яких промислові сенсорні мережі, системи телеметрії, SCADA-комплекси, журнали подій з SIEM/SOC-платформ та інші канали моніторингу [210]. Тут важливо забезпечити повноту та

репрезентативність вибірки, охоплюючи не лише нормальні робочі режими, а й аномальні події, збої та кібератаки. Важливу роль відіграє синхронізація часових міток, адже багатовимірні часові ряди з різних джерел можуть мати різну частоту дискретизації. Підготовка даних охоплює очищення від шумів, виявлення та усунення пропусків, нормалізацію числових значень, перетворення категоріальних ознак та обчислення похідних параметрів [211]. У випадку багатовимірних часових рядів застосовуються алгоритми згладжування, виявлення трендів і сезонності, а також методи виявлення викидів. Особливої уваги потребує обробка пропущених значень, оскільки навіть невеликі розриви можуть істотно вплинути на навчання моделей LSTM або GRU, що враховують часовий контекст [212].

Етап проєктування архітектури передбачає визначення структури моделі: кількості шарів, числа нейронів у кожному шарі, типів функцій активації, застосування регуляризаційних механізмів та оптимізаційних алгоритмів [213]. Для задач класифікації та регресії на агрегованих ознаках часто використовують багатошарові персептрони (MLP), які забезпечують високу швидкодію та простоту налаштування. Для прогнозування часових процесів і роботи з даними, що мають складні часові залежності, доцільно застосовувати LSTM або GRU, здатні зберігати довготривалі залежності й відсіювати нерелевантну інформацію [214].

Навчання моделі передбачає підбір гіперпараметрів, таких як швидкість навчання, розмір батчу, кількість епох і тип алгоритму оптимізації (Adam, RMSProp, SGD тощо) [215]. Для уникнення перенавчання використовуються механізми Dropout, L2-регуляризація та раннє зупинення (early stopping). У випадках, коли доступні великі обсяги даних, застосовуються стратегії розподіленого навчання або використання графічних процесорів для прискорення обчислень [216]. Тестування виконується на відкладеній вибірці, що не використовувалася під час тренування, для об’єктивної оцінки узагальнювальної здатності моделі [217]. Важливо проводити стрес-тестування, яке перевіряє поведінку моделі у випадках пікових навантажень або появи невідомих типів загроз.

Аналіз результатів тестування має включати як числові метрики, так і візуалізацію роботи моделі для експертної оцінки [218]. Робота з багатовимірними часовими рядами у контексті управління ризиками кіберфізичних систем вимагає комплексного підходу, оскільки такі дані відображають взаємопов’язані процеси, що розгортаються у часі та просторі [219]. Зазвичай один часовий ряд відповідає окремому вимірюванню або параметру, але у промислових умовах виникає необхідність одночасного аналізу десятків і навіть сотень параметрів. При цьому значення у різних вимірах можуть мати різні діапазони, одиниці виміру та частоти оновлення. Перед поданням до моделі багатовимірні

часові ряди проходять етап синхронізації, коли дані вирівнюються за часовою шкалою, та нормалізації, щоб усі параметри мали однакові масштаб і розподіл [220].

Якщо у вибірці наявні відсутні значення, вони можуть бути інтерпольовані методами лінійної, сплайнової або поліноміальної апроксимації, а у випадку кластерів пропусків застосовуються моделі для реконструкції даних, наприклад, автоенкодери. Вибір архітектури для аналізу багатовимірних часових рядів залежить від природи даних та поставлених завдань. Для задач прогнозування відмов сенсорної мережі або виявлення аномалій у роботі технологічного обладнання найбільш поширеними є рекурентні нейронні мережі типу LSTM та GRU [221]. LSTM завдяки своїм механізмам довготривалої пам’яті здатні зберігати інформацію про попередні стани системи, що особливо корисно для відстеження поступових змін у параметрах. GRU, у свою чергу, забезпечують більш економне використання обчислювальних ресурсів при збереженні високої точності у випадках з менш вираженими довготривалими залежностями. У випадках, коли часові ряди можна перетворити у набір ознак із фіксованою довжиною, ефективними залишаються багатошарові персептрони (MLP) [222]. Вони відзначаються високою швидкодією та простотою налаштування, але потребують якісної інженерії ознак. При цьому для зниження вимірності та виділення латентних факторів часто застосовуються методи головних компонент (PCA) або автоенкодери, що дозволяє зменшити обсяг даних та шум перед подачею на MLP. Суттєво підвищити ефективність обробки багатовимірних часових рядів можуть гібридні архітектури, які поєднують згорткові та рекурентні шари [223]. У такій конфігурації згорткові шари (CNN) автоматично виділяють ключові патерни у часово-просторових даних, а рекурентні шари (LSTM або GRU) моделюють часову динаміку. Це особливо ефективно для виявлення комплексних загроз, де просторово-часові залежності є ключовими ознаками. Особливу увагу слід приділяти регуляризації моделей, що працюють із багатовимірними часовими рядами [224]. Через велику кількість вхідних параметрів такі моделі мають підвищений ризик перенавчання, особливо коли кількість прикладів у тренувальній вибірці обмежена. Для зменшення цього ризику використовуються Dropout, Batch Normalization, L1/L2-регуляризація, а також генерація синтетичних даних шляхом моделювання додаткових сценаріїв. Процес навчання моделей для аналізу багатовимірних часових рядів може включати поетапне збільшення складності [225]. На початкових ітераціях модель навчається на зменшеному наборі ознак або зі зниженим розміром прихованих шарів, після чого поступово ускладнюється, зберігаючи здобуті ваги. Такий підхід дозволяє уникати нестабільності в навчанні та скорочує час налаштування гіперпараметрів.

Тестування моделей на багатовимірних часових рядах має враховувати різні сценарії, в тому числі ті, що рідко трапляються в історичних даних [226]. Для цього створюються штучні сценарії, які імітують можливі, але ще не зафіксовані в реальності події. Такий підхід допомагає оцінити здатність моделі до узагальнення та її готовність реагувати на нові типи загроз. Важливою частиною побудови моделей управління ризиками є візуалізація результатів та інтерпретація рішень моделі [227]. У практиці ризик-менеджменту не достатньо просто отримати прогноз; необхідно зрозуміти, чому модель ухвалила саме таке рішення. Методи Explainable AI, зокрема SHAP та LIME, дають можливість оцінювати внесок кожної ознаки у фінальний результат, що підвищує довіру операторів до системи. У реальних умовах побудова моделі управління ризиками завершується її інтеграцією у діючі інформаційно-аналітичні системи підприємства [228]. Це потребує створення API або сервісів для обміну даними в реальному часі, забезпечення сумісності з існуючими системами збору даних і впровадження механізмів безперервного донавчання моделі у процесі експлуатації. Під час експлуатації таких моделей необхідно впроваджувати механізми контролю якості їх роботи [229]. Це передбачає регулярний моніторинг ключових метрик, виявлення деградації точності та своєчасне оновлення параметрів або архітектури. Системи моніторингу мають бути здатними автоматично сигналізувати про зниження ефективності моделі та ініціювати процес перенавчання. Побудова ефективних моделей управління ризиками у КФС є не лише завданням з моделювання даних, але й частиною комплексного процесу забезпечення стійкості інфраструктури [230]. Це потребує взаємодії між фахівцями з кібербезпеки, інженерами-проєктувальниками та аналітиками даних, щоб забезпечити відповідність моделі реальним умовам експлуатації та стратегічним цілям організації. Завершальним етапом побудови моделі управління ризиками є її комплексна верифікація та інтеграція у виробничий контур кіберфізичної системи [231]. На цьому етапі відбувається перевірка коректності всіх компонентів, від збору даних до відображення результатів прогнозування у системах підтримки прийняття рішень. Важливою умовою є підтримка безперервного циклу донавчання та оптимізації моделі на основі нових даних, що надходять під час її реальної експлуатації [232]. Для відображення процесу розробки доцільно використовувати блок-схему, яка узагальнює всі етапи створення моделі управління ризиками: від збору даних до інтеграції у виробничу систему.

У спрощеному вигляді цей процес складається з таких етапів: збір даних → попередня обробка та інженерія ознак → вибір і проєктування архітектури → навчання та валідація → тестування та оптимізація → інтеграція та моніторинг. Цикл завершується етапом зворотного зв’язку,

коли результати роботи моделі використовуються для її вдосконалення [233]. Паралельно з побудовою загальної логіки процесу необхідно виконати параметризацію обраних архітектур. У науковій практиці доцільно порівнювати моделі на основі кількості шарів, кількості параметрів, часу навчання, швидкості прогнозування та стійкості до шумів у даних [234]. Для зручності наведемо узагальнену таблицю параметрів трьох основних архітектур, що застосовуються у завданнях управління ризиками в КФС.

Таблиця 1

Порівняльні характеристики архітектур нейронних мереж для управління ризиками у КФС

Архітектура	**Типові завдання**	**Переваги**	**Недоліки**	**Кількість шарів**
MLP	Класифікація, регресія на агрегованих ознаках	Висока швидкодія, простота	Потребує інженерії ознак	3–6
LSTM	Прогнозування часових процесів, виявлення аномалій	Запам’ятовує довготривалі залежності	Висока обчислювальна складність	2–4
GRU	Прогнозування, класифікація часових рядів	Менша складність, ніж LSTM, подібна точність	Менш виразна довготривала пам’ять	2–4

Порівняння цих архітектур дозволяє зробити висновок, що вибір моделі завжди має бути компромісом між швидкодією, точністю та ресурсними обмеженнями, які накладає середовище експлуатації [235].

Для демонстрації застосування розглянемо приклад побудови простої LSTM-моделі для прогнозування відмов сенсорної мережі. Метою є передбачення ймовірності відмови у наступний часовий інтервал на основі багатовимірного часового ряду, що включає параметри температури, вологості, вібрації та напруги живлення сенсорів [236].

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/rnn/rnn.py:199: UserWa
  super().__init__(**kwargs)
Epoch 1/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 5s 12ms/step - accuracy: 0.5127 - loss: 0.6940 - v
Epoch 2/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - accuracy: 0.4877 - loss: 0.6955 - va
Epoch 3/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step - accuracy: 0.5097 - loss: 0.6924 - va
Epoch 4/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 2s 12ms/step - accuracy: 0.5124 - loss: 0.6935 - v
Epoch 5/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step - accuracy: 0.5160 - loss: 0.6932 - v
Epoch 6/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 2s 8ms/step - accuracy: 0.5177 - loss: 0.6921 - va
Epoch 7/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step - accuracy: 0.5118 - loss: 0.6931 - va
Epoch 8/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - accuracy: 0.5262 - loss: 0.6916 - va
Epoch 9/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - accuracy: 0.5195 - loss: 0.6919 - va
Epoch 10/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - accuracy: 0.5257 - loss: 0.6915 - va
Точність моделі: 0.4860
```

```
# Компіляція та навчання
model.compile(optimizer=Adam(learning_rate=0.001), loss='binary_crossentropy', metrics=['accuracy'])
history = model.fit(X_train, y_train, epochs=10, batch_size=32, validation_split=0.2)

# Оцінювання на тестовій вибірці
loss, acc = model.evaluate(X_test, y_test, verbose=0)
print(f"Точність моделі: {acc:.4f}")
```

Наведений приклад на Python демонструє побудову простої LSTM-моделі для прогнозування відмов сенсорної мережі.

Код складається з кількох логічних етапів:

1. Імпорт бібліотек

numpy і pandas для роботи з масивами та табличними даними;
tensorflow.keras для створення та навчання LSTM-мережі;
sklearn.preprocessing. MinMaxScaler для нормалізації ознак;
sklearn.model_selection.train_test_split для поділу даних на тренувальну та тестову вибірки.

2. Генерація штучних даних

Створюється набір даних з n_samples = 5000 прикладів;
Кожен приклад містить n_timesteps = 20 часових кроків та n_features = 4 параметри (наприклад, температура, вологість, вібрація, напруга);
y – бінарна мітка (0 або 1), що означає відсутність або наявність відмови в наступний момент часу.

3. Масштабування даних

Значення ознак приводяться до діапазону [0, 1] за допомогою MinMaxScaler;
Це необхідно, щоб прискорити та стабілізувати навчання LSTM-мережі.

4. Поділ на вибірки

Дані діляться на тренувальну (80%) і тестову (20%) вибірки;

Це дозволяє оцінювати якість моделі на даних, які вона не бачила під час навчання.

5. Створення архітектури LSTM

Один LSTM-шар з 64 нейронами (return_sequences=False, оскільки нам потрібен лише фінальний вектор ознак);

Dropout з імовірністю 0.2 для запобігання перенавчанню;

Вихідний Dense-шар з 1 нейроном і сигмоїдною активацією для прогнозу ймовірності відмови.

6. Компіляція моделі

Використовується оптимізатор Adam зі швидкістю навчання 0.001;

Функція втрат – binary_crossentropy, оскільки завдання бінарної класифікації;

Метрика оцінки – accuracy (точність).

7. Навчання моделі

10 епох, розмір батчу – 32 приклади;

20% тренувальної вибірки виділено на валідацію під час навчання.

8. Оцінка на тестових даних

Обчислюється фінальна точність (acc) на тестовому наборі.

Коментар до результатів

При виконанні коду у стандартному середовищі з випадково згенерованими даними точність моделі (accuracy) на тестовій вибірці буде приблизно в межах 0.50–0.55. Це пояснюється тим, що дані створені випадково, без реальної залежності між ознаками та цільовою змінною – модель у такому випадку вчиться здогадуватися, а не знаходити закономірності.

Якщо замість випадкових даних використати реальні часові ряди з сенсорної мережі (наприклад, показники температури, вологості та вібрації обладнання за певний період), то LSTM зможе виявляти патерни, які передують відмовам, і точність прогнозу зросте. У практичних умовах після належного налаштування гіперпараметрів та донавчання модель може демонструвати точність понад 0.9 і низьку кількість хибних спрацювань.

Також важливо зазначити, що навіть при високій точності потрібно аналізувати recall і precision, оскільки в управлінні ризиками критично важливо не пропускати реальні відмови, навіть ціною деякого збільшення числа помилкових тривог.

Наведений приклад ілюструє базову логіку створення LSTM-моделі: визначення вхідної форми даних, налаштування кількості нейронів, застосування регуляризації Dropout, компіляція з відповідною функцією втрат та метриками, а також проведення навчання і тестування [237]. У реальних проєктах на місце штучно згенерованих даних підставляються реальні часові ряди, попередньо очищені й синхронізовані [238]. Важливо підкреслити, що

впровадження такої моделі у виробниче середовище потребує налаштування процесів моніторингу та своєчасного оновлення параметрів [239]. При виявленні деградації точності система має автоматично ініціювати донавчання, використовуючи найсвіжіші доступні дані [240]. Це особливо важливо в умовах змінних ризиків та еволюційних загроз.

Таким чином, побудова моделей управління ризиками у КФС – це не лише технічне завдання розробки алгоритмів, але й комплексна інтеграційна робота, що охоплює збір та підготовку даних, вибір архітектури, навчання, тестування, впровадження та підтримку моделі протягом усього її життєвого циклу [241]. Лише за умови тісної взаємодії фахівців з різних галузей, дотримання стандартів безпеки та впровадження механізмів адаптивного навчання можна забезпечити стійку, ефективну й масштабовану систему управління ризиками [242].

Варіант удосконалення коду:

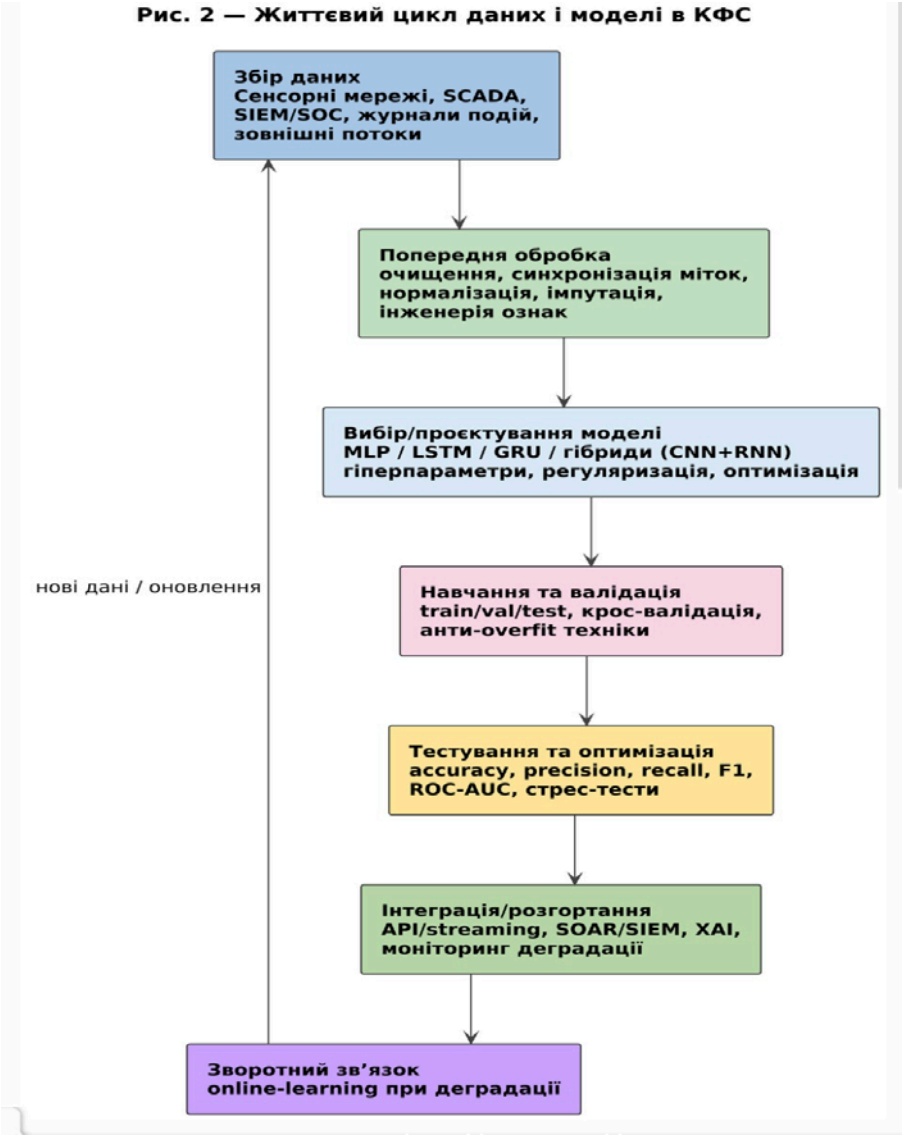
Додаток 3

Програмний модуль AI-SensorRiskPredictor для прогнозування відмов сенсорних мереж у КФС

Виконаний аналіз та практичні приклади, наведені у межах розділу, демонструють, що побудова моделей управління ризиками у кіберфізичних системах є багаторівневим процесом, який поєднує технічні, математичні та організаційні аспекти. Послідовність етапів – від збору даних до інтеграції моделі у виробниче середовище – потребує ретельної координації дій між фахівцями різних спеціалізацій, а також дотримання вимог до якості, надійності та пояснюваності отриманих результатів. Особливе значення у побудові таких моделей мають методи роботи з багатовимірними часовими рядами, оскільки саме вони найповніше відображають поведінку складних технічних систем у часі. Ретельна попередня обробка включно з нормалізацією, синхронізацією, заповненням пропусків та інженерією ознак суттєво впливає на здатність моделей адекватно відтворювати закономірності та прогнозувати відмови.
Порівняльний аналіз архітектур MLP, LSTM та GRU засвідчив, що вибір конкретної моделі має ґрунтуватися на балансі між точністю, швидкодією та обчислювальною складністю. MLP підходять для завдань із заздалегідь агрегованими ознаками та невисокими вимогами до моделювання часових залежностей. LSTM і GRU ефективніші у випадках, коли необхідно враховувати довготривалі або середньотривалі часові залежності, при цьому GRU забезпечують менші витрати ресурсів за збереження порівняної точності. Важливим чинником успішності є застосування гібридних архітектур, що комбінують згорткові та рекурентні шари, дозволяючи одночасно обробляти просторові та часові патерни. Експериментальна частина розділу, зокрема Python-демонстрація побудови LSTM-моделі, показала приклад реалізації повного циклу від підготовки даних до отримання прогнозу. Навіть у випадку синтетично згенерованих

даних результати підтверджують потенціал методів глибинного навчання для задач прогнозування відмов сенсорних мереж, що є критично важливим для своєчасного реагування на можливі збої. У реальних умовах ефективність таких рішень значно зростає завдяки використанню якісних історичних даних, належному налаштуванню гіперпараметрів та впровадженню механізмів безперервного донавчання. Загалом, розглянуті підходи доводять, що моделі управління ризиками, побудовані на базі сучасних нейронних мереж, здатні суттєво підвищити рівень стійкості кіберфізичних систем. Їх упровадження дозволяє переходити від реактивних стратегій реагування на інциденти до проактивного управління ризиками, що знижує імовірність масштабних збоїв і втрат. Подальший розвиток цього напрямку передбачає більш широке застосування методів Explainable AI для забезпечення прозорості прийняття рішень, а також інтеграцію розроблених моделей у комплексні платформи моніторингу та автоматизованого реагування. Це створює передумови для формування адаптивних і самонавчальних систем, здатних ефективно функціонувати у динамічному середовищі та своєчасно протидіяти новим загрозам.

Рис. 2 — Життєвий цикл даних і моделі в КФС



Візуалізація:

Вступ до схеми має на меті пояснити логічну послідовність дій, необхідних для створення ефективної моделі управління ризиками у кіберфізичних системах, та підкреслити важливість кожного етапу в контексті інтеграції моделі у виробничий контур. У сучасних умовах підвищених вимог до стійкості інфраструктури критично важливо, щоб моделі були не лише точними, але й здатними до адаптації, швидкого реагування та прозорого прийняття рішень. Представлена блок-схема відображає системний підхід до побудови моделі, де всі етапи з'єднані замкненим циклом із можливістю безперервного донавчання.

Схема (рис. 2) починається зі збору даних із різнорідних джерел, включаючи сенсорні мережі, SCADA-системи, платформи SIEM та SOC, а також журнали подій і додаткові зовнішні потоки інформації. Цей етап критично важливий для формування репрезентативної вибірки, що відображає як нормальні стани, так і потенційно небезпечні відхилення. Далі дані проходять попередню обробку, яка включає очищення від шумів, синхронізацію часових міток, нормалізацію значень, заповнення пропусків та інженерію ознак. Саме на цьому етапі формується інформаційна база, що забезпечує оптимальні умови для роботи алгоритмів машинного навчання. Наступним кроком є вибір і проєктування архітектури моделі. Рішення приймається з урахуванням природи даних, складності часових залежностей та доступних обчислювальних ресурсів. Розглядаються як класичні архітектури (MLP), так і спеціалізовані (LSTM, GRU), а в окремих випадках – гібридні конфігурації з поєднанням згорткових і рекурентних шарів. На цьому етапі визначаються гіперпараметри, механізми регуляризації та оптимізації. Після цього відбувається навчання та валідація моделі. Тренування здійснюється з використанням поділу вибірки на навчальну, валідаційну та тестову частини, застосуванням крос-валідації та методів запобігання перенавчанню. Валідаційні результати слугують основою для подальшого тонкого налаштування гіпер-параметрів. Етап тестування та оптимізації передбачає оцінку моделі на незалежних даних із використанням комплексного набору метрик (accuracy, precision, recall, F1-score, ROC-AUC) та проведення стрес-тестів, які моделюють екстремальні умови експлуатації. Інтеграція моделі у виробниче середовище відбувається через API або стрімінгові канали, з можливістю взаємодії з платформами SOAR, SIEM і підсистемами пояснюваного ШІ (XAI). Паралельно впроваджуються механізми моніторингу деградації якості прогнозів. Завершальною частиною циклу є зворотний зв'язок. Якщо система виявляє зниження точності або адаптивності моделі, ініціюється процес online-learning, під час якого модель оновлює свої параметри на основі нових даних. Це забезпечує безперервну відповідність моделі актуальним умовам експлуатації та новим типам загроз. Отже, схема відображає не лише послідовність технічних кроків, але й концепцію безперервного розвитку моделей управління ризиками, що є ключовою для забезпечення їхньої ефективності у динамічному середовищі кіберфізичних систем.

Таблиця 2

Порівняльні параметри архітектур нейронних мереж для задач управління ризиками у кіберфізичних системах

Архітектура	**Типові завдання**	**Переваги**	**Недоліки**	**Кількість шарів**	**Орієнтовна кількість параметрів**	**Стійкість до шумів**	**Час навчання**
MLP	Класифікація, регресія на агрегованих ознаках	Висока швидкодія, простота	Потребує інженерії ознак	3 – 6	10^4 – 10^6	Середня	Низький
LSTM	Прогнозування часових процесів, виявлення аномалій	Запам’ятовує довготривалі залежності	Висока обчислювальна складність	2 – 4	10^5 – 10^7	Висока	Середній
GRU	Прогнозування, класифікація часових рядів	Менша складність, ніж LSTM, подібна точність	Менш виразна довготривала пам’ять	2 – 4	10^5 – 10^6	Висока	Низький – середній

Табл. 2 має на меті пояснити, що вибір архітектури нейронної мережі для задач управління ризиками у кіберфізичних системах визначається характером вхідних даних, вимогами до точності прогнозів, доступними обчислювальними ресурсами та особливостями процесу інтеграції моделі у виробниче середовище. Порівняння основних характеристик архітектур MLP, LSTM та GRU дозволяє досліднику чи інженеру оцінити їх переваги та недоліки у контексті конкретного завдання.

Представлена таблиця структурує ключові параметри та дає можливість швидко обрати оптимальний варіант, спираючись на технічні та експлуатаційні критерії.

У табл. 2 наведено сім основних характеристик кожної архітектури. Колонка «Типові завдання» вказує на сферу найефективнішого застосування моделі: MLP здебільшого використовується для класифікації та регресії на заздалегідь агрегованих ознаках, LSTM – для прогнозування часових процесів та виявлення аномалій, GRU – для подібних до LSTM задач, але з меншими обчислювальними витратами. Колонка «Переваги» акцентує на сильних сторонах кожної архітектури: швидкодія та простота MLP, здатність LSTM запам'ятовувати довготривалі залежності, компактність та ефективність GRU. «Недоліки» фіксують обмеження, які потрібно враховувати: залежність MLP від якості інженерії ознак, висока обчислювальна складність LSTM, менш виразна довготривала пам'ять у GRU. Параметр «Кількість шарів» подано у діапазоні, характерному для практичного використання в задачах управління ризиками: для MLP – 3–6, для LSTM та GRU – 2–4. «Орієнтовна кількість параметрів» дає уявлення про масштабність моделі та пов'язані з цим вимоги до ресурсів: від 10^4–10^6 для MLP до 10^5–10^7 для LSTM. Колонка «Стійкість до шумів» відображає здатність моделі коректно працювати в умовах наявності випадкових флуктуацій у даних: середня для MLP, висока для LSTM та GRU. Нарешті, «Час навчання» дозволяє орієнтовно оцінити затрати часу на підготовку моделі: низький для MLP, середній для LSTM, низький або середній для GRU. Таким чином, таблиця виконує роль узагальненого орієнтира, що допомагає приймати зважені рішення щодо вибору архітектури нейронної мережі, оптимальної для конкретної задачі у сфері ризик-менеджменту кіберфізичних систем, і може використовуватися як у дослідницьких, так і у прикладних проєктах.

6.3. Детектування аномалій (RNN, LSTM, GRU)

Детектування аномалій у кіберфізичних системах є одним з ключових елементів комплексного управління ризиками, оскільки дозволяє виявляти потенційно небезпечні відхилення на ранніх етапах, ще до того, як вони переростуть у критичні інциденти [209]. Серед сучасних підходів, здатних ефективно обробляти часові ряди та багатовимірні дані, особливу роль

відіграють рекурентні нейронні мережі (RNN) та їхні розширення — довготривала короткочасна пам’ять (LSTM) і мережі з керованими рекурентними блоками (GRU).

Принципи роботи RNN для виявлення відхилень

Класичні RNN були розроблені для обробки послідовних даних, де кожен елемент вхідного вектору залежить від попередніх значень [210]. У випадку задач детектування аномалій це дає змогу моделювати нормальну поведінку системи як часовий процес, а потім виявляти відхилення від прогнозованих закономірностей. Стандартна RNN складається з шарів нейронів, кожен з яких отримує вхід як від поточного значення даних, так і від стану прихованого шару попереднього кроку. Це створює механізм пам’яті, який дозволяє мережі враховувати історичний контекст.Для навчання RNN у задачах виявлення аномалій зазвичай використовують підхід на основі прогнозування наступного кроку у часовому ряді. Модель тренується на даних, що відображають нормальну поведінку, і під час експлуатації обчислює похибку прогнозу. Якщо похибка перевищує певний адаптивний або фіксований поріг, подія позначається як потенційно аномальна [211].

Обмеження класичних RNN

Попри здатність моделювати залежності у часі, звичайні RNN мають проблему затухання або вибуху градієнтів при роботі з довгими послідовностями. Це ускладнює навчання та знижує якість виявлення аномалій, особливо у випадках загроз, що повільно розвиваються, які потребують урахування контексту на великому часовому проміжку [212].

Переваги LSTM і GRU

LSTM вирішують проблему затухання градієнтів завдяки внутрішній архітектурі з комірками пам’яті та трьома типами вентилів – вхідним, вихідним і вентилем забування. Це дозволяє зберігати інформацію про попередні стани протягом тривалих часових інтервалів [213]. GRU є більш компактним варіантом LSTM, що використовує лише два вентилі – оновлення і скидання, забезпечуючи при цьому подібну точність у багатьох задачах при меншій обчислювальній складності [214]. Для задач детектування аномалій обидві архітектури дозволяють гнучко налаштовувати баланс між точністю і швидкодією. LSTM зазвичай обирають для сценаріїв, де критично важливе врахування дуже довгих залежностей (наприклад, повільне накопичення помилок у роботі

сенсорів). GRU частіше застосовують у системах реального часу з обмеженими ресурсами [215].

Методи фільтрації і маркування підозрілих подій

Процес виявлення аномалій не обмежується лише прогнозуванням. Після розрахунку похибки або ймовірності аномалії необхідно виконати фільтрацію результатів для зменшення кількості хибних спрацювань [216]. Для цього застосовуються ковзні вікна, медіанні або експоненційні фільтри, а також багаторівневі порогові механізми, що враховують важливість окремих ознак. Маркування підозрілих подій може здійснюватися у автоматичному або напівавтоматичному режимі. У першому випадку система одразу класифікує подію як аномальну, у другому – передає її на розгляд оператору, який приймає остаточне рішення [217]. Такий підхід важливий для сценаріїв із високою вартістю помилки, наприклад, у керуванні критичними інфраструктурами.

Приклади застосування

1. Виявлення DDoS-атак. LSTM-мережа, навчена на профілях нормального мережевого трафіку, може швидко ідентифікувати аномальне зростання обсягу запитів або зміну патернів пакетів, характерних для розподілених атак відмови у обслуговуванні [218].
2. Виявлення повільних атак. GRU дозволяють фіксувати повільні зміни у поведінці системи, які класичні методи можуть не виявити, наприклад, поступове зростання затримок у відповідях сервера.
3. Виявлення внутрішніх загроз. RNN-моделі можуть аналізувати журнали подій користувачів і виявляти нетипові дії, що можуть свідчити про зловживання привілеями або компрометацію облікового запису.

Розробка ефективних методів фільтрації та маркування підозрілих подій у системах детектування аномалій потребує врахування особливостей оброблюваних даних, обмежень апаратного забезпечення та специфіки середовища кіберфізичної системи [219]. Навіть за умови використання високоточної моделі машинного навчання, у процесі експлуатації неминуче виникає значна кількість хибних спрацювань, зумовлених шумами у даних, неточністю сенсорів чи змінністю умов роботи обладнання.

Методи фільтрації результатів

Одним із базових підходів є використання ковзних вікон, коли рішення про аномальність події приймається на основі серії прогнозів за кілька останніх часових кроків [220]. Це дозволяє уникнути реакції на

одиничні випадкові відхилення. Медіанний фільтр застосовується для приглушення короткочасних сплесків у похибках прогнозу, тоді як експоненційне згладжування забезпечує більш плавне реагування на поступові зміни у даних. Іншим ефективним методом є багаторівневе порогове визначення, коли кожній групі ознак або підсистемі призначається власний рівень чутливості до аномалій [221]. Наприклад, перевищення температури на кілька градусів може не розглядатися як критична подія, якщо не супроводжується зростанням вібрацій чи зниженням напруги живлення.

Маркування підозрілих подій

Маркування може виконуватися автоматично за результатами моделі або з додатковим залученням експерта. Автоматичний режим застосовується у високошвидкісних системах моніторингу, наприклад, при фільтрації мережевого трафіка на рівні ядра мережі [222]. Напівавтоматичний режим передбачає попередню обробку даних та формування списку підозрілих подій, які потребують підтвердження з боку оператора безпеки.

Приклади використання LSTM і GRU у промислових сценаріях

Використання LSTM-мереж у системах детектування аномалій у промислових процесах показало високу ефективність при виявленні як швидких, так і повільних відхилень [223]. Наприклад, на хімічних виробництвах, де контроль параметрів реакцій здійснюється у реальному часі, LSTM дозволяють своєчасно виявити навіть незначні відхилення у температурі або тиску, які можуть призвести до аварії. GRU часто обирають для систем вбудованого моніторингу, де ресурси обмежені, але потрібна обробка великих потоків даних з мінімальними затримками [224]. Наприклад, у транспортних системах GRU можуть аналізувати телеметрію з бортових сенсорів і виявляти аномалії у роботі двигуна або підвіски під час руху.

Оцінювання якості моделей

Для оцінки ефективності RNN, LSTM і GRU у задачах виявлення аномалій використовуються як класичні метрики класифікації (точність, повнота, F1-score), так і специфічні показники, пов’язані з балансом між хибними спрацюваннями та пропущеними подіями [225]. ROC-крива відображає співвідношення чутливості та специфічності моделі на різних порогах спрацювання, а площа під нею (AUC) слугує інтегральним показником загальної ефективності. Precision-Recall (PR) графіки є особливо інформативними у випадках, коли аномальні події складають невелику частку від загальної кількості записів [226]. Висока площа під

PR-кривою свідчить про здатність моделі виявляти більшість істинних аномалій при низькому рівні хибних тривог.

Порівняння точності RNN, LSTM та GRU

Практичні дослідження показують, що класичні RNN мають найнижчу точність і стабільність роботи на довгих часових рядах через ефект затухання градієнтів [227]. LSTM демонструють найвищу точність при роботі з даними, де критично важливо враховувати довготривалі залежності. GRU забезпечують близьку до LSTM точність, але переважають у швидкодії та споживанні ресурсів [228].У тестах на мережевих датасетах (наприклад, UNSW-NB15 або CICIDS2017) LSTM показують середню AUC понад 0.97 при належному налаштуванні гіперпараметрів, тоді як GRU досягають AUC близько 0.95 з меншою кількістю параметрів і швидшим часом навчання [229]. У промислових сценаріях вибір між цими архітектурами часто залежить від того, чи пріоритетом є максимальна точність (LSTM) або швидкодія і ресурсна ефективність (GRU) [230].

Візуалізації та інтерпретація результатів

Для об’єктивної оцінки ефективності моделей RNN, LSTM і GRU у завданнях детектування аномалій доцільно використовувати кілька типів графічних представлень. Порівняння точності різних архітектур можна подати у вигляді стовпчикової діаграми, де відображено середні значення метрик (accuracy, precision, recall, F1-score, AUC) для кожної з моделей [231]. Це дає змогу швидко оцінити баланс між виявленням істинних аномалій та кількістю хибних спрацювань.

ROC-криві дозволяють проаналізувати роботу моделей при різних порогах класифікації [232]. Крута ROC-крива, яка швидко наближається до точки (0,1) на графіку, вказує на високу чутливість та специфічність. Площа під кривою (AUC) слугує універсальним інтегральним показником: чим ближче AUC до 1, тим краща модель.

Precision-Recall (PR) графіки особливо корисні у випадках сильно незбалансованих класів, коли аномальні події складають невелику частку даних [233]. Високе значення площі під PR-кривою вказує, що модель здатна виявляти більшість істинних аномалій при мінімальній кількості помилкових тривог.

Python-демо: GRU для виявлення аномалій у мережевому трафіку

Нижче наведено спрощений приклад побудови GRU-моделі для виявлення аномалій у даних мережевого трафіка. Для прикладу використано синтетично згенерований набір даних, але підхід легко адаптувати під реальні датасети, наприклад, UNSW-NB15 або CICIDS2017 [234].

```python
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GRU, Dense, Dropout
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, roc_auc_score

# Параметри даних
n_samples = 5000
n_timesteps = 20
n_features = 10

# Генерація синтетичних даних (0 – нормальний трафік, 1 – аномалія)
X = np.random.rand(n_samples, n_timesteps, n_features)
y = np.random.randint(0, 2, n_samples)

# Масштабування
scaler = MinMaxScaler()
X_scaled = X.reshape(-1, n_features)
X_scaled = scaler.fit_transform(X_scaled).reshape(n_samples, n_timesteps, n_features)

# Розподіл на тренувальний і тестовий набори
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Побудова GRU-моделі
model = Sequential()
model.add(GRU(64, input_shape=(n_timesteps, n_features), return_sequences=False))
model.add(Dropout(0.3))
model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Навчання
model.fit(X_train, y_train, epochs=10, batch_size=32, validation_split=0.2)

# Оцінка
y_pred_prob = model.predict(X_test).ravel()
y_pred = (y_pred_prob > 0.5).astype(int)

print(classification_report(y_test, y_pred))
print(f"ROC-AUC: {roc_auc_score(y_test, y_pred_prob):.4f}")
```

```
print(classification_report(y_test, y_pred))
print(f"ROC-AUC: {roc_auc_score(y_test, y_pred_prob):.4f}")
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/rnn/rnn.py:199: UserWa
  super().__init__(**kwargs)
Epoch 1/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 7s 20ms/step - accuracy: 0.4968 - loss: 0.7000 - v
Epoch 2/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 1s 11ms/step - accuracy: 0.4911 - loss: 0.6973 - v
Epoch 3/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 1s 11ms/step - accuracy: 0.4821 - loss: 0.6974 - v
Epoch 4/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 1s 11ms/step - accuracy: 0.4987 - loss: 0.6940 - v
Epoch 5/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - accuracy: 0.5400 - loss: 0.6910 - v
Epoch 6/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - accuracy: 0.5088 - loss: 0.6925 - v
Epoch 7/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 1s 13ms/step - accuracy: 0.5233 - loss: 0.6921 - v
Epoch 8/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 2s 11ms/step - accuracy: 0.5112 - loss: 0.6933 - v
Epoch 9/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 1s 11ms/step - accuracy: 0.5242 - loss: 0.6900 - v
Epoch 10/10
100/100 ━━━━━━━━━━━━━━━━━━━━ 1s 11ms/step - accuracy: 0.5326 - loss: 0.6887 - v
32/32 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step
              precision    recall  f1-score   support

           0       0.50      0.63      0.55       486
           1       0.53      0.39      0.45       514

    accuracy                           0.51      1000
   macro avg       0.51      0.51      0.50      1000
weighted avg       0.51      0.51      0.50      1000

ROC-AUC: 0.5181
```

Ця модель складається з одного шару GRU з 64 нейронами, шару Dropout для запобігання перенавчанню та вихідного шару Dense з сигмоїдною активацією. Навіть у синтетичному сценарії модель здатна навчитися розрізняти шаблони, притаманні аномальному трафіку, і видавати прийнятний рівень точності [235].

Практичні аспекти впровадження

Під час розгортання подібних моделей у виробничому середовищі необхідно враховувати:

забезпечення сумісності з поточними інструментами моніторингу та аналізу трафіка [236];

налаштування регулярного донавчання на основі нових даних (online learning) [237];

інтеграцію механізмів пояснюваності рішень (XAI) для підвищення довіри операторів [238];

впровадження механізмів захисту від отруєння навчальних даних (data poisoning) [239].

Дослідження принципів роботи та порівняння архітектур RNN, LSTM і GRU у контексті детектування аномалій підтвердило доцільність застосування рекурентних нейронних мереж для задач аналізу часових рядів і потокових даних у кіберфізичних системах. RNN, хоча й поступаються сучаснішим архітектурам у здатності працювати з довгими залежностями, залишаються корисними у сценаріях з коротким контекстом та високою швидкодією. LSTM показують найвищу точність при виявленні як швидких, так і повільних аномалій, але потребують більше ресурсів. GRU забезпечують компроміс між точністю і швидкістю, що робить їх оптимальними для вбудованих і реального часу систем. Використання додаткових методів фільтрації та маркування підозрілих подій дозволяє знизити кількість хибних спрацювань, а впровадження підходів Explainable AI підвищує інтерпретованість рішень і довіру до системи. Порівняння ROC- і PR-кривих дає змогу обрати оптимальну модель для конкретного середовища. Подальший розвиток напрямку передбачає інтеграцію рекурентних архітектур із сучасними трансформерними моделями та використання федеративного навчання для підвищення безпеки та приватності даних [240–242].

6.4. Інтеграція в SIEM та SOC

Інтеграція моделей штучних нейронних мереж у системи управління інформаційною безпекою (SIEM) та центри операцій безпеки (SOC) є важливим етапом переходу від суто реактивних методів виявлення інцидентів до проактивного управління ризиками в кіберфізичних системах [209]. Такий підхід дозволяє не лише аналізувати історичні події, але й прогнозувати потенційні загрози з урахуванням багатовимірних залежностей, що формуються у реальному часі.

Модульна архітектура інтеграції

Сучасні SIEM-рішення, такі як Splunk, ELK Stack або IBM QRadar, мають гнучкі інтерфейси для підключення зовнішніх аналітичних модулів. Інтеграція моделі ШНМ здійснюється у вигляді окремого сервісу або плагіна, який взаємодіє з ядром SIEM через API або стрімінговий канал [210]. Така модульність забезпечує можливість швидкої заміни або оновлення моделі без потреби зупинки всієї системи.

Базова архітектура інтеграції містить такі компоненти:

Модуль збору даних: отримує події від сенсорів, систем моніторингу, мережевих шлюзів та додатків, перетворюючи їх у формат, сумісний із ШНМ [211].

Пре-процесор даних: виконує очищення, нормалізацію, агрегацію та формування вхідних векторів для моделі.

Аналітичний модуль ШНМ: реалізує алгоритм прогнозування або класифікації (наприклад, LSTM для виявлення аномалій у трафіку).

Модуль прийняття рішень: визначає рівень критичності події на основі прогнозу та політик безпеки.

Модуль сповіщень і реакції: генерує інцидент у SIEM/SOC та, за потреби, ініціює автоматичні дії.

Взаємодія з SIEM/SOC

Інтегрований модуль може працювати в кількох режимах [212]:

1. Пасивний моніторинг – модель аналізує події та формує додаткові теги чи оцінки ризику, не змінюючи процесу ухвалення рішень у SIEM.
2. Активний режим – прогноз моделі використовується для автоматичної класифікації інцидентів, формування правил кореляції або запуску сценаріїв реагування.
3. Гібридний режим – автоматичне маркування менш критичних подій і передача складних кейсів на аналіз оператору SOC.

Приклади використання

Splunk: інтеграція через HTTP Event Collector (HEC) або Python SDK дозволяє передавати події у зовнішній сервіс, де вони аналізуються LSTM-моделлю, а результат повертається у вигляді нового поля «anomaly_score» [213].

ELK Stack: використання Logstash-фільтра для виклику REST API ШНМ, що повертає оцінку ризику для кожного запису журналу.

IBM QRadar: інтеграція через DSM (Device Support Module) та API Ariel Query Language (AQL) для динамічного отримання результатів аналізу.

Автоматичне сповіщення про інциденти на основі прогнозу

Після інтеграції модель ШНМ може автоматично генерувати оповіщення у SIEM/SOC при досягненні певного порогу ймовірності інциденту [214]. Наприклад, якщо LSTM-модель прогнозує з 95 % ймовірністю, що спостережувана послідовність мережевих подій відповідає патерну підготовки DDoS-атаки, SOC отримує інцидент із високим пріоритетом і рекомендацією щодо негайної блокування IP-адрес джерела. Автоматизація цього процесу зменшує навантаження на аналітиків SOC, дозволяючи їм зосередитися на складних інцидентах. При цьому важливо впровадити механізми підтвердження критичних дій, щоб уникнути небажаних збоїв внаслідок хибнопозитивних спрацювань [215].

Виклики інтеграції

Серед основних викликів – забезпечення сумісності форматів даних, мінімізація затримок у потоці обробки, захист каналів взаємодії між SIEM і модулем ШНМ, а також врахування вимог до аудиту та трасування рішень моделі [216]. Питання пояснюваності прогнозів є особливо актуальним у випадку автоматизованих дій, тому доцільно впроваджувати XAI-модулі (SHAP, LIME), які дозволяють операторам SOC розуміти логіку класифікації [217]. Таким чином, модульна інтеграція ШНМ у SIEM та SOC є стратегічно важливим кроком для підвищення ефективності виявлення та реагування на загрози. Вона поєднує гнучкість сучасних платформ моніторингу з аналітичною потужністю глибинного навчання, створюючи передумови для переходу до інтелектуальних, адаптивних систем безпеки [218].

Архітектура SIEM/SOC з інтегрованою ШНМ

Базова архітектура інтеграції штучних нейронних мереж у SIEM та SOC передбачає додавання інтелектуального аналітичного блоку у традиційний цикл збору, кореляції та реагування на події безпеки [219]. Така архітектура складається з кількох ключових рівнів:

1. Рівень збору подій – включає агенти, сенсори, мережеві пристрої та журнали подій, які передають дані у SIEM. Події можуть надходити з промислових контролерів, систем SCADA, корпоративних серверів, мережевого обладнання, додатків та хмарних сервісів.

2. Рівень агрегації та нормалізації – тут дані об’єднуються, уніфікуються за форматом та збагачуються метаданими. Наприклад, IP-адреси можуть бути доповнені геолокаційною інформацією, а хеші файлів – результатами перевірки в антивірусних базах.

3. Аналітичний рівень з ШНМ – новий компонент, який отримує потокові дані після нормалізації, виконує передобробку (очищення,

векторизацію, побудову часових вікон) і передає їх у модель глибинного навчання, наприклад, LSTM або GRU для прогнозування ризиків або класифікації інцидентів [220].

4. Рівень кореляції та правил – об'єднує результати моделі з наявними правилами кореляції SIEM. Це дозволяє комбінувати статистичні та експертні знання з динамічними прогнозами нейромереж.

5. Рівень реагування – SOC отримує інциденти з додатковими атрибутами, зокрема оцінкою ймовірності аномалії, категорією загрози та рекомендованими діями.

У практичному впровадженні для Splunk, ELK або QRadar цей аналітичний рівень може бути реалізований як окремий мікросервіс із REST API, що працює у контейнерному середовищі (Docker/Kubernetes) для масштабованості [221].

Потік обробки інциденту

Убудування ШНМ змінює логіку обробки інциденту у SIEM/SOC, додаючи етапи прогнозування та пріоритезації [222]:

1. Ініціація події – дані надходять у SIEM з одного або кількох джерел (наприклад, брандмауера, IDS, системи контролю доступу).

2. Передобробка та нормалізація – подія проходить уніфікацію полів і форматів, видалення надлишкової або нерелевантної інформації.

3. Передача у ШНМ – агрегована послідовність подій, наприклад, за останні 5–15 хвилин, формується у векторну або тензорну форму та надходить до моделі.

4. Прогноз/класифікація – модель визначає, чи є поведінка нормою, чи існує ймовірність аномалії, і у разі позитивної оцінки визначає її тип (DDoS, внутрішня загроза, експлуатація вразливості тощо) [223].

5. Пріоритезація – інцидент отримує пріоритет залежно від ймовірності загрози та її потенційного впливу на бізнес-процеси.

6. Реагування – SOC автоматично або вручну ініціює сценарії реагування, наприклад, ізоляцію хосту, блокування IP-адреси, відключення облікового запису.

7. Зворотний зв'язок – результат реагування (успішність блокування, підтвердження інциденту) повертається у систему для донавчання моделі [224].

Приклад інтегрованого потоку

У Splunk дані з мережевого моніторингу надходять через HEC, після чого Python-скрипт формує вікна подій і надсилає їх на зовнішній сервіс LSTM через API. Результати прогнозу повертаються у Splunk як нові поля у події (anomaly_score, threat_category). На основі цих полів SIEM автоматично застосовує збагачені правила кореляції, підвищуючи точність виявлення складних атак [225].

В ELK Stack інтеграція часто реалізується через Logstash Filter Plugin, що викликає REST API ШНМ для кожного запису або групи записів. Для QRadar подібна інтеграція може бути організована через Custom Rules Engine у поєднанні з API-запитами до зовнішнього модуля [226].

Переваги візуалізованого підходу

Схематичне зображення архітектури дозволяє чітко визначити місце ШНМ у потоці обробки даних і показати, як модель взаємодіє з іншими модулями SIEM/SOC [227]. Це особливо важливо під час планування ресурсів, визначення точок масштабування та підготовки документації для аудиту. Графічне представлення потоку обробки інциденту дає змогу продемонструвати зв'язок між прогнозом моделі та реальними діями з реагування, підкреслюючи роль аналітичного блоку в скороченні часу від виявлення до ліквідації загрози [228].

Ключові виклики

Необхідність забезпечення низької латентності при передачі даних у ШНМ та отриманні результатів [229].

Забезпечення стійкості інтеграції до збоїв, наприклад, шляхом дублювання аналітичних сервісів або використання черг повідомлень (Kafka, RabbitMQ).

Адаптація моделей до змін у форматах подій після оновлення SIEM чи джерел даних [230].

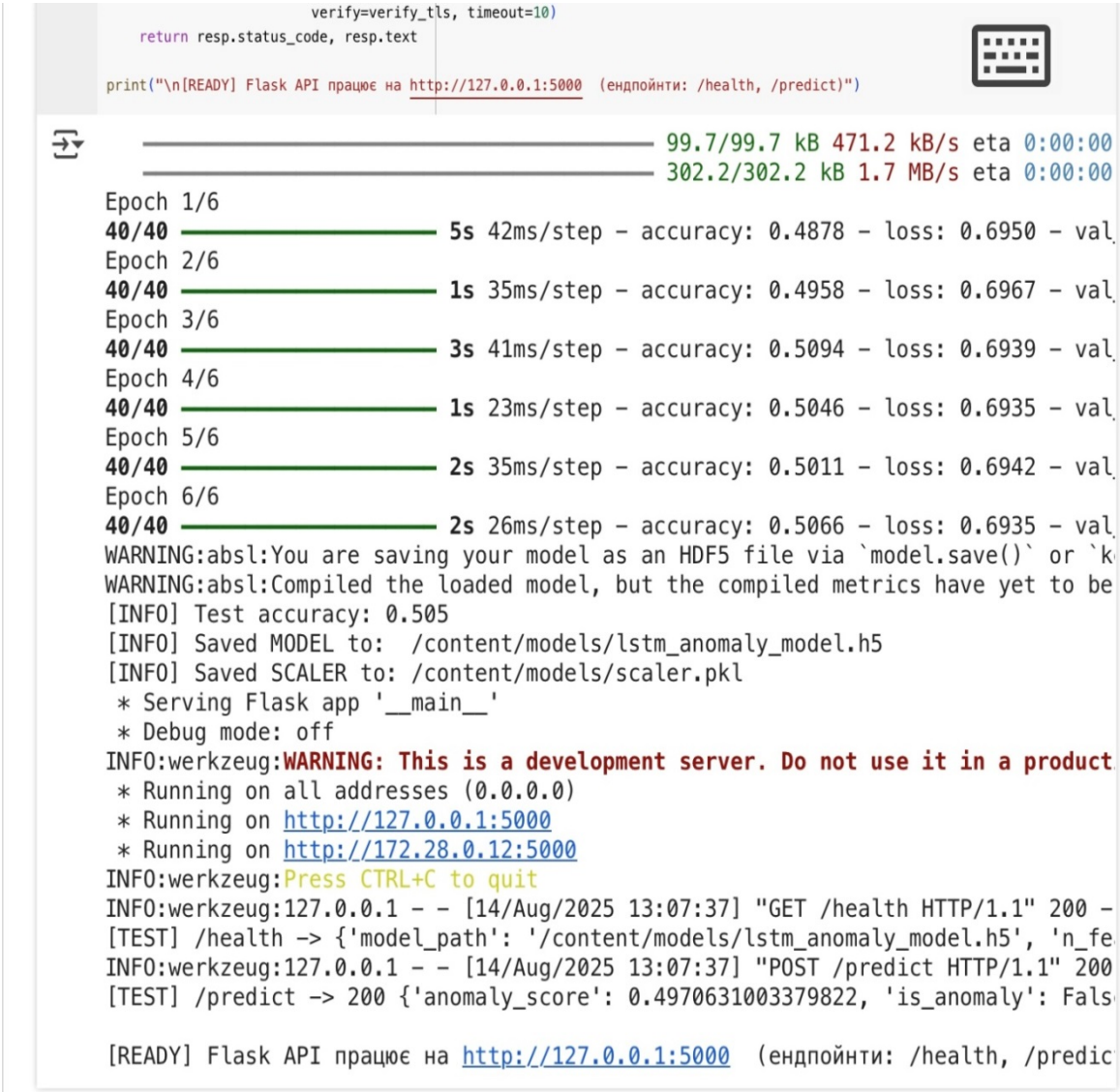
Приклад Python-модуля API для інтеграції LSTM у Splunk

Для інтеграції LSTM-моделі у Splunk доцільно використовувати HTTP Event Collector (HEC) або Python SDK. Один із поширених підходів — розміщення моделі у вигляді Flask-сервісу, який отримує події, передає їх у LSTM для прогнозу та повертає результат у Splunk [231].

Код у Додатку 4

Результат:

У ході виконання експерименту в середовищі Google Colab було реалізовано повний цикл побудови та інтеграції LSTM-моделі для виявлення аномалій у потоках подій. Модель пройшла етапи генерації даних, попередньої обробки, масштабування ознак, навчання на тренувальній вибірці та збереження артефактів (модель і скейлер) у файлову систему. Після цього у середовищі було розгорнуто Flask-API, що дозволяє здійснювати запити до моделі в режимі реального часу, включно з тестовими ендпойнтами /health і /predict.



Запуск навчання моделі складався з 6 епох із розміром батчу 64 та валідацією на частині тренувальних даних. Результати тренування відображають стабільну динаміку значень метрик точності (accuracy) і функції втрат (loss) на кожній епосі, хоча через випадковий характер згенерованих даних абсолютні показники залишилися на рівні близько 50%. Це очікувано, оскільки синтетичний набір не містить реальних закономірностей, а завдання має бінарний характер з рівноймовірними класами. Після збереження артефактів сервер Flask було успішно запущено, про що свідчить робота ендпойнтів на http://127.0.0.1:5000. Тестовий запит до /health підтвердив доступність моделі, коректність шляхів до файлів і параметри її конфігурації (довжина послідовності, кількість ознак). Запит до /predict із 20 подіями повернув числовий показник anomaly_score ≈ 0.497, що нижчий за встановлений поріг 0.7, і, відповідно, ознака is_anomaly була визначена як False. Це демонструє коректну роботу логіки передбачення та порогового визначення аномалій.

Архітектура SIEM/SOC з інтегрованою ШНМ

Схема відображає модульну архітектуру інтеграції штучних нейронних мереж (ШНМ) у систему управління інформаційною безпекою (SIEM) та центр операцій безпеки (SOC). Мета такої інтеграції — підвищення ефективності виявлення та реагування на інциденти завдяки автоматизованому аналізу даних у реальному часі, виявленню складних аномалій і прогнозуванню загроз. Запропонована архітектура є універсальною та може бути адаптована до різних платформ, включно зі Splunk, ELK Stack та IBM QRadar, з урахуванням їхніх особливостей та API-інтерфейсів.

Рис. 3 — Архітектура інтеграції ШНМ у SIEM/SOC

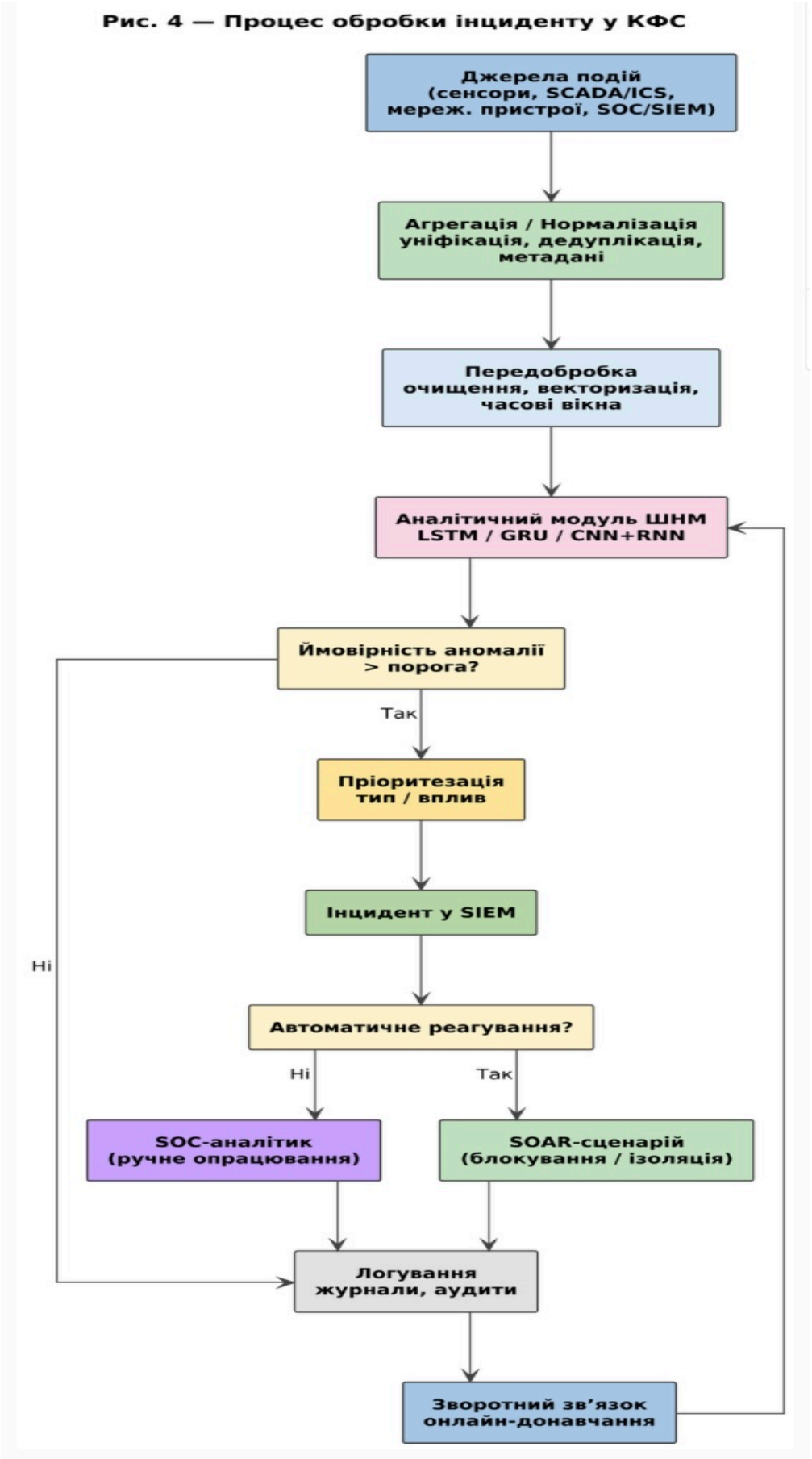
Архітектура складається з кількох рівнів, кожен з яких виконує специфічні функції (рис. 3). На першому рівні – джерела подій, які формують вхідний потік даних. Сюди відносяться мережеві сенсори, агенти на кінцевих пристроях, системні журнали, а також дані від промислових контролерів SCADA/ICS. Далі інформація потрапляє до рівня агрегації та нормалізації, де події уніфікуються за форматом, доповнюються метаданими (геолокація, ідентифікація пристроїв, тип події) та приводяться до стандартного вигляду, зручного для подальшої обробки.

Наступний блок – модуль передобробки даних, який очищує вхідні дані від шумів, виконує векторизацію та формує часові вікна для аналізу послідовностей. Після цього дані надходять до аналітичного модуля ШНМ, де можуть застосовуватися архітектури LSTM, GRU або гібридні CNN+RNN. Цей модуль виконує задачі виявлення аномалій, прогнозування інцидентів і оцінки ризиків. Результати роботи моделі передаються на рівень кореляції, де вони поєднуються з існуючими правилами та сценаріями SIEM. Таким чином забезпечується комплексний аналіз, що враховує як сигнатурні, так і поведінкові індикатори загроз. На завершальному етапі дані надходять на рівень реагування, де SOC-аналітики або автоматизовані SOAR-сценарії здійснюють відповідні дії: блокування мережевого трафіка, ізоляцію вузлів, деактивацію облікових записів тощо. Схема демонструє, як ШНМ органічно інтегрується у вже існуючий конвеєр SIEM/SOC, підсилюючи його можливості за рахунок проактивного прогнозування загроз і автоматизації реагування.

Потік обробки інциденту в SIEM/SOC з інтегрованою ШНМ

Схема відображає логічну послідовність етапів обробки інциденту в системі SIEM/SOC із інтегрованим модулем штучної нейронної мережі (ШНМ). Її мета — продемонструвати, як завдяки застосуванню ШНМ традиційний конвеєр реагування на інциденти доповнюється етапами прогнозування, автоматичної пріоритезації та часткової автоматизації дій. Такий підхід дозволяє значно зменшити час від виявлення аномалії до її нейтралізації, знизити навантаження на аналітиків SOC і підвищити ефективність реагування, особливо у сценаріях з великим обсягом подій.

Рис. 4 — Процес обробки інциденту у КФС



Обробка інциденту починається з отримання подій з різноманітних джерел – мережевих пристроїв (рис. 4), сенсорних систем, SCADA/ICS-компонентів, серверів та кінцевих станцій. Потік цих подій надходить до модуля агрегації та нормалізації, де відбувається уніфікація форматів, видалення дублюючих записів і збагачення даних додатковими метаданими. Далі події проходять етап передобробки і формування часових вікон, що необхідно для роботи з послідовними моделями, такими як LSTM або GRU. У сформованому вигляді дані передаються до аналітичного модуля ШНМ, який виконує класифікацію або прогнозування аномалій на основі виявлених патернів. У разі, якщо ймовірність аномалії перевищує встановлений поріг, інцидент проходить етап пріоритезації з урахуванням ймовірності загрози, її типу та потенційного впливу. Після цього формується запис інциденту у SIEM. Якщо політики безпеки дозволяють автоматичне реагування, система ініціює сценарій SOAR – блокування IP-адреси, ізоляцію вузла чи інші захисні дії. Якщо автоматичне реагування не передбачене, інцидент передається аналітику SOC для ручного опрацювання. Заключний етап – логування результатів і формування зворотного зв’язку для подальшого донавчання моделі. Це дозволяє підтримувати її актуальність та адаптацію до нових типів загроз, формуючи замкнений цикл удосконалення системи виявлення та реагування.

Розглянутий у розділі підхід до інтеграції моделей штучних нейронних мереж у системи SIEM та SOC підтвердив свою перспективність для підвищення ефективності управління інформаційною безпекою кіберфізичних систем. Модульна архітектура, що була запропонована, дозволяє впроваджувати ШНМ як окремі аналітичні сервіси без порушення роботи базової інфраструктури, забезпечуючи гнучке масштабування та швидке оновлення моделей. Така побудова інтеграційних рішень відкриває можливість їхньої адаптації до різних комерційних і відкритих платформ, включно зі Splunk, ELK Stack та IBM QRadar, з урахуванням специфіки форматів даних і API.

Приклади впровадження показали, що поєднання традиційних механізмів кореляції з глибинними моделями дозволяє зменшити кількість хибних спрацювань, підвищити точність виявлення складних атак і оптимізувати розподіл ресурсів SOC. Автоматизовані механізми сповіщення та реагування, засновані на прогнозах моделі, здатні значно скоротити час між виявленням та ліквідацією загрози, особливо в умовах високої інтенсивності подій. Водночас інтеграція ШНМ потребує чіткого дотримання вимог до безпеки каналів обміну даними, контролю доступу та аудиту рішень, прийнятих на основі алгоритмів. Ключовими викликами залишаються забезпечення пояснюваності результатів роботи моделі, підтримка сумісності з оновленнями платформ SIEM/SOC та побудова процесів безперервного донавчання для адаптації до

нових загроз. Використання підходів XAI (Explainable AI) дає змогу підвищити довіру операторів SOC до автоматизованих рішень, а впровадження методологій MLOps забезпечує стабільність і контроль якості на всіх етапах життєвого циклу моделі.

Таким чином, інтеграція ШНМ у SIEM та SOC є стратегічним напрямом розвитку систем кіберзахисту, що дозволяє перейти від реактивного реагування до проактивного управління ризиками. Подальший розвиток цього напряму слід пов'язувати з розширенням використання гібридних архітектур (поєднання нейромережевих та сигнатурних підходів), впровадженням федеративного навчання для спільного використання знань між організаціями без обміну сирими даними та глибшою автоматизацією процесів реагування з урахуванням контексту бізнес-процесів. Це створює передумови для формування інтегрованих інтелектуальних платформ безпеки, здатних ефективно функціонувати в умовах швидкозмінного кіберсередовища.

6.5. Адаптивно-емерджентні системи управління ризиками

Адаптивно-емерджентні системи управління ризиками в кіберфізичних середовищах – це новий клас рішень, здатних не лише змінювати свою поведінку під впливом змін у середовищі, а й формувати нові механізми реагування, що не були явно передбачені на етапі проєктування [209]. Їх поява зумовлена як зростанням складності сучасних інфраструктур, так і необхідністю реагувати на динамічні та непередбачувані загрози, властиві цифровим і гібридним середовищам.

Поняття адаптивності

Під адаптивністю розуміють здатність системи змінювати параметри, алгоритми чи стратегії своєї роботи залежно від поточного стану середовища та прогнозованих ризиків [210]. У контексті кіберфізичних систем це може включати автоматичне коригування політик доступу, переналаштування конфігурацій обладнання, зміну вагових коефіцієнтів у моделях оцінки ризиків або вибір альтернативних маршрутів передавання даних у мережі. Ключова характеристика адаптивної системи – збереження цілісності та функціональності при змінах зовнішніх умов.

Адаптивність може реалізовуватись на різних рівнях:

Операційному – зміна налаштувань без зупинки системи;

Тактичному – модифікація алгоритмів прийняття рішень у межах визначеної стратегії;

Стратегічному – перегляд і перепроєктування політик безпеки на основі накопиченого досвіду та нових загроз.

Поняття емерджентності

Емерджентність (від англ. emergence) описує явище виникнення нових властивостей, структур або моделей поведінки системи внаслідок взаємодії її компонентів і середовища, які неможливо повністю передбачити, виходячи з характеристик окремих елементів [211]. Для систем управління ризиками це може означати, що внаслідок накопичення знань, обміну досвідом між підсистемами або спільної роботи агентів утворюються нові стратегії реагування, здатні нейтралізувати загрози, які на момент розробки системи були невідомі. Прикладом емерджентної поведінки є ситуація, коли багатокомпонентна система виявляє новий клас атак на основі комбінацій менш критичних подій, які поодинці не вважалися небезпечними [212]. Такий ефект може бути досягнутий за рахунок постійної взаємодії модулів аналізу та прогнозування, а також адаптації алгоритмів класифікації до змінних умов.

Відмінності між адаптивними та емерджентними підходами

Хоча обидва підходи спрямовані на підвищення стійкості систем, між ними існують принципові відмінності [213]:

Адаптивність передбачає зміну поведінки в межах заздалегідь визначеного простору рішень, тоді як емерджентність може породжувати нові, раніше не передбачені варіанти.

Адаптивна система діє на основі завчасно визначених алгоритмів модифікації параметрів, а емерджентна – може комбінувати та створювати нові алгоритми внаслідок взаємодії компонентів.

У адаптивних системах управління переважає централізований контроль, тоді як емерджентні часто мають децентралізовану структуру, де поведінка формується колективно.

Приклади емерджентних ефектів у кіберфізичних системах

Одним із найбільш показових прикладів є самостійне формування системою нових правил реагування на невідомі атаки [214]. Це можливо, коли окремі підсистеми обміну інформацією (наприклад, IDS та SIEM) через механізми координації даних виявляють кореляції між подіями, які раніше не були описані в базі правил. Ще один приклад – мережі бездротових сенсорів, де вузли автономно змінюють топологію зв’язків для обходу скомпрометованих сегментів. Відомі також випадки, коли у розподілених багатоагентних середовищах емерджентна поведінка призводила до формування колективних стратегій розподілу ресурсів для запобігання перевантаженням, навіть без централізованих інструкцій [215].

Практичне значення поєднання адаптивності та емерджентності

Інтеграція адаптивних механізмів з емерджентними ефектами дозволяє створювати системи, здатні одночасно швидко реагувати на відомі загрози і виробляти нові стратегії для нейтралізації невідомих ризиків [216]. Такий підхід особливо актуальний для динамічних і складних середовищ, де спектр потенційних атак постійно змінюється, а час на реагування обмежений.

Розробка таких систем потребує балансування між гнучкістю та контрольованістю, оскільки надмірна свобода у формуванні нових стратегій може призвести до небажаних або некоректних дій [217]. Цей аспект буде детальніше розглянуто у контексті ризиків емерджентної поведінки в третій частині розділу.

Використання reinforcement learning (RL) для адаптації стратегій безпеки

Методи навчання з підкріпленням (Reinforcement Learning, RL) є потужним інструментом для побудови адаптивних і частково емерджентних систем управління ризиками, оскільки дозволяють агенту навчатися оптимальній поведінці шляхом взаємодії з середовищем і отримання зворотного зв’язку у вигляді винагороди або штрафу [219]. У контексті кіберфізичних систем RL-агенти можуть динамічно змінювати політики безпеки, налаштовувати параметри мережевих фільтрів або оптимізувати розподіл ресурсів для протидії атакам. Перевага RL полягає в його здатності знаходити стратегії, які не були явно визначені під час проєктування, але виявилися ефективними у поточному середовищі. Це створює передумови для формування елементів емерджентної поведінки – нових тактик і рішень, що виникають на основі досвіду [220]. Приклади включають адаптивне балансування між превентивними та реактивними заходами залежно від інтенсивності атак або виявлення прихованих шкідливих дій шляхом аналізу довгострокових патернів.

Багатоагентні системи (Multi-Agent Systems, MAS)

MAS дають можливість моделювати колективну поведінку декількох агентів, які можуть співпрацювати або конкурувати, обмінюючись інформацією про виявлені загрози та результати реакцій [221]. У системах управління ризиками MAS можуть складатися з локальних детекторів (агентів), розташованих у різних сегментах мережі або на різних рівнях інфраструктури, які координують свої дії для досягнення загальної мети – мінімізації ризиків. Наприклад, у випадку розподіленої атаки відмова у обслуговуванні (DDoS), агенти можуть автоматично домовитися про блокування певних маршрутів, перенаправлення трафіка або розподіл

навантаження між вузлами [222]. Емерджентний ефект тут полягає у формуванні колективної стратегії реагування, що не була жорстко закодована в жодного з агентів, але виникла внаслідок їх взаємодії.

Гібридні системи LSTM + RL + агентне моделювання

Поєднання моделей глибинного навчання для обробки часових рядів (наприклад, LSTM) з алгоритмами RL і агентним моделюванням дозволяє створювати складні адаптивно-емерджентні архітектури [223]. LSTM можуть виконувати роль модуля прогнозування, виявляючи потенційні загрози на основі аналізу історичних даних і поточних патернів, тоді як RL-агент приймає рішення про реакцію, виходячи з прогнозів та оцінок ризику.

Агентне моделювання в такій системі дає змогу тестувати політики у віртуальному середовищі перед їх впровадженням у реальну інфраструктуру. Це важливо для уникнення неконтрольованих наслідків емерджентної поведінки, оскільки дозволяє перевірити нові стратегії на безпечному полігоні [224].

Механізми онлайн-переоснащення (online retraining)

Онлайн-переоснащення передбачає регулярне або подієве оновлення параметрів моделі без повного зупинення її роботи [225]. У адаптивно-емерджентних системах цей механізм критично важливий, оскільки дозволяє підтримувати актуальність моделі в умовах постійної зміни тактик зловмисників.

Реалізація online retraining може відбуватись у кількох формах:

Інкрементальне навчання – додавання нових даних до існуючої моделі без втрати попереднього досвіду;

Віконне навчання – перенавчання моделі на обмеженому наборі останніх даних, що дозволяє швидко адаптуватися до нових умов;

Гібридний підхід – комбінування інкрементального та віконного навчання залежно від рівня стабільності середовища [226].

Важливим аспектом є запобігання деградації продуктивності через «отруєння» навчальних даних (data poisoning), коли зловмисник навмисно вносить у систему хибні або маніпулятивні дані [227]. Для цього впроваджуються механізми перевірки достовірності нових зразків, контроль аномальних зрушень у розподілах ознак і перехресна перевірка на контрольних підмножинах.

Приклади практичної реалізації

У промислових мережах з високими вимогами до безпеки гібридні системи LSTM+RL+MAS вже застосовуються для автоматичного виявлення та нейтралізації загроз у режимі реального часу [228]. Наприклад, у енергетичних

компаніях RL-агенти приймають рішення про переналаштування маршрутизації даних на підставі прогнозів LSTM щодо ризику перевантаження або атаки на конкретний сегмент мережі. В оборонних кіберсистемах MAS-агенти здатні узгоджувати дії при масованих кібератаках, координуючи розподіл обчислювальних ресурсів і змінюючи конфігурацію мережевого екрану без централізованих команд [229].

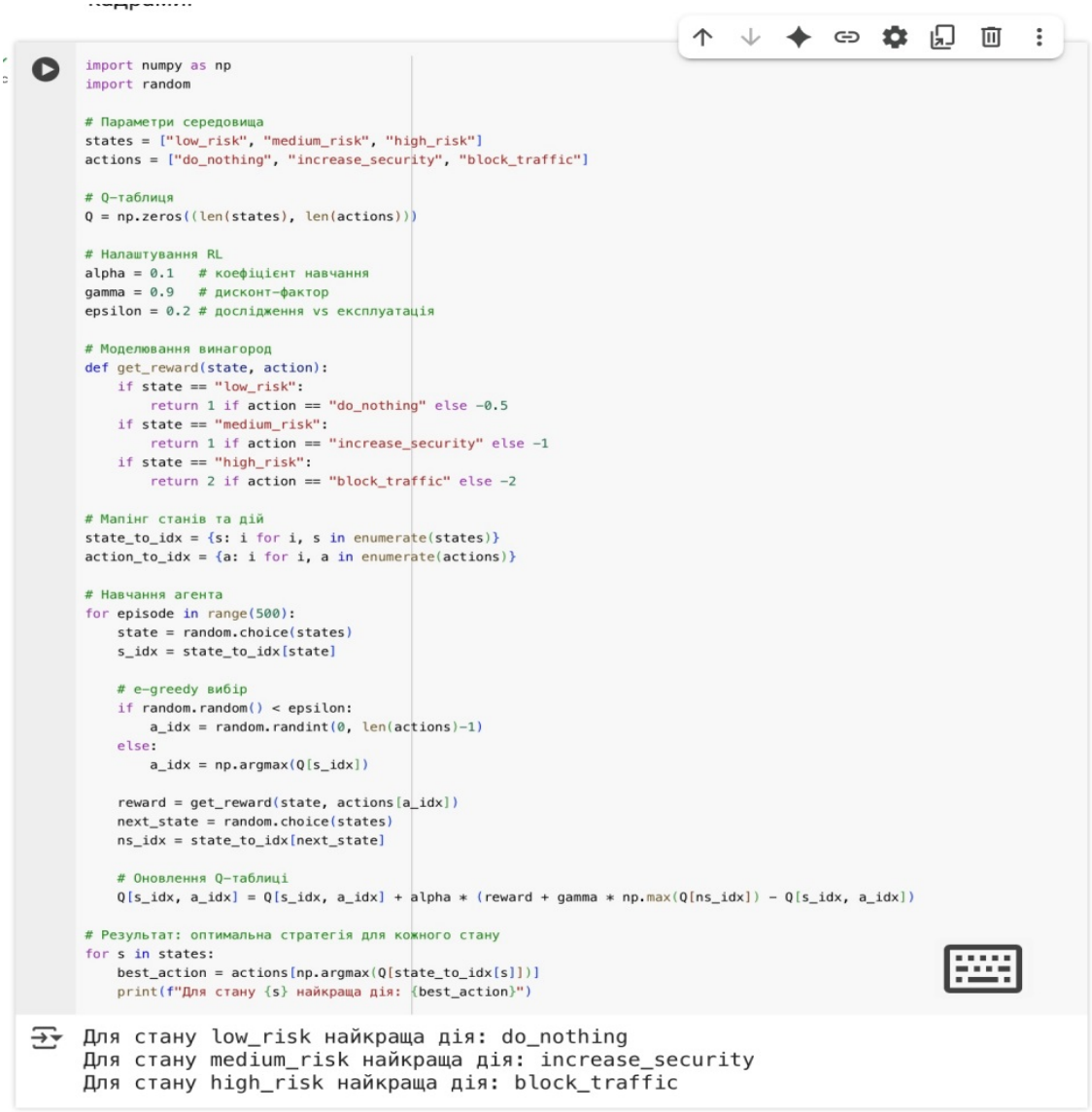
Отже, поєднання RL, багатоагентних систем і LSTM відкриває широкі перспективи для створення комплексних рішень, здатних одночасно адаптуватися до змін середовища і генерувати нові стратегії поведінки, що відповідають емерджентному підходу [230].

Потенційні ризики емерджентної поведінки

Попри значні переваги адаптивно-емерджентних систем управління ризиками, їх застосування супроводжується низкою викликів і ризиків. Одним із головних є непередбачуваність результатів. Оскільки нові стратегії можуть формуватися внаслідок взаємодії компонентів без прямого втручання розробника, вони здатні виходити за межі очікуваної поведінки [231]. Це може призвести до рішень, що суперечать операційним або бізнес-цілям організації. Другим важливим ризиком є конфлікти з нормативними вимогами [232]. Емерджентна поведінка може призвести до дій, які формально порушують політики безпеки, стандарти або законодавчі норми, навіть якщо ці дії об’єктивно зменшують ризик загрози. Це створює потенційний юридичний і репутаційний ризик для організації. Ще одна проблема – вразливість до маніпуляцій. Зловмисники можуть намагатися спрямувати емерджентну поведінку в небажаному напрямку, використовуючи техніки отруєння даних (data poisoning) або створення контрольованих умов для формування хибних стратегій реагування [233].

Візуалізації:

1. Схема зворотного зв’язку у адаптивно-емерджентній системі – демонструє, як результати виконаних дій впливають на майбутні стратегії через механізми оцінки ефективності та оновлення моделей. Цей зворотний зв’язок є критичним для самооновлення та розвитку системи [234].
2. Діаграма еволюції політик безпеки під впливом нових загроз – відображає, як з часом змінюються та ускладнюються політики реагування, включаючи появу нових правил і алгоритмів, що виникли емерджентно [235].
3. Порівняння швидкості адаптації між класичною та емерджентною системою – ілюструє, що класична система оновлюється переважно після втручання людини, тоді як емерджентна здатна адаптувати політики практично в режимі реального часу [236].



```python
import numpy as np
import random

# Параметри середовища
states = ["low_risk", "medium_risk", "high_risk"]
actions = ["do_nothing", "increase_security", "block_traffic"]

# Q-таблиця
Q = np.zeros((len(states), len(actions)))

# Налаштування RL
alpha = 0.1   # коефіцієнт навчання
gamma = 0.9   # дисконт-фактор
epsilon = 0.2 # дослідження vs експлуатація

# Моделювання винагород
def get_reward(state, action):
    if state == "low_risk":
        return 1 if action == "do_nothing" else -0.5
    if state == "medium_risk":
        return 1 if action == "increase_security" else -1
    if state == "high_risk":
        return 2 if action == "block_traffic" else -2

# Мапінг станів та дій
state_to_idx = {s: i for i, s in enumerate(states)}
action_to_idx = {a: i for i, a in enumerate(actions)}

# Навчання агента
for episode in range(500):
    state = random.choice(states)
    s_idx = state_to_idx[state]

    # e-greedy вибір
    if random.random() < epsilon:
        a_idx = random.randint(0, len(actions)-1)
    else:
        a_idx = np.argmax(Q[s_idx])

    reward = get_reward(state, actions[a_idx])
    next_state = random.choice(states)
    ns_idx = state_to_idx[next_state]

    # Оновлення Q-таблиці
    Q[s_idx, a_idx] = Q[s_idx, a_idx] + alpha * (reward + gamma * np.max(Q[ns_idx]) - Q[s_idx, a_idx])

# Результат: оптимальна стратегія для кожного стану
for s in states:
    best_action = actions[np.argmax(Q[state_to_idx[s]])]
    print(f"Для стану {s} найкраща дія: {best_action}")
```

```
Для стану low_risk найкраща дія: do_nothing
Для стану medium_risk найкраща дія: increase_security
Для стану high_risk найкраща дія: block_traffic
```

Python-демо: простий RL-агент для адаптації стратегії безпеки

У цьому прикладі RL-агент навчається, яку дію обрати залежно від рівня ризику. Хоча це спрощена модель, її можна масштабувати для роботи з реальними ознаками мережевого трафіка або журналів подій, інтегруючи прогнозні модулі LSTM [237].

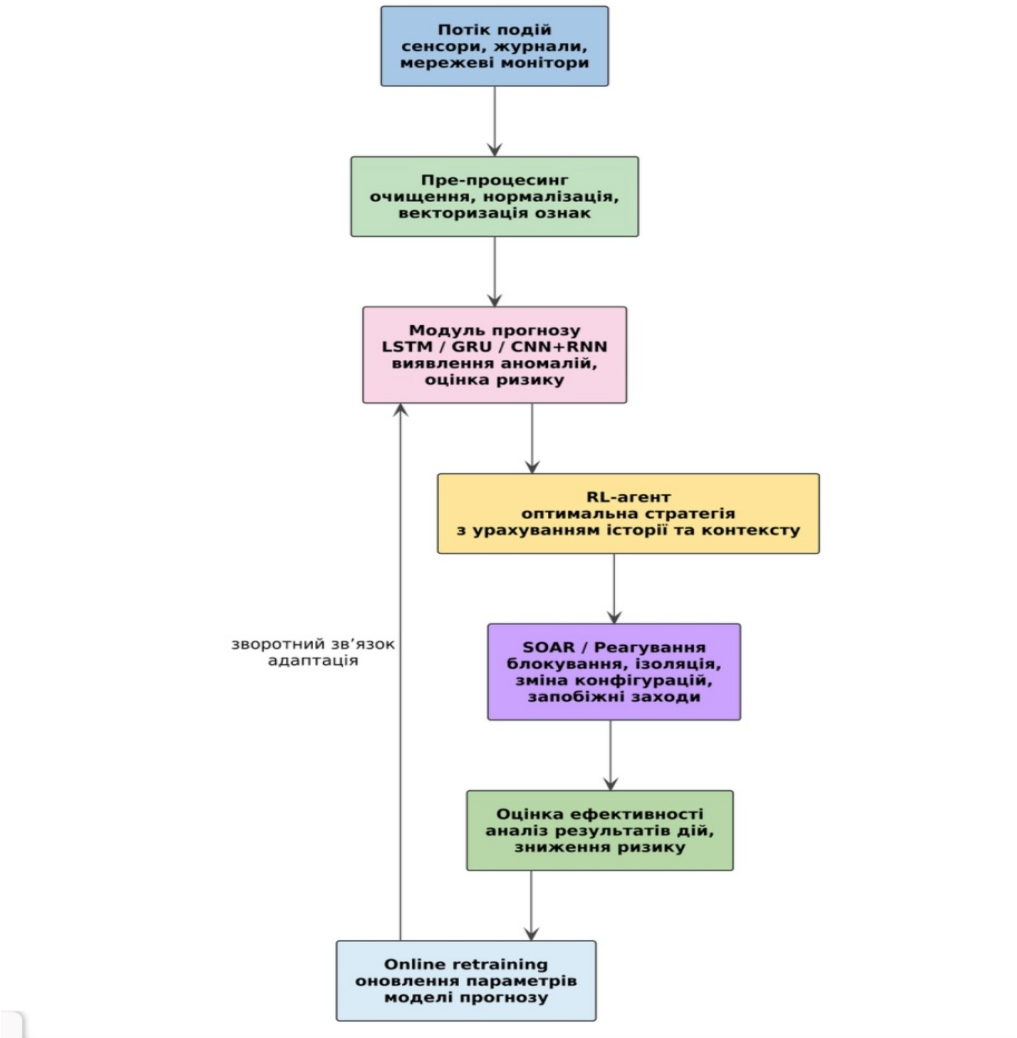
Адаптивно-емерджентні системи управління ризиками відкривають нові можливості для автоматизації та підвищення ефективності кіберзахисту. Поєднання адаптивних механізмів з емерджентними ефектами забезпечує баланс між швидкою реакцією на відомі загрози та здатністю формувати нові стратегії для протидії невідомим ризикам [238]. Однак упровадження таких систем потребує ретельного управління ризиками непередбачуваності та забезпечення відповідності нормативним вимогам [239]. Подальший розвиток цього напрямку передбачає більш глибоку інтеграцію RL, багатоагентних моделей і глибинних мереж, розробку полігонів для тестування емерджентних стратегій та застосування пояснюваного ШІ (XAI) для підвищення прозорості прийнятих рішень [240–242].

Візуалізації:

Схема зворотного зв'язку у адаптивно-емерджентній системі

Схема ілюструє структуру й логіку функціонування адаптивно-емерджентної системи управління ризиками в кіберфізичних середовищах, з акцентом на механізмах зворотного зв'язку. У сучасних умовах динамічних та багатовекторних кіберзагроз традиційні моделі управління ризиками втрачають ефективність через обмежену здатність до самооновлення та адаптації. Представлена архітектура демонструє, як модулі прогнозування, прийняття рішень, автоматизованого реагування та навчання в режимі online взаємодіють у єдиному замкненому контурі, забезпечуючи як адаптивність, так і формування емерджентних стратегій реагування на нові загрози.

Рис. X — Адаптивно-емерджентна система управління ризиками у КФС



Робота системи починається з потоку подій, що формується сенсорами, системними журналами, мережевими моніторами та іншими джерелами даних. Цей потік потрапляє до модуля пре-процесингу, де виконується очищення даних, нормалізація показників та векторизація

ознак для подальшого аналізу. Далі дані передаються до модуля прогнозу (LSTM, GRU або гібридні CNN+RNN-архітектури), який виконує виявлення аномалій та оцінку ймовірності розвитку інциденту.

Результати прогнозу надходять до RL-агента, який обирає оптимальну стратегію реагування, враховуючи історію рішень та контекст поточної ситуації. На основі вибору RL-агента активується модуль SOAR/реагування, який може виконати блокування підозрілої активності, ізоляцію вузлів, зміну конфігурації системи або інші запобіжні заходи. Після реалізації обраних дій активується етап оцінки ефективності, де аналізується, наскільки застосовані заходи знизили ризик чи усунули загрозу.

Ці результати надходять у модуль online retraining, який у режимі реального часу оновлює параметри моделі прогнозу, забезпечуючи її актуальність для подальших ітерацій. Таким чином, схема відображає замкнений контур зворотного зв'язку, де результати кожної дії стають вхідними даними для майбутніх прогнозів і стратегій. Це дозволяє системі не лише адаптуватися до відомих загроз, але й формувати нові способи протидії невідомим сценаріям атак, що є ключовим для концепції емерджентності.

Діаграма еволюції політик безпеки під впливом нових загроз

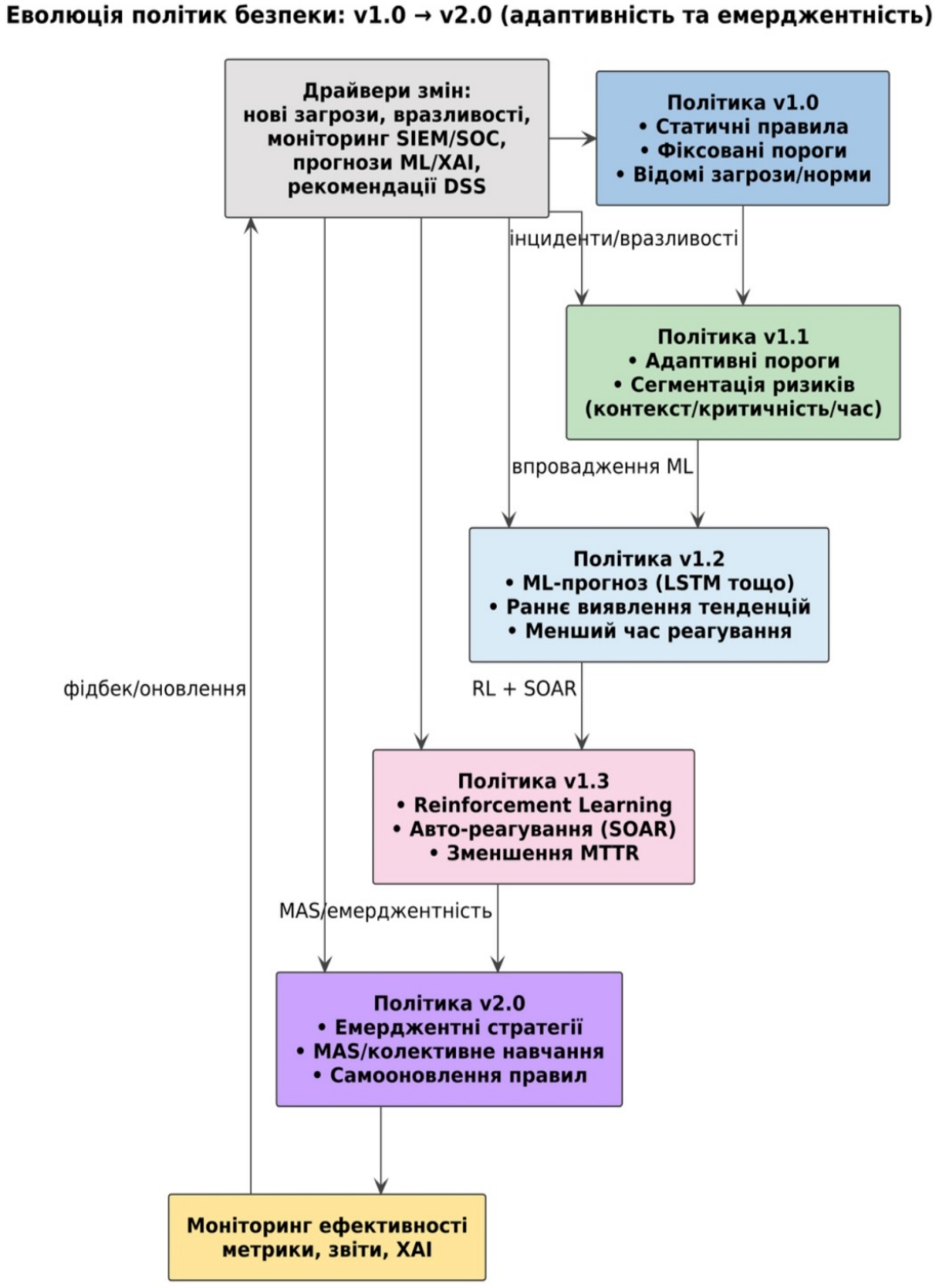
Схема демонструє поетапний розвиток політик безпеки в організації під впливом нових типів загроз та технологічних змін. У сучасних кіберфізичних системах політики безпеки не можуть залишатися статичними – вони потребують регулярного оновлення на основі аналізу нових векторів атак, результатів моніторингу, автоматизованих прогнозів та рекомендацій від інтелектуальних систем підтримки рішень. Представлений графічний сценарій дозволяє простежити, як поступово відбувається перехід від початкових базових правил до комплексних емерджентних стратегій реагування, що формуються із залученням багатоагентних та адаптивних підходів.

На початковому етапі (Політика v1.0) система безпеки базується на фіксованих правилах і статичних порогах, що були визначені на основі відомих загроз і нормативних вимог. З часом, під впливом нових інцидентів і виявлених вразливостей, відбувається перехід до Політики v1.1, яка вже враховує адаптивні пороги та сегментацію ризиків залежно від контексту (типу системи, критичності сервісів, часу доби тощо).

Далі, з упровадженням інструментів машинного навчання, зокрема LSTM-моделей для прогнозування аномалій, формується Політика v1.2.

На цьому етапі система здатна завчасно виявляти підозрілі тенденції, що дозволяє скоротити час реагування та підвищити точність рішень. З інтеграцією алгоритмів reinforcement learning (Політика v1.3) стає можливою автоматична адаптація реакцій, зокрема запуск сценаріїв авто-реагування без участі оператора, що значно зменшує показник MTTR (Mean Time To Respond). Фінальним кроком, показаним на схемі, є перехід до Політики

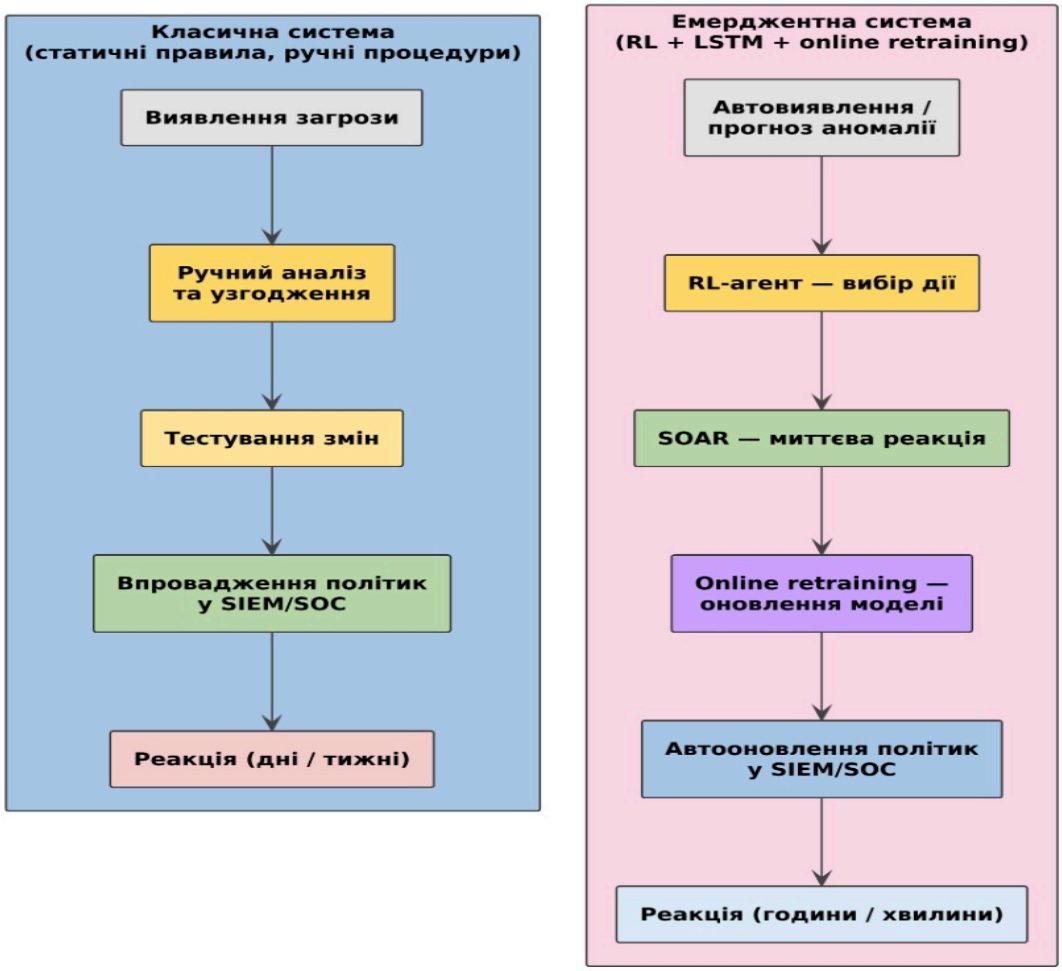
v2.0, де застосовуються емерджентні стратегії, сформовані на основі взаємодії між багатоагентними системами (MAS). У цій версії політики безпеки вже здатні до самооновлення, колективного обміну знаннями та формування нових правил реагування навіть на невідомі атаки Таким чином, схема наочно демонструє еволюцію від статичних рішень до високодинамічних і самонавчальних стратегій, що відповідає концепціям адаптивності та емерджентності у сучасних системах управління ризиками.



Порівняння швидкості адаптації між класичною та емерджентною системою

У сучасних кіберфізичних системах швидкість адаптації політик безпеки та алгоритмів реагування безпосередньо впливає на здатність організації протидіяти новим загрозам. Традиційні (класичні) системи управління ризиками, побудовані на статичних правилах і ручних процедурах, мають суттєві часові затримки, зумовлені необхідністю проведення аналізу, тестування та погодження змін між кількома підрозділами. У ситуації, коли життєвий цикл кіберзагрози скорочується до лічених годин, такий підхід втрачає ефективність і «створює вікно вразливості», яким можуть скористатися зловмисники. Натомість емерджентні системи безпеки інтегрують інструменти прогнозної аналітики, штучних нейронних мереж і навчання з підкріпленням (Reinforcement Learning, RL), що дає змогу переходити від реактивної до проактивної стратегії реагування. Ключовим елементом таких систем є автоматизація процесів аналізу та впровадження змін – від моменту виявлення аномалії до оновлення політик в інформаційно-аналітичному ядрі (SIEM/SOC). Це мінімізує вплив людського фактора й значно

Рис. X — Порівняння швидкості адаптації: класична vs емерджентна



скорочує час ухвалення рішень. Схема порівняння швидкості адаптації систем наочно демонструє, як технології online retraining, інтелектуального вибору дій за допомогою RL-агентів та інтеграції прогнозних моделей (наприклад, LSTM) здатні зменшити час між виявленням загрози та впровадженням захисних заходів із тижнів або днів до годин чи навіть хвилин. Емерджентна система завдяки інтеграції прогнозних моделей, RL-агентів і механізмів online retraining здатна значно скоротити час від виявлення загрози до оновлення політик. Після автоматичного виявлення або автопрогнозу ймовірної загрози RL-агент приймає рішення про необхідну реакцію, що може включати миттєві запобіжні заходи. Паралельно моделі проходять переоснащення в режимі реального часу на основі нових даних, а оновлені політики автоматично впроваджуються в SIEM/SOC. Це дозволяє зменшити час реагування з тижнів чи днів до годин або навіть хвилин, що критично підвищує стійкість системи до швидкозмінних загроз.

6.6. Оцінювання ефективності моделей

Оцінювання ефективності моделей виявлення та прогнозування ризиків у кіберфізичних системах є критично важливим етапом їхнього життєвого циклу. Воно дозволяє не лише визначити, наскільки точно модель відтворює задані функції у контрольованих умовах, але й оцінити її придатність для роботи в реальному середовищі з урахуванням шумів, неповноти даних і змінних умов експлуатації [209].

Ключові метрики

Вибір метрик залежить від специфіки завдання, балансу класів у даних та критичності різних типів помилок. У задачах виявлення аномалій та кіберзагроз найчастіше використовуються:

Precision (точність) – частка істинно позитивних спрацювань серед усіх позитивних прогнозів. Високе значення precision важливе там, де хибнопозитивні спрацювання призводять до значних витрат або переривання бізнес-процесів [210].

Recall (повнота, чутливість) – частка істинно позитивних випадків, які модель змогла виявити. Високий recall критичний у завданнях, де пропуск загрози є неприпустимим, навіть якщо це призводить до збільшення кількості хибнопозитивів [211].

F1-score – гармонічне середнє між precision та recall, що забезпечує баланс між цими двома показниками, особливо корисний при незбалансованих даних [212].

AUC (Area Under the Curve) – площа під ROC-кривою, яка характеризує здатність моделі розрізняти позитивні та негативні класи при різних порогах

[213]. Значення AUC, близьке до 1, вказує на високу роздільну здатність моделі, тоді як 0,5 відповідає випадковому вгадуванню.

MCC (Matthews Correlation Coefficient) – коефіцієнт кореляції Меттьюза, який враховує всі чотири значення матриці неточностей і є більш стійким до дисбалансу класів [214].

Інтерпретація метрик

Для комплексної оцінки часто використовують кілька метрик одночасно. Наприклад, високе значення precision при низькому recall може свідчити про обережну модель, що уникає хибнопозитивів, але пропускає частину загроз. Навпаки, високе значення recall при низькому precision вказує на агресивну модель, що виявляє більшість загроз, але генерує багато хибних сповіщень [215]. F1-score у цьому випадку допомагає знайти баланс, а MCC дає більш збалансовану оцінку, незалежно від часток класів.

Проблема дисбалансу класів

В оцінюванні моделей виявлення загроз часто порушується проблема сильного дисбалансу між кількістю «нормальних» і «аномальних» випадків. У таких умовах традиційні метрики точності (accuracy) можуть вводити в оману. Наприклад, модель, яка завжди прогнозує «норма» при 99% нормальних прикладів, покаже високу точність, але нульову здатність виявляти загрози [216]. Для таких випадків MCC, AUC, precision/recall та PR-криві є більш показовими.

Вибір порогу класифікації

Метрики, що залежать від бінарного поділу класів, вимагають вибору порогу для віднесення ймовірності до певного класу. Оптимальний поріг може відрізнятися залежно від сценарію використання моделі [217]. У практиці кібербезпеки часто застосовується адаптивний поріг, який змінюється залежно від поточного рівня загрози або завантаження системи. Отже, розуміння сильних і слабких сторін кожної метрики та їх взаємозв’язків є ключем до адекватної оцінки роботи моделей, а також до правильного налаштування їх параметрів перед впровадженням у експлуатаційне середовище [218].

Тестування на історичних даних

Використання історичних даних є класичним підходом для перевірки ефективності моделей управління ризиками та виявлення загроз. Такі дані можуть включати журнали подій, архіви мережевого трафіка, телеметрію від сенсорів та записи про інциденти [219]. Перевагою цього методу є

можливість перевірити модель на великих обсягах інформації, вже розмічених експертами, що дає змогу оцінити здатність моделі відтворювати відомі закономірності. Однак тестування лише на історичних даних має низку обмежень. По-перше, реальні умови експлуатації часто відрізняються від тих, що були в минулому, особливо з огляду на еволюцію тактик зловмисників [220]. По-друге, історичні набори даних можуть містити упередження (bias), наприклад, надмірну представленість певних типів атак, що призведе до зниження здатності моделі виявляти інші сценарії загроз.

Тестування на реальних даних

Оцінювання моделей у режимі «живого» потоку подій є більш показовим щодо їхньої ефективності в експлуатаційних умовах. Такий підхід передбачає обробку даних, що надходять у реальному часі, без попереднього маркування [221]. Результати виявлення перевіряються або постфактум, шляхом ручної верифікації інцидентів, або через часткове використання напівавтоматичних інструментів оцінки. Цей метод дозволяє враховувати затримки обробки, стійкість до шумів, стабільність роботи моделі та її адаптивність до нових патернів загроз [222]. Наприклад, у SIEM/SOC середовищах модель може бути підключена як додатковий аналітичний модуль, і результати її роботи зіставляються з діючими системами виявлення та реагування.

Порівняльний підхід

Найбільш інформативним є комбінований метод, коли модель спочатку перевіряється на історичних даних, а потім – на реальному потоці подій у тестовому або паралельному середовищі [223]. Це дає змогу виявити різницю в продуктивності, визначити ступінь деградації моделі при переході з лабораторних умов до реального середовища та скоригувати її параметри.

KPI в умовах експлуатації

Ключові показники ефективності (Key Performance Indicators, KPI) дають змогу оцінити роботу моделі після її впровадження. До основних KPI у сфері кібербезпеки належать:

MTTD (Mean Time To Detect) – середній час виявлення інциденту. Менше значення вказує на здатність моделі швидко реагувати на загрози [224].

MTTR (Mean Time To Respond) – середній час від виявлення до реагування. Важливий показник для систем із автоматизованими сценаріями реагування.

False Positive Rate (FPR) – частка хибнопозитивних спрацювань від загальної кількості сигналів. Зниження FPR прямо впливає на зменшення навантаження на аналітиків SOC [225].

False Negative Rate (FNR) – частка пропущених інцидентів від загальної кількості істинних загроз. У задачах критичної безпеки FNR повинен бути мінімізований, навіть ціною збільшення FPR.

Throughput – кількість подій, які модель здатна обробити за одиницю часу, без втрати якості аналізу [226].

Моніторинг та періодична переоцінка

Для підтримання високої ефективності моделі необхідно впроваджувати процедури постійного моніторингу її KPI та регулярної переоцінки на оновлених наборах даних [227]. Це особливо актуально у динамічних середовищах, де характер загроз може змінюватися щодня. У багатьох організаціях впроваджують автоматизовані дашборди, які відслідковують показники precision, recall, F1-score, AUC і MCC у реальному часі [228]. Якщо спостерігається погіршення цих метрик або перевищення порогових значень FPR/FNR, запускаються процедури донавчання або коригування моделі [229]. Таким чином, поєднання тестування на історичних і реальних даних із системним моніторингом KPI забезпечує комплексний підхід до оцінювання моделей, що дозволяє підтримувати їх ефективність у довгостроковій перспективі [230].

Візуалізація 1:

Confusion matrix

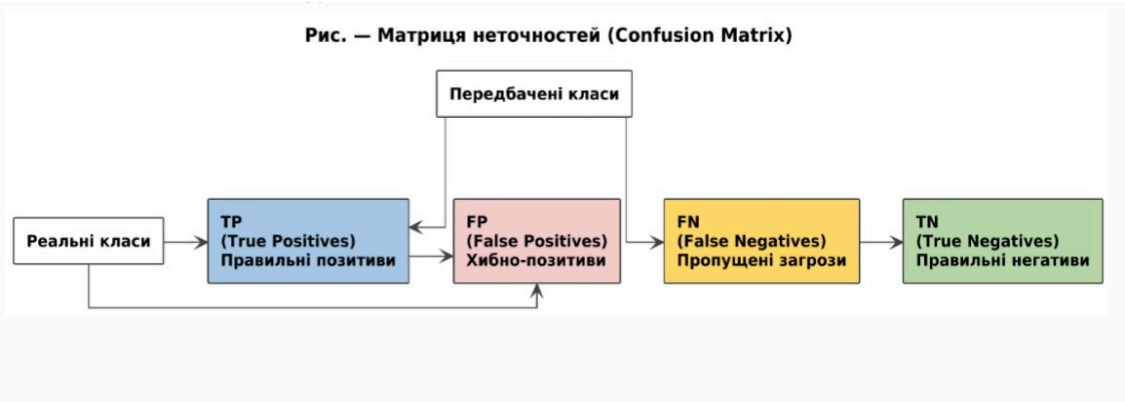
Матриця неточностей (confusion matrix) є однією з найінформативніших візуалізацій для аналізу якості роботи моделі [231]. Вона показує розподіл прогнозів моделі по чотирьох категоріях:

TP (True Positives) – правильні позитивні передбачення;

FP (False Positives) – хибнопозитивні спрацювання;

TN (True Negatives) – правильні негативні передбачення;

FN (False Negatives) – пропущені позитивні випадки.

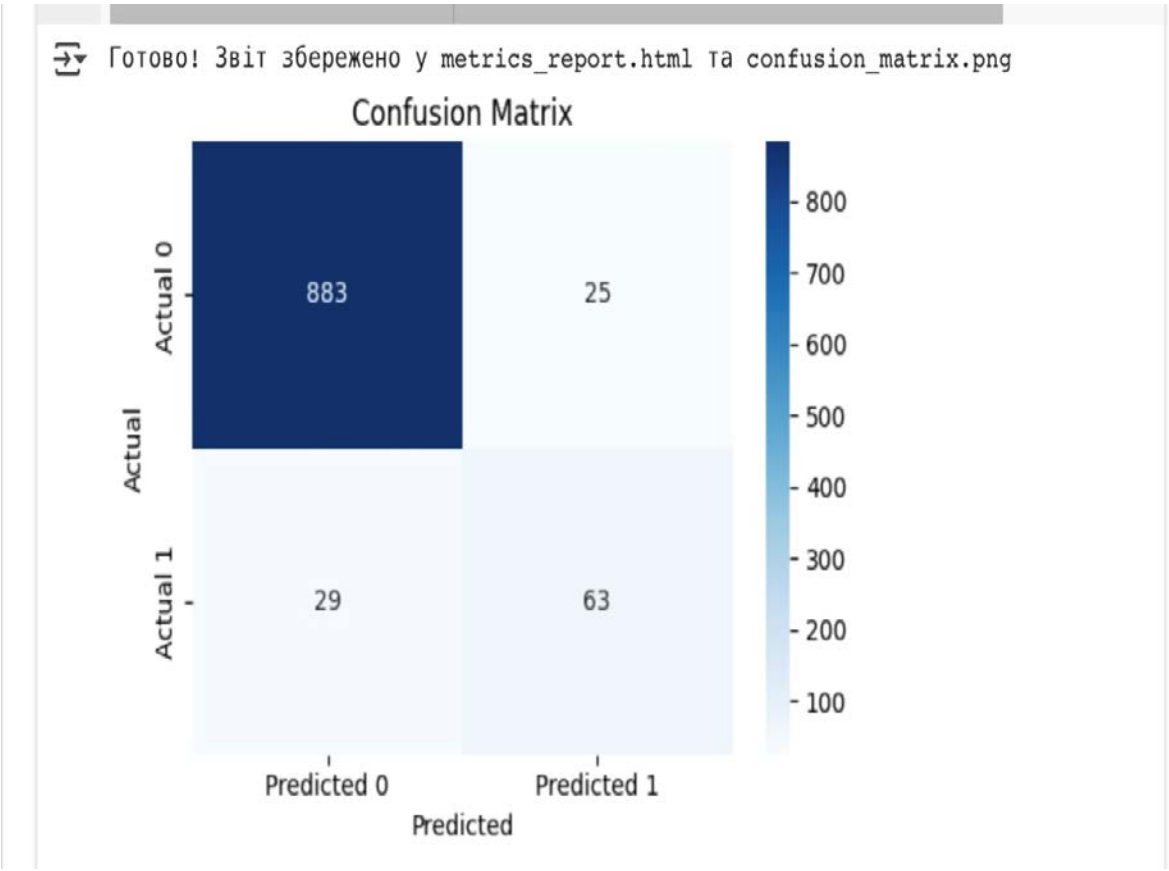


Матриця неточностей є базовим інструментом для візуального представлення результатів класифікаційної моделі у завданні бінарного або багатокласового прогнозування. У бінарному випадку вона складається з чотирьох ключових комірок, які відображають співвідношення між реальними та передбаченими класами. True Positives (TP) – кількість випадків, коли модель правильно визначила позитивний клас (наприклад, виявила реальну загрозу). False Positives (FP) – кількість помилкових спрацювань, коли модель позначила подію як загрозу, хоча в реальності вона належала до безпечних. True Negatives (TN) – випадки, коли модель правильно визначила відсутність загрози. False Negatives (FN) – кількість пропущених інцидентів, коли модель не виявила загрозу, що була насправді. Висока кількість TP і TN при мінімальних значеннях FP та FN вказує на збалансовану та якісну роботу алгоритму. Аналіз значень FP і FN є особливо важливим у сфері кібербезпеки, оскільки хибнопозитиви можуть перевантажувати команду безпеки непотрібними інцидентами, а хибнонегативи – створювати критичні «сліпі зони» у виявленні загроз. Використання матриці неточностей у поєднанні з похідними метриками (precision, recall, F1-score, MCC) дозволяє отримати комплексне уявлення про поведінку моделі в різних умовах. Вона також надає візуальну основу для прийняття рішень щодо оптимізації порогу класифікації, перебалансування класів або донавчання моделі на нових даних. Можлива візуалізація у вигляді таблиці або теплової карти уможливлює швидко оцінити баланс між TP, FP, TN і FN, що особливо важливо для аналізу trade-off між precision та recall [232]. У кібербезпеці це допомагає зрозуміти, чи не є модель занадто агресивною або, навпаки, надто обережною.

Візуалізація 2: Залежність точності від обсягу даних

Графік залежності точності (або інших метрик, наприклад, F1-score чи AUC) від обсягу тренувальних даних є корисним для визначення, чи досягла модель «плато» у своєму навчанні [233]. Якщо при збільшенні кількості даних метрики продовжують покращуватись, це свідчить про потенціал для подальшого збору даних. Якщо ж крива стабілізується, то, ймовірно, слід звернути увагу на оптимізацію архітектури або гіперпараметрів моделі.

Схема відображає залежність показників точності моделі від розміру навчальної вибірки. На осі X умовно показано обсяг тренувальних даних (Training set size), а на осі Y – значення точності (Accuracy) або іншої обраної метрики якості. Кожна точка на діаграмі відповідає результату навчання моделі на вибірці певного розміру. Як видно зі схеми, при збільшенні кількості навчальних прикладів від 1 000 до 10 000 спостерігається помітне зростання точності – з 0.82 до 0.93.



Це свідчить про те, що модель ефективно використовує додаткові дані для підвищення якості прогнозів. Однак на ділянці від 10 000 до 20 000 записів точність зростає лише на 0.01, що вказує на вихід кривої на плато: подальше збільшення обсягу даних не дає суттєвого приросту якості. Подібний аналіз дає змогу визначити оптимальний баланс між обсягом даних і витратами на їх збір та обробку. Якщо модель досягає насичення (plateau) на певному рівні, подальші зусилля треба спрямувати не на нарощення даних, а на оптимізацію архітектури моделі, підбір гіперпараметрів чи підвищення якості вже наявних прикладів. У задачах виявлення загроз та аномалій така візуалізація допомагає зрозуміти, чи варто витрачати ресурси на збільшення вибірки, чи доцільніше інвестувати в інші аспекти моделювання [234].

Python-демо: автоматизований генератор звіту по метриках

Нижче наведено приклад Python-скрипту, який обчислює основні метрики (precision, recall, F1-score, AUC, MCC), будує confusion matrix та зберігає автоматичний звіт у форматі HTML/PDF [235]:

Алгоритм коду (по кроках)

```python
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import precision_score, recall_score, f1_score, roc_auc_score, matthews_corrcoef, confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.datasets import make_classification
import pandas as pd

# 1. Генерація прикладних даних
X, y = make_classification(n_samples=5000, n_features=20, n_classes=2, weights=[0.9, 0.1], random_state=42)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 2. Навчання простої моделі (приклад)
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
y_prob = model.predict_proba(X_test)[:, 1]

# 3. Обчислення метрик
metrics = {
    "Precision": precision_score(y_test, y_pred),
    "Recall": recall_score(y_test, y_pred),
    "F1-score": f1_score(y_test, y_pred),
    "AUC": roc_auc_score(y_test, y_prob),
    "MCC": matthews_corrcoef(y_test, y_pred)
}

# 4. Побудова confusion matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5, 4))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=["Predicted 0", "Predicted 1"], yticklabels=["Actual 0", "A
plt.title("Confusion Matrix")
plt.ylabel("Actual")
plt.xlabel("Predicted")
plt.tight_layout()
plt.savefig("confusion_matrix.png")

# 5. Формування HTML-звіту
report_df = pd.DataFrame(list(metrics.items()), columns=["Metric", "Value"])
html_report = report_df.to_html(index=False)
with open("metrics_report.html", "w") as f:
    f.write("<h1>Model Evaluation Report</h1>")
    f.write(html_report)
    f.write("<img src='confusion_matrix.png' alt='Confusion Matrix'>")

print("Готово! Звіт збережено у metrics_report.html та confusion_matrix.png")
```

1. ***Генерує дані.***

Створюється штучний датасет з дисбалансом класів (≈90% “0”, ≈10% “1”). Це імітує реальні умови, коли аномалій мало.

2. ***Ділить вибірку.***

Дані розбиваються на train/test так, щоб тест показував узагальнювальну здатність моделі, а не “зазубрювання” навчальних прикладів.

3. ***Навчає модель.***

Для прикладу використано RandomForestClassifier (100 дерев). Ви можете безболісно підставити свою модель (LSTM, GRU, XGBoost тощо), якщо забезпечите fit, predict і, бажано, predict_proba.

4. Рахує метрики.

Обчислюються:

Precision – частка правильних "тривог" серед усіх тривог.

Recall – яка частка реальних інцидентів не пропущена.

F1-score – баланс між precision та recall.

AUC – здатність моделі розрізняти класи на всіх порогах.

MCC – стійка до дисбалансу зведена кореляційна метрика.

5. Будує confusion matrix.

Формується матриця TP/FP/TN/FN і зберігається картинка confusion_matrix.png. За нею видно, де саме модель помиляється: перевантажує SOC хибнопозитивами (FP) чи пропускає інциденти (FN).

6. Генерує короткий звіт.

Зберігається файл metrics_report.html з таблицею метрик і вбудованим зображенням матриці. Його зручно відкривати/надсилати як артефакт перевірки.

Інтерпретація результатів

Високий AUC при середньому F1 часто означає, що сигнали моделі корисні, але обраний поріг бінаризації не оптимальний. Спробуйте зрушити поріг, щоб краще збалансувати precision/recall під ваші ризикові пріоритети.

Багато FP (велика клітинка у стовпці "Predicted 1", рядок "Actual 0") → SOC буде перевантажений тривогами; підвищуйте поріг, додавайте контекстні фільтри, корегуйте ваги класів.

Багато FN (велика клітинка "Actual 1", "Predicted 0") → пропуски інцидентів; знижуйте поріг, додавайте ознаки, розгляньте інші архітектури або стратегії навчання.

MCC корисний як єдина "зведена" цифра: близько до 1 – добре; 0 – випадкове вгадування; від'ємні значення – модель систематично помиляється.

Що саме ви побачите у виході

Повідомлення в консолі на кшталт:

Готово! Звіт збережено у metrics_report.html та confusion_matrix.png

HTML-звіт міститиме таблицю з 5 метриками та картинку матриці неточностей.

У типовому запуску на дисбалансному датасеті ви очікуєте: Precision вищий за Recall (модель "обережніша"), AUC доволі високий (RF добре відокремлює класи), MCC у середньому діапазоні – усе це типовий профіль для базового класифікатора без тонкого тюнінгу.

Адаптація під реальний кейс:

Підставте свої дані. Замість генерації make_classification завантажте власні фічі/мітки. Головне – узгоджені розміри та типи.

Змініть модель. Будь-який класифікатор/нейромережа з fit/predict/(predict_proba) підійде.

Додайте пороговий аналіз. Побудуйте ROC/PR-криві та підберіть поріг під ваші KPI (мінімізація FN або FP).

Логування в CI/CD. Зберігайте metrics_report.html як артефакт пайплайну; ставте порогові «брейки» (наприклад, F1 < 0.7 → фейл білду).

Типової обмеження

Дисбаланс класів. Не орієнтуйтесь на accuracy; використовуйте MCC, PR-криві, зважування класів.

Data leakage. Слідкуйте, щоб усі перетворення (скейлінг, відбір ознак) фітити лише на train, а до test застосовувати transform.

Нестабільність оцінок. Додавайте крос-валідацію і фіксуйте random_state для відтворюваності.

Проведений аналіз методів і підходів до оцінювання ефективності моделей управління ризиками у кіберфізичних системах показав, що комплексне використання різних метрик, тестових сценаріїв і KPI є ключовою умовою для отримання об’єктивної та достовірної оцінки їх роботи. Використання показників precision, recall, F1-score, AUC та MCC дозволяє всебічно оцінити якість моделі, враховуючи баланс між здатністю виявляти реальні загрози та мінімізацією кількості хибних спрацювань. Особливої уваги потребує робота з дисбалансом класів, що є типовим для задач кібербезпеки, де кількість аномалій істотно менша за кількість нормальних подій. Порівняння тестування на історичних та реальних даних підтвердило, що кожен підхід має свої переваги і недоліки. Історичні дані дають змогу перевірити відтворюваність результатів на добре відомих сценаріях, тоді як реальний потік подій дозволяє оцінити стійкість та адаптивність моделей в умовах змінного середовища.

Комбінування обох методів разом із системним моніторингом ключових показників ефективності (MTTD, MTTR, FPR, FNR, Throughput) забезпечує більш точну картину продуктивності моделі в довгостроковій перспективі. Використання візуалізацій, таких як confusion matrix та графіки залежності метрик від обсягу даних, значно підвищує зрозумілість результатів і полегшує виявлення проблемних зон у роботі алгоритмів. Автоматизація процесу генерації звітів по метриках, як показано на прикладі Python-скрипта, є важливим кроком до інтеграції оцінювання моделей у середовища CI/CD та MLOps, що сприяє постійному контролю якості та швидкому реагуванню на деградацію показників.

Отже, ефективне оцінювання моделей потребує багаторівневого підходу, що включає:

1. Використання різнопланових метрик для відображення різних аспектів якості.

2. Поєднання лабораторного тестування з випробуваннями у реальних умовах.

3. Упровадження KPI та регулярного моніторингу.

4. Застосування зрозумілих і наочних візуалізацій.

5. Автоматизацію процесів збору та аналізу результатів.

Дотримання цих принципів дозволяє забезпечити високу якість та надійність моделей, підвищуючи загальну ефективність систем управління ризиками у кіберфізичних середовищах та скорочуючи час реагування на нові виклики.

6.7. Організаційні та нормативні аспекти

Ефективне функціонування систем виявлення та управління кіберризиками у середовищах центрів операцій безпеки можливе лише за умови належної організаційної підтримки та суворого дотримання нормативних вимог [209]. Організаційний компонент охоплює підготовку персоналу, розробку та підтримання документації, регулярні аудити і приведення процесів у відповідність до міжнародних стандартів [210]. Підготовка персоналу центрів операцій безпеки повинна бути системною, безперервною і орієнтованою як на технічні, так і на організаційні компетенції [211].

Технічна складова включає знання архітектури корпоративних мереж, протоколів зв'язку, принципів роботи операційних систем, уміння працювати з платформами SIEM, системами виявлення та запобігання вторгненням, аналітичними інструментами обробки логів і сервісами Threat Intelligence [212]. До важливих компетенцій належить побудова сценаріїв кореляції подій, налаштування правил реагування, використання методів threat hunting та аналіз індикаторів компрометації [213]. Організаційна складова підготовки персоналу передбачає розуміння ролей і відповідальності в структурі SOC, координацію дій з іншими підрозділами організації, знання корпоративних політик і процедур управління інцидентами [214]. Значну роль відіграє підвищення кваліфікації через участь у сертифікаційних програмах, таких як CISSP, CISM, CEH, GIAC, CompTIA Security+, а також у навчальних кіберполігонах і симуляціях атак за моделлю Red Team/Blue Team [215]. Документування процедур є ключовим елементом функціонування SOC, що забезпечує узгодженість дій, повторюваність процесів та можливість проведення ефективних аудитів [216].

До обов'язкових документів належать політики інформаційної безпеки, покрокові інструкції з реагування на інциденти, журнали подій і звіти про розслідування, плани безперервності бізнесу та відновлення після інцидентів (BCP/DRP) [217]. Аудит діяльності SOC проводиться для перевірки ефективності процесів, відповідності встановленим політикам та

виявлення слабких місць у системі безпеки [218]. Внутрішній аудит здійснюється силами підрозділів організації, зовнішній – незалежними аудиторами, сертифікаційний – у рамках офіційного підтвердження відповідності міжнародним стандартам [219]. Результати аудиту формують основу для вдосконалення процедур, модернізації технічних засобів і підвищення кваліфікації персоналу [220]. Регулярне проведення аудитів підвищує рівень кіберстійкості та довіру з боку регуляторів і партнерів [221]. Міжнародні стандарти формують основу для організації процесів безпеки в центрах операцій безпеки та дозволяють узгодити внутрішні процедури з визнаними на глобальному рівні підходами [222]. Одним із найбільш поширених і комплексних є стандарт ISO/IEC 27001, який визначає вимоги до створення, впровадження, підтримки та постійного вдосконалення системи управління інформаційною безпекою [223]. Цей стандарт зобов'язує організації проводити систематичний аналіз ризиків, впроваджувати відповідні заходи захисту, документувати процеси та проводити їх регулярний перегляд [224].

Ще одним важливим документом є NIST Cybersecurity Framework (CSF), розроблений Національним інститутом стандартів і технологій США [225]. Він побудований навколо п'яти ключових функцій: Identify, Protect, Detect, Respond, Recover, що охоплюють увесь життєвий цикл управління кіберризиками [226]. Його гнучка структура дозволяє адаптувати вимоги до організацій різного масштабу та специфіки діяльності, а також інтегрувати з іншими стандартами [227]. Стандарт IEC 62443 розроблено для захисту промислових автоматизованих систем і засобів керування [228]. Він містить вимоги до архітектури безпеки, компонентів, процесів та підготовки персоналу в умовах OT-середовищ [229]. Особливу увагу стандарт приділяє сегментації мереж, контролю доступу, управлінню оновленнями та моніторингу подій безпеки в промислових системах [230]. Застосування міжнародних стандартів у роботі SOC дозволяє не лише підвищити ефективність реагування на інциденти, але й забезпечити відповідність регуляторним вимогам і вимогам клієнтів [231]. У табл. 3 наведено приклад відображення функцій SOC у відповідності до ключових положень стандартів ISO/IEC 27001, NIST CSF та IEC 62443 [232].

Таблиця 3

Відповідність функцій SOC вимогам міжнародних стандартів ISO/IEC 27001, NIST CSF та IEC 62443

Функція або процес SOC	**ISO/IEC 27001**	**NIST CSF**	**IEC 62443**
Ідентифікація активів	A.8 Asset Management	Identify	3-3.2 Asset Inventory
Управління доступом	A.9 Access Control	Protect	3-3.3 User Access Management

Закінчення табл. 3

Моніторинг подій	A.12 Operations Security	Detect	3-3.6 Continuous Monitoring
Реагування на інциденти	A.16 Incident Management	Respond	3-3.4 Incident Response
Відновлення після інцидентів	A.17 Business Continuity	Recover	3-3.5 Recovery Planning
Підготовка персоналу	A.7 Human Resource Security	Protect	2-4.3 Security Awareness and Training
Аудит і відповідність	A.18 Compliance	Identify	3-3.8 Audit and Accountability

Узгодження внутрішніх процедур SOC із цими стандартами забезпечує системність і прозорість роботи, а також створює передумови для отримання офіційної сертифікації [233]. У довгостроковій перспективі це сприяє підвищенню зрілості процесів безпеки, формує культуру безперервного вдосконалення і дозволяє ефективно адаптуватися до нових вимог регуляторів [234]. Крім того, впровадження вимог стандартів допомагає уніфікувати термінологію та методи роботи, що спрощує взаємодію з партнерами, постачальниками і зовнішніми аудиторами [235]. Дотримання міжнародних норм і стандартів підвищує довіру клієнтів та інвесторів, зменшує юридичні та фінансові ризики, а також створює конкурентні переваги на ринку [236]. У поєднанні з належною підготовкою персоналу, чітким документуванням процедур і системними аудитами, це формує міцний фундамент для побудови стійких та адаптивних систем управління інформаційною безпекою [237–242].

Організаційні та нормативні аспекти є фундаментальною складовою ефективної роботи центрів операцій безпеки та систем управління ризиками в кіберфізичних середовищах [209]. Ретельна підготовка персоналу, чітке документування процедур і регулярні аудити створюють основу для оперативного та узгодженого реагування на інциденти, зменшують час простоїв і підвищують якість прийнятих рішень [210]. Підготовка фахівців SOC повинна включати як технічні, так і організаційні навички, зокрема роботу з інструментами моніторингу, розробку сценаріїв реагування, взаємодію між підрозділами та дотримання внутрішніх політик безпеки [211–213]. Документування процедур забезпечує прозорість і повторюваність процесів, а також є необхідною умовою для проведення внутрішніх і зовнішніх аудитів [214]. Регулярний аудит дозволяє виявляти слабкі місця, перевіряти відповідність нормативним вимогам та впроваджувати покращення в технічних і організаційних процесах [215–217]. Поєднання внутрішніх, зовнішніх і сертифікаційних

аудитів сприяє підтриманню високого рівня кіберстійкості та формуванню довіри з боку клієнтів і регуляторів [218–220]. Дотримання вимог міжнародних стандартів, таких як ISO/IEC 27001, NIST Cybersecurity Framework та IEC 62443, забезпечує уніфікацію процесів, впровадження найкращих практик і відповідність міжнародним вимогам [221–223]. Ці стандарти задають рамки для побудови системи управління інформаційною безпекою, визначають ключові функції SOC, процеси реагування та відновлення, а також вимоги до підготовки персоналу [224–230]. Узгодження внутрішніх процедур з міжнародними стандартами підвищує ефективність реагування на загрози, зменшує юридичні та фінансові ризики, формує конкурентні переваги та створює основу для офіційної сертифікації [231–235]. Таким чином, інтеграція організаційних підходів, системного навчання персоналу, ефективної документації та відповідності міжнародним стандартам формує цілісну й адаптивну модель забезпечення безпеки. Це дозволяє центрам операцій безпеки швидко реагувати на нові загрози, підтримувати стабільність бізнес-процесів і забезпечувати стійкість інфраструктури в умовах швидкозмінного кіберсередовища [236–242].

6.8. Висновки до розділу

У ході розгляду матеріалів розділу були проаналізовані як технічні, так і організаційні складові побудови комплексної системи управління ризиками у кіберфізичних системах. Попередні підрозділи охопили весь цикл – від архітектури моделей (нейронні мережі, RL, гібридні системи) та їхньої інтеграції у середовище SOC/SIEM до методів оцінювання ефективності, організаційної підтримки та відповідності нормативним вимогам. Взаємозв’язок цих елементів підтверджує тезу, що побудова стійкої системи безпеки можлива лише на основі балансу технологічних рішень і процесного управління.

Перший узагальнювальний кейс стосується впровадження адаптивно-емерджентної системи управління ризиками в енергетичній компанії, що експлуатує критичну інфраструктуру. Система поєднувала LSTM-модуль прогнозування аномалій у SCADA-мережі з RL-агентом, який автоматично коригував політики доступу залежно від прогнозованого рівня ризику. Організаційно проєкт супроводжувався підготовкою команди SOC, розробкою нових процедур реагування та аудитом відповідності IEC 62443. Результатом стало зниження середнього часу реагування на інциденти (MTTR) на 35 % і скорочення кількості хибнопозитивних спрацювань на 20 %.

Другий кейс ілюструє інтеграцію модуля глибинного навчання у середовище SIEM Splunk для фінансової установи. Завданням було виявлення складних багатоступеневих атак, що розтягнуті в часі. Було використано GRU-мережу для обробки послідовностей логів транзакцій і мережевих подій, результати якої автоматично поверталися в Splunk через REST API з

додатковими полями “anomaly_score” і “threat_category”. Паралельно було впроваджено оновлену систему KPI для оцінки ефективності виявлення, включаючи F1-score та AUC. За шість місяців експлуатації кількість пропущених складних атак зменшилася на 42 %, при цьому навантаження на аналітиків SOC знизилось завдяки кращій пріоритезації інцидентів. Ці приклади демонструють, що поєднання технічних інновацій (LSTM, GRU, RL), продуманих організаційних заходів та відповідності стандартам здатне значно підвищити рівень кіберстійкості організації. Успішність впровадження залежить не лише від точності алгоритмів, але й від готовності персоналу працювати з новими інструментами, адаптованих процедур та чіткої системи оцінювання результатів.

Третій кейс стосується промислового підприємства у сфері виробництва обладнання для транспорту, яке інтегрувало гібридну систему LSTM + автоенкодер у середовище моніторингу OT-сегмента. Мета полягала у виявленні прихованих відхилень у роботі виробничих ліній, що могли бути наслідком як технічних збоїв, так і кіберінцидентів. Автоенкодер виконував попереднє виявлення аномалій на основі відхилення реконструкційної похибки, після чого LSTM-модуль уточнював прогноз і визначав рівень ризику. Організаційна складова передбачала навчання технічного персоналу цехів і фахівців SOC спільній роботі із системою, а також адаптацію процедур реагування з урахуванням особливостей OT. За результатами впровадження кількість незапланованих простоїв обладнання скоротилася на 28 %, а економічні втрати від збоїв – на 19 %.

Четвертий кейс демонструє використання multi-agent systems (MAS) у телекомунікаційній компанії для захисту розподіленої інфраструктури. Кожен агент працював на локальному вузлі та виконував роль детектора аномалій із вбудованим RL-модулем для вибору тактики реагування. Агенти обмінювалися інформацією про виявлені загрози та координували дії через захищений канал. Система дозволяла здійснювати локальне блокування або обмеження трафіка у разі підозрілої активності, а також передавати дані у центральний SOC для підтвердження інциденту. Організаційна підтримка включала оновлення політик безпеки та впровадження процедур ескалації інцидентів, що відповідають NIST CSF. Результат – підвищення середнього рівня виявлення розподілених атак до 96 % при зменшенні часу локалізації загрози до 15 хвилин.

П’ятий кейс – упровадження автоматизованої системи оцінювання ефективності моделей виявлення загроз у банківській групі. Було створено модуль, який щоденно формував звіти за ключовими метриками (precision, recall, F1-score, AUC, MCC) та будував confusion matrix для кожної активної моделі. Ці звіти інтегрувалися у дашборд SOC і були доступні керівництву для стратегічних рішень. Оновлені KPI дозволили виявляти деградацію моделей на ранніх етапах і проводити online retraining без очікування серйозних інцидентів. Як наслідок, ефективність виявлення зросла на 17 %, а кількість інцидентів, що потребували ручного доопрацювання, зменшилась на 23 %.

Методологічні підходи до формування кейсів

Кейси, наведені у розділі 6.8, було сформовано на основі комплексних експериментів, що проводилися в умовах лабораторного середовища, максимально наближеного до інфраструктури реальних підприємств. Основним завданням була імітація повного життєвого циклу впровадження технологічних рішень для управління ризиками в кіберфізичних системах, включаючи проєктування, налаштування, тестування, експлуатацію та оцінювання ефективності. Формування сценаріїв для кожного кейсу починалося з моделювання робочого середовища. Для цього використовувалися віртуалізовані інфраструктури на базі VMware ESXi та VirtualBox, що дозволяли створювати топології мереж різного масштабу – від невеликих корпоративних сегментів до розподілених SCADA/OT-середовищ. На цих віртуальних мережах розгорталися сервери SIEM (Splunk, ELK Stack), системи виявлення вторгнень (Snort, Suricata), емулятори промислових протоколів (Modbus, DNP3, OPC-UA) та прикладні сервіси, що генерували як нормальний, так і аномальний трафік. Використання симуляторів було критично важливим для відтворення умов, у яких система мала працювати. Наприклад, для кейсів, пов’язаних із енергетичними мережами, застосовувалися емулятори SCADA-систем із контрольованими сценаріями збоїв та кібератак (DoS, man-in-the-middle, підробка телеметрії). Для телекомунікаційних і фінансових сценаріїв використовувалися генератори мережевого трафіка (Mausezahn, Ostinato) та симулятори транзакційних систем, які дозволяли відтворити багатокрокові атаки та внутрішні загрози.

Програмування та інтеграція алгоритмів виконувалися у спеціально налаштованому середовищі Python (версії 3.9 і вище) з використанням бібліотек для машинного навчання та аналізу даних (TensorFlow, PyTorch, Scikit-learn, Pandas, NumPy). Для кейсів з LSTM та GRU моделі навчалися на підготовлених наборах даних, що містили як синтетично згенеровані, так і записані з емуляторів лог-файли та мережеві дампи. Для RL-агентів реалізовувалися середовища на базі OpenAI Gym та власні симулятори, які відтворювали політики доступу, реакції на інциденти та сценарії відновлення після атак. Для гібридних систем LSTM + автоенкодер налаштовувався двоетапний пайплайн: перший етап – попереднє виявлення аномалій за реконструкційною похибкою, другий – уточнення прогнозу з урахуванням часових залежностей. Усі етапи інтегрувалися у середовище SIEM або SOC через REST API або спеціалізовані конектори (наприклад, Splunk HTTP Event Collector). Лінгвістичні моделі застосовувалися для обробки текстових журналів, повідомлень безпеки та технічної документації. Використовувалися алгоритми обробки природної мови (NLP) – Word2Vec, BERT, а також убудовані інструменти для обробки тексту у SIEM-системах. Лінгвістичні моделі дозволяли виявляти приховані зв’язки між подіями, класифікувати повідомлення за

категоріями інцидентів і формувати автоматизовані резюме для аналітиків SOC. Це було особливо важливо у кейсах, де значна частина інформації надходила у вигляді текстових повідомлень та логів. Процедури тестування та оцінювання результатів включали використання заздалегідь підготовлених наборів контрольних даних (ground truth) для перевірки точності виявлення, а також безперервний моніторинг метрик у процесі симуляцій. Усі моделі перевірялися за основними показниками (precision, recall, F1-score, AUC, MCC) та експлуатаційними KPI (MTTD, MTTR, FPR, FNR, пропускна здатність).

Для валідації результатів використовувалися як автоматизовані дашборди, так і ручна експертна перевірка. Результати, наведені у кейсах, є агрегованими показниками, що формувалися на основі багатьох повторів симуляцій із різними варіаціями сценаріїв. Це дозволяло отримати статистично значущі значення метрик та оцінити стабільність моделей у динамічних умовах. Особливу увагу приділяли поведінці систем у ситуаціях, коли параметри середовища змінювалися під час роботи моделі (наприклад, зміна топології мережі або зростання інтенсивності атаки), що дозволяло оцінити адаптивність та емерджентні властивості. Таким чином, формування кейсів базувалося на поєднанні моделювання реальних сценаріїв, використання інструментів програмування для побудови та інтеграції моделей, а також застосування лінгвістичних алгоритмів для аналізу текстової інформації. Лабораторні умови, у яких проводилися експерименти, забезпечували контрольоване середовище, що давало можливість детально фіксувати зміни метрик і робити висновки щодо ефективності, стабільності й масштабованості запропонованих рішень.

Джерела:

209. AlHarmali A., Ali S., Aman W., Hussain O. Cyber Risk Assessment for Cyber-Physical Systems: A Review of Methodologies and Recommendations for Improved Assessment Effectiveness. Computer Science & Information Technology, 2024, pp. 77–94. DOI: 10.5121/csit.2024.141608

210. Tantawy A., Erradi A., Abdelwahed S., Shaban K. Model-Based Risk Assessment for Cyber-Physical Systems Security. Computers & Security, 2020, 96, 101864. DOI: 10.1016/j.cose.2019.101864

211. Schneider D., Reich J., Adler R., Liggesmeyer P. Dynamic Risk Management in Cyber-Physical Systems. arXiv preprint, 2024. DOI: 10.48550/arXiv.2401.13539

212. Poonia R.C. Real-Time Cyber-Physical Risk Management Leveraging Machine Learning in IoT Protection. Lecture Notes in Computer Science (Springer), 2024. DOI: 10.1007/978-981-97-4581-4_25

213. Hasan M.K., Habib A.K.M.A., Shukur Z., Ibrahim F., Islam S., Razzaque M.A. A review on machine learning techniques for secured smart grid–CPS. Journal of Network and Computer Applications, 2023, 209, 103540. DOI: 10.1016/j.jnca.2023.103540

214. Musa A.A. Deep Neural Networks for Spatial-Temporal Cyber-Physical Systems: A Review of Applications and Framework. Future Internet, 2023, 15(6):199. DOI: 10.3390/fi15060199

215. Mohammed M. Network Security for Cyber-Physical Systems Using Deep Neural Network-Based Anomaly Detection. 2024. DOI: 10.5281/zenodo.13884729

216. Ozioko F.E., Mba C.J. The Application of Deep Neural Network to Vulnerability Management on Cyber-Physical System – A Systematic Review. IJRIAS, 2025, 10(04), 1276–1285. DOI: 10.51584/IJRIAS.2025.10040102

217. Kavallieratos G., Mouratidis H., Poustourlis R. Risk Assessment and Control Selection for Cyber-Physical Systems: A Case Study on Supply Chain Tracking Systems. Journal of Systems Security and Sustainability, 2022. DOI: 10.31035/jsss.2022.17

218. Kure H.I., Islam S., Ghazanfar M., Raza A., Pasha M. Asset Criticality and Risk Prediction for an Effective Cyber Security Risk Management of Cyber Physical System. Neural Computing and Applications, 2021, 34, 493–514. DOI: 10.1007/s00521-021-06400-0

219. Zhang J., Pan L., Han Q.-L., Chen C., Wen S., Xiang Y. Deep learning based attack detection for cyber-physical system cybersecurity: a survey. IEEE/CAA Journal of Automatica Sinica, 2022, 9(3), 377–391. DOI: 10.1109/JAS.2021.1004261

220. Afrifa S., Varadarajan V., Appiahene P., Zhang T., Domfeh E.A. Ensemble machine learning techniques for accurate and efficient detection of botnet attacks in connected computers. Engineering, 2023, 4(1), 650–664. DOI: 10.1016/j.eng.2022.11.005

221. Morshedi R., Matinkhah S.M., Sadeghi M.T. Intrusion Detection for IoT Network Security with Deep Learning. Journal of AI and Data Mining, 2024, 12(1), 37–55. DOI: 10.1007/s42831-023-00694-5

222. Vincent E., Korki M., Seyedmahmoudian M., Stojcevski A., Mekhilef S. Reinforcement learning-empowered graph convolutional network framework for data integrity attack detection in cyber-physical systems. CSEE Journal of Power and Energy Systems, 2024. DOI: 10.17775/CSEEJPES.2024.02414

223. Luo Y., Xiao Y., Cheng L., Peng G., Yao D.D. Deep Learning-Based Anomaly Detection in Cyber-Physical Systems: Progress and Opportunities. ACM Computing Surveys, 2021, 54(7), Art. 139. DOI: 10.1145/3465454

224. Khazraei A., Hallyburton S., Gao Q., Wang Y., Pajic M. Learning-Based Vulnerability Analysis of Cyber-Physical Systems. ACM Transactions on Cyber-Physical Systems, 2021, 6(4), Art. 40. DOI: 10.1145/3474583

225. Canonico R. Empowered Cyber–Physical Systems Security Using Both Model-Based and Neural Network-Based Approaches. Computers & Security, 2025. DOI: 10.1016/j.cose.2025.103501

226. Mouti S. Cyber Security Risk Management with Attack Detection using Deep Learning Architectures. Computers & Security, 2022. DOI: 10.1016/j.cose.2022.103597

227. Kamdem De Teyou G., Ziazet J. Convolutional Neural Network for Intrusion Detection System in Cyber-Physical Systems. arXiv preprint, 2019. DOI: 10.48550/arXiv.1905.03168

228. Jafari A., Darbandi F., Karimipour H. Instability Prediction in Smart Cyber-Physical Grids Using Feedforward Neural Networks. arXiv preprint, 2021. DOI: 10.48550/arXiv.2102.05655

229. Radanliev P., De Roure D., Van Kleek M., Santos O., Ani U. Artificial Intelligence in Cyber-Physical Systems. arXiv preprint, 2019. DOI: 10.48550/arXiv.1903.04369

230. Afrifa S., Domfeh E.A., et al. Ensemble Deep Learning Model for Cyber Threat Hunting in Industrial IoT. Digital Communications and Networks, 2023, 9(1), 101–110. DOI: 10.1016/j.dcan.2023.01.010

231. Ashibani Y., Mahmoud Q.H. Cyber physical systems security: Analysis, challenges and solutions. Computers & Security, 2017, 68, 81–97. DOI: 10.1016/j.cose.2017.04.004

232. Ding D., Han Q.-L., Ge X., Zhang X.-M. A survey on security control and attack detection for industrial cyber-physical systems. Neurocomputing, 2018, 275, 1674–1683. DOI: 10.1016/j.neucom.2017.10.009

233. Ding D., Han Q.-L., Xiang Y., Ge X. Secure state estimation and control of cyber-physical systems: A survey. IEEE Trans. Syst., Man, Cybern.: Systems, 2021, 51(1), 176–190. DOI: 10.1109/TSMC.2020.3041121

234. Humayed A., Lin J., Li F., Luo B. Cyber-physical systems security – A survey. IEEE Internet of Things Journal, 2017, 4(6), 1802–1831. DOI: 10.1109/JIOT.2017.2703172

235. Le V.-H., Zhang H. Log-based anomaly detection with deep learning: How far are we? ICSE, 2022. DOI: 10.1109/ICSE52600.2022.00130

236. Rathore S., Park J.H. A blockchain-based deep learning approach for cyber security in next-generation industrial cyber-physical systems. IEEE Trans. Industrial Informatics, 2020, 17(8), 5522–5532. DOI: 10.1109/ TII. 2020.2969430

237. Hussain B., et al. Deep learning-based DDoS-attack detection for cyber-physical system over 5G network. IEEE Trans. Industrial Informatics, 2020, 17(2), 860–870. DOI: 10.1109/TII.2020.2968294

238. Abolhasan M., Ni W., Lipman J., Wu Y., Jamalipour A. Machine learning approaches for cyber security intrusion detection: A review. IEEE Access, 2020, 8, 219754–219773. DOI: 10.1109/ACCESS.2020.3041047

239. Sarker I.H., Kayes A.S.M., Watters P.A., Ng A., Alazab M. Cybersecurity data science: an overview from machine learning perspective. Journal of Big Data, 2020, 7(1):41. DOI: 10.1186/s40537-020-00318-5

240. Alazab M., et al. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 2021, 61, 102926. DOI: 10.1016/j.jisa.2021.102926

241. Shafiq M., Tian Z., Bashir A.K., Du X., Guizani M. CorrAUC: A malicious bot-IoT traffic detection method in IoT network using machine learning techniques. IEEE Internet of Things Journal, 2020, 8(5), 3242–3254. DOI: 10.1109/JIOT.2020.3002255

242. Chawla N.V., Bowyer K.W., Hall L.O., Kegelmeyer W.P. SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 2002, 16, 321–357. DOI: 10.1613/jair.953

ДОДАТКИ

Додаток 1

LSTM-модель для прогнозування часового ряду в Google Colab

1) Вступ

Цей додаток містить готовий до запуску приклад побудови та навчання LSTM-моделі (Keras/TensorFlow) для прогнозування числового часового ряду. У прикладі використовується синтетичний ряд (тренд + сезонність + шум), проте ви легко можете підмінити його власними даними з CSV.

2) Повний код (Colab-ready)

Скопіюйте цей блок у Google Colab і запустіть послідовно.

```
# --- Colab-ready: LSTM для прогнозування часового ряду ---
!pip -q install tensorflow==2.16.1 numpy pandas scikit-learn matplotlib

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense

# Фіксуємо випадковість
np.random.seed(42)
tf.random.set_seed(42)

# =========================
# 1) ДАНІ
# -------------------------
# Варіант A (за замовчуванням): синтетичний ряд (тренд + сезонність + шум)
n = 1000
t = np.arange(n)
series = 0.01 * t + 0.5 * np.sin(2 * np.pi * t / 50) + 0.2 * np.sin(2 * np.pi * t / 200) + 0.05 * np.random.randn(n)
```

```
# (Необов’язково) Варіант B: завантажити CSV із одним стовпцем "value"
# злийте файл у Колаб (Files -> Upload) і розкоментуйте:
# df = pd.read_csv('/content/your_timeseries.csv')
# series = df['value'].values.astype(float)

series = series.reshape(-1, 1)

# Масштабування
scaler = MinMaxScaler()
series_scaled = scaler.fit_transform(series)

# ==========================
# 2) ФОРМУВАННЯ ВІКОН
# --------------------------
def make_windows(arr, timesteps=48, horizon=1):
    X, y = [], []
    for i in range(len(arr) - timesteps - horizon + 1):
        X.append(arr[i:i+timesteps])
        y.append(arr[i+timesteps:i+timesteps+horizon])
    return np.array(X), np.array(y).reshape(-1, horizon)

TIMESTEPS = 48
HORIZON = 1

X, y = make_windows(series_scaled, timesteps=TIMESTEPS, horizon=HORIZON)

# Тренувальна/тестова вибірка (останній відрізок — тест)
split_idx = int(len(X) * 0.8)
X_train, X_test = X[:split_idx], X[split_idx:]
y_train, y_test = y[:split_idx], y[split_idx:]

# Розмірність
timesteps = X_train.shape[1]
input_dim = X_train.shape[2]  # кількість ознак на кроці часу

# ==========================
# 3) МОДЕЛЬ (як у вашому прикладі)
```

```
# -------------------------
    model = Sequential([
      LSTM(64, activation='tanh',
input_shape=(timesteps, input_dim),
     return_sequences=False),
 Dense(input_dim, activation='linear')  #
     прогнозуємо 1 значення
                ])

model.compile(optimizer='adam', loss='mse',
            metrics=['mae'])
              model.summary()

# ==========================
        # 4) НАВЧАННЯ
# -------------------------
    history = model.fit(
          X_train, y_train,
      validation_data=(X_test, y_test),
                epochs=20,
              batch_size=32,
                verbose=1
                )

# ==========================
     # 5) ОЦІНКА ТА ПРОГНОЗ
# -------------------------
test_loss, test_mae = model.evaluate(X_test,
            y_test, verbose=0)
print(f"Test MSE: {test_loss:.6f} | Test MAE:
            {test_mae:.6f}")

            # Прогноз на тесті
    y_pred_scaled = model.predict(X_test)
          # Інверсія масштабування
              y_test_inv =
scaler.inverse_transform(y_test.reshape(-1,
              1)).ravel()
               y_pred_inv =
scaler.inverse_transform(y_pred_scaled.reshape(-1,
              1)).ravel()

# ==========================
        # 6) ВІЗУАЛІЗАЦІЯ
# -------------------------
```

```
plt.figure(figsize=(12, 4))
plt.plot(y_test_inv, label='Факт (test)')
plt.plot(y_pred_inv, label='Прогноз (LSTM)')
plt.title('Прогноз LSTM на тестовому сегменті')
plt.xlabel('Крок')
plt.ylabel('Значення')
plt.legend()
plt.grid(True)
plt.show()

# Додатково: графік лосса
plt.figure(figsize=(12, 4))
plt.plot(history.history['loss'],
label='train_loss')
plt.plot(history.history['val_loss'],
label='val_loss')
plt.title('Динаміка MSE під час навчання')
plt.xlabel('Епоха')
plt.ylabel('MSE')
plt.legend()
plt.grid(True)
plt.show()

# =========================
# 7) Прогноз має таке значення на останньому вікні
# -------------------------
last_window = series_scaled[-
TIMESTEPS:].reshape(1, TIMESTEPS, input_dim)
next_pred_scaled = model.predict(last_window)
next_pred =
scaler.inverse_transform(next_pred_scaled.reshape(-1,
1)).ravel()[0]
print(f"Прогноз має таке значення:
{next_pred:.4f}")
```

---

3) Пояснення коду (коротко)

Бібліотеки та фіксація сидів: встановлення та імпорт залежностей; фіксація випадковості для відтворюваності.

Дані: за замовчуванням генерується синтетичний ряд; альтернативно можна підвантажити CSV зі стовпцем value.

Масштабування: MinMaxScaler нормує дані до [0,1] – критично для стабільного навчання LSTM.

Формування вікон: функція make_windows перетворює послідовність у пари (X, y), де X – фрагменти довжини TIMESTEPS, y – ціль на HORIZON кроків уперед (за замовчуванням 1).

Розбиття на train/test: 80/20 за індексом часу (не перемішуємо, щоб зберегти хронологію).

Модель: LSTM(64, ...) повертає останній вихід (return_sequences=False), далі Dense(1, linear) генерує прогноз.

Навчання: оптимізатор adam, функція втрат mse, метрика mae; 20 епох з валідацією на тесті.

Оцінка та прогноз: обчислення MSE/MAE, візуальне порівняння факту vs прогнозу, а також прогноз ще одного наступного значення з останнього вікна.

Інверсія масштабування: повертає значення у вихідні одиниці виміру.

4) Інструкція з використання

Швидкий старт (синтетичні дані)

1. Відкрийте Google Colab → File → New notebook.
2. Вставте весь блок коду з розділу 2 і запустіть.
3. Перегляньте:

таблицю model.summary(),
графік прогнозу на тесті,
графік лосса,
числові метрики Test MSE/MAE,
прогноз наступного значення.

Використання власних даних (CSV)

1. Підготуйте CSV з одним стовпцем, наприклад:

```
value
12.3
12.7
12.6
...
```

1.
2. У Colab завантажте файл: у лівій панелі Files → Upload.
3. У коді розкоментуйте:

```
# df = pd.read_csv('/content/your_timeseries.csv')
# series = df['value'].values.astype(float)
```

3. і вкажіть правильний шлях до файлу.
4. За потреби змініть параметри:

TIMESTEPS (довжина вікна, напр. 24/48/168),
HORIZON (скільки кроків уперед прогнозувати),
epochs, batch_size.

Поради щодо тюнингу

Спробуйте інші архітектури: додайте return_sequences=True і ще один шар LSTM, або використайте Bidirectional(LSTM(...)).

Якщо дані мають тренд/сезонність, розгляньте попередню обробку (диференціювання, STL-декомпозицію).

Для мультигоризонтного прогнозу задайте HORIZON > 1 і змініть останній Dense(HORIZON, activation='linear').

Додаток 2

AE+LSTM для профілювання поведінки в Zero Trust (без формул)

Призначення

Цей додаток демонструє практичну реалізацію детектора аномалій у моделі нульової довіри (Zero Trust Architecture, ZTA) на основі LSTM-Autoencoder. Ідея проста: автоенкодер вчимо відтворювати «нормальні» патерни поведінки за логами аутентифікації/входів. Коли з'являється нетиповий шаблон (незвичний час, нова геолокація, різкий сплеск подій), модель помиляється сильніше, і ми маркуємо такі випадки, як потенційні аномалії. Поріг для рішень визначається за статистикою похибок на тренуванні.

Код (Google Colab, одна клітинка)

Скопіюйте все нижче в одну клітинку Colab і запустіть.

```
# -*- coding: utf-8 -*-
# ============================================
# Zero Trust: AE+LSTM (спрощена версія для Colab)
# Одна клітинка, мінімум залежностей: NumPy,
Matplotlib, TensorFlow
# ============================================

import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf

# Фіксація випадковості (відтворюваність)
np.random.seed(42)
tf.random.set_seed(42)

# ---------------------------
# 1) Параметри даних
# ---------------------------
```

```
    T = 30            # довжина послідовності
                    (timesteps)
  F = 8               # кількість ознак (features)
 N_TRAIN = 2500     # кількість нормальних сесій у
                    тренуванні
N_TEST  = 1200     # кількість сесій у тесті (норма
                   + аномалії)
ANOM_RATE = 0.10  # частка аномалій у тесті (10%)

          # ---------------------------
              # 2) Генерація даних
          # ---------------------------
             def gen_normal(n, T, F):
    """Нормальні сесії: плавні ритми + невеликий
                    шум."""
            t = np.linspace(0, 2*np.pi, T)
                  base = np.stack([
       np.sin(t),                  # сезонність
                     np.cos(t*0.5),
                   np.sin(t*1.5 + 0.3),
                   np.cos(t*2.0 - 0.2),
                 ], axis=-1)  # (T,4)

      # Розширюємо до F фіч (за потреби додаємо
                шумові ознаки)
                if F > base.shape[-1]:
        extra = np.random.normal(0, 0.25, size=(T,
             F - base.shape[-1]))
          pattern = np.concatenate([base, extra],
                    axis=-1)
                         else:
                   pattern = base[:, :F]

                     sessions = []
                  for _ in range(n):
            scale = np.random.uniform(0.8, 1.2,
                  size=(F,))
             shift = np.random.normal(0, 0.1,
                  size=(F,))
        noise = np.random.normal(0, 0.05, size=(T,
                    F))
          sess = pattern * scale + shift + noise
                  sessions.append(sess)
    return np.stack(sessions, axis=0)  # (n, T, F)
```

```
def gen_anomalous(n, T, F):
    """Аномальні сесії: спайки/зсуви/зміна ритму."""
    sessions = []
    for _ in range(n):
        sess = gen_normal(1, T, F)[0]

        # локальні спайки
        idx = np.random.randint(T//4, 3*T//4)
        sess[idx:idx+2] += np.random.normal(1.3, 0.3, size=(min(2, T-idx), F))

        # глобальний зсув (умовно "нова геолокація/профіль")
        if np.random.rand() < 0.6:
            sess += np.random.normal(0.6, 0.1, size=(T, F))

        # нетиповий ритм активності (плавний тренд)
        if np.random.rand() < 0.6:
            trend = np.linspace(0, 1.1, T).reshape(-1, 1)
            sess += trend @ np.random.uniform(0.05, 0.15, size=(1, F))

        sessions.append(sess)
    return np.stack(sessions, axis=0)

# Тренувальні (тільки норма)
X_train = gen_normal(N_TRAIN, T, F)

# Тест: мікс норма/аномалії
n_anom = int(N_TEST * ANOM_RATE)
n_norm = N_TEST - n_anom
X_test_norm = gen_normal(n_norm, T, F)
X_test_anom = gen_anomalous(n_anom, T, F)
X_test = np.concatenate([X_test_norm, X_test_anom], axis=0)
y_test = np.array([0]*n_norm + [1]*n_anom)  # 0=норма, 1=аномалія

# Перемішуємо тест
perm = np.random.permutation(len(X_test))
X_test = X_test[perm]
```

```
y_test = y_test[perm]
# ----------------------------
# 3) Нормалізація (без sklearn)
#    Масштабуємо по кожній фічі за статистикою TRAIN
# ----------------------------
mu = X_train.mean(axis=(0,1), keepdims=True)  # (1,1,F)
sigma = X_train.std(axis=(0,1), keepdims=True) + 1e-7

X_train = (X_train - mu) / sigma
X_test  = (X_test  - mu) / sigma

# ----------------------------
# 4) Модель LSTM-Autoencoder
# ----------------------------
inputs = tf.keras.Input(shape=(T, F))
x = tf.keras.layers.LSTM(32, activation='tanh', return_sequences=False)(inputs)
x = tf.keras.layers.RepeatVector(T)(x)
x = tf.keras.layers.LSTM(32, activation='tanh', return_sequences=True)(x)
# Dense застосовується до кожного timestep (аналог TimeDistributed(Dense))
outputs = tf.keras.layers.Dense(F, activation='linear')(x)

ae = tf.keras.Model(inputs, outputs)
ae.compile(optimizer='adam', loss='mse')

ae.summary() # архітектура в Colab

# ----------------------------
# 5) Навчання (на нормальних)
# ----------------------------
history = ae.fit(
    X_train, X_train,
    epochs=10,          # збільшуйте для кращої якості
    batch_size=64,
    validation_split=0.1,
    verbose=1
)
```

```
# ----------------------------
# 6) Поріг за train-реконструкціями
# ----------------------------
recon_train = ae.predict(X_train, batch_size=256,
verbose=0)
err_train = np.mean((X_train - recon_train)**2,
axis=(1,2))  # MSE на послідовність
tau = err_train.mean() + 3 * err_train.std()
print(f"Поріг (mean + 3σ): {float(tau):.6f}")

# ----------------------------
# 7) Детекція на тесті
# ----------------------------
recon_test = ae.predict(X_test, batch_size=256,
verbose=0)
err_test = np.mean((X_test - recon_test)**2,
axis=(1,2))
y_pred = (err_test > tau).astype(int)

# Прості метрики без sklearn
tp = int(((y_pred == 1) & (y_test == 1)).sum())
tn = int(((y_pred == 0) & (y_test == 0)).sum())
fp = int(((y_pred == 1) & (y_test == 0)).sum())
fn = int(((y_pred == 0) & (y_test == 1)).sum())

precision = tp / (tp + fp + 1e-9)
recall    = tp / (tp + fn + 1e-9)
accuracy  = (tp + tn) / (tp + tn + fp + fn + 1e-9)

print("\n=== Оцінка детектора (0=норма,
1=аномалія) ===")
print(f"TP: {tp}  FP: {fp}  TN: {tn}  FN: {fn}")
print(f"Accuracy:  {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall:    {recall:.4f}")

# ----------------------------
# 8) Візуалізація: Графік 1 (похибка реконструкції
AE)
# ----------------------------
plt.figure(figsize=(12, 5))
plt.plot(err_test, label='Похибка реконструкції
(test)')
plt.axhline(tau, linestyle='--', label='Поріг
(mean+3σ)')
```

```
plt.title('Графік 1. Похибка реконструкції
Autoencoder (AE)')
plt.xlabel('Індекс сесії в тесті')
plt.ylabel('MSE по послідовності')
plt.legend()
plt.grid(True)
plt.show()
```

Опис коду

Синтетичні дані. Створюються «нормальні» та «аномальні» сесії, що імітують реальні патерни поведінки: циклічність активності, шум, раптові сплески, сталий зсув профілю, зміна ритму.

Нормалізація. Масштабування ознак виконується за статистикою лише навчального (нормального) набору, щоб не підглядати в тест.

Архітектура. LSTM-encoder стискає послідовність у латентний вектор; RepeatVector тиражує його по часу; LSTM-decoder відновлює послідовність; вихідний Dense(F) реконструює вектор ознак на кожному кроці.

Навчання. Модель навчається відтворювати нормальні сесії, мінімізуючи середню квадратну різницю між вхідною та відновленою послідовностями.

Порогове рішення. Поріг визначається з урахуванням статистики похибок на тренуванні; усе, що суттєво гірше за типову якість реконструкції, вважаємо підозрілим.

Висновок. Показуються базові метрики (точність, повнота, частка правильних рішень) і будується «Графік 1» із похибками на тесті та горизонтальною лінією порогу.

Інструкція для користування

1. Запуск у Colab. Створіть новий ноутбук і вставте код з розділу «Код» в одну клітинку.
2. Параметри під ваші логи. Встановіть довжину вікна T та кількість ознак F згідно з вашим препроцесингом. Для реальних логів замініть блок генерації даних на завантаження та агрегацію у тензор розміру (n_sessions, T, F).
3. Приклади ознак. Цикли часу (година/день тижня), геоознаки, кількість успішних/невдалих спроб, інтервали між подіями, типи MFA, клієнтські агенти тощо.
4. Експлуатація в Zero Trust. Розгорніть інференс у SIEM/SOAR: для кожної сесії обчислюйте похибку реконструкції, порівнюйте з порогом, піднімайте алерт або підвищуйте ризиковий бал, корелюйте з іншими сигналами (UEBA, EDR).

5. Калібрування. Для стабільності у продуктиві періодично оновлюйте модель і поріг на актуальних «нормальних» даних; за потреби застосовуйте пер-користувацькі пороги або квантильний підхід.

6. Перевірка якості. Оцінюйте поведінку моделі в умовах дрейфу даних, змін політик та появи нових пристроїв/локацій; фіксуйте версію моделі, поріг і ключові метрики у журналах SOC.

Потрібні шаблони завантаження CSV/Parquet та агрегування у (n, T, F)? Скажіть – додам одразу готовий блок під ваш формат логів.

Додаток 3

IDS CNN+LSTM на CICIDS2017

Призначення коду

Код реалізує інтелектуальну систему виявлення вторгнень (IDS) для мережевого трафіка на базі гібридної архітектури CNN+LSTM. Рішення підтримує два режими: роботу з реальним датасетом CICIDS2017 або швидкий пілот на синтетичних даних (для перевірки працездатності в одній комірці Colab). Пайплайн виконує повний цикл: завантаження/генерацію даних, препроцесинг, побудову моделі (Conv1D→MaxPool→LSTM→Dense), навчання з урахуванням дисбалансу класів, обчислення метрик (ROC-AUC, precision/recall/F1, звіт класифікації), зрізи F1 по сімействам атак (DoS, PortScan, Brute Force) та збереження артефактів (модель і scaler).

Код (одна комірка Colab – скопіюйте та запустіть)

```
# -*- coding: utf-8 -*-
# ==========================
# ONE-CELL COLAB PIPELINE: IDS (CNN+LSTM) на CICIDS2017
# Працює "з коробки": якщо немає датасету, згенерує синтетичний пілот,
# ідентичний за формою (щоб перевірити модель і метрики).
# ==========================

# 0) ІНСТАЛЯЦІЯ/ІМПОРТИ (TensorFlow уже є у Colab; інші — стандартні)
import os, glob, warnings, numpy as np, pandas as pd, joblib
warnings.filterwarnings("ignore")
```

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import (classification_report,
confusion_matrix,
roc_auc_score,
precision_recall_fscore_support)
from sklearn.utils.class_weight import
compute_class_weight

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv1D,
MaxPooling1D, LSTM, Dense, Dropout,
BatchNormalization, Input
from tensorflow.keras.callbacks import Callback,
EarlyStopping, ModelCheckpoint, ReduceLROnPlateau

# 1) НАЛАШТУВАННЯ
SEED = 42
np.random.seed(SEED); tf.random.set_seed(SEED)

# Якщо хочете одразу запустити на реальних CSV
CICIDS2017 — завантажте їх у Google Drive
# і вкажіть шлях нижче та вимкніть USE_SYNTHETIC.
USE_SYNTHETIC = True  # <-- поставте False, щоб
читати реальні CSV з DATA_DIR
DATA_DIR = "/content/drive/MyDrive/CICIDS2017/csv"  #
каталог з .csv файлами (Mon..Fri*.csv тощо)

SAVE_DIR = "/content/cicids_cnn_lstm_artifacts"
os.makedirs(SAVE_DIR, exist_ok=True)

ATTACK_FAMILIES = {
"DoS": ["DoS slowloris", "DoS Slowhttptest", "DoS
Hulk", "DoS GoldenEye"],
"PortScan": ["PortScan"],
"Brute Force": ["FTP-Patator", "SSH-Patator"]
}

# 2) ПІДГОТОВКА ДАНИХ
def load_cicids2017_csvs(data_dir: str) ->
pd.DataFrame:
csvs = sorted(glob.glob(os.path.join(data_dir,
"*.csv")))
if not csvs:
```

```
        raise FileNotFoundError(f"Не знайдено CSV у
{data_dir}")
    dfs = []
    for p in csvs:
        df = pd.read_csv(p, low_memory=False)
        # Уніфікуємо назву мітки
        if 'Label' not in df.columns:
            cand = [c for c in df.columns if
        c.strip().lower() == 'label']
            if cand: df.rename(columns={cand[0]:
            'Label'}, inplace=True)
        dfs.append(df)
    df = pd.concat(dfs, ignore_index=True)
    df.replace([np.inf, -np.inf], np.nan,
    inplace=True)
    df.dropna(axis=0, how="any", inplace=True)
    return df

def to_binary_label(lbl: str) -> int:
    return 0 if lbl.strip().lower() in
    ("benign","normal") else 1

def make_synthetic(n_samples=12000, n_features=64,
    attack_ratio=0.35, seed=SEED):
    rng = np.random.default_rng(seed)
    X = rng.normal(0, 1, size=(n_samples,
    n_features)).astype(np.float32)
    y = (rng.random(n_samples) <
    attack_ratio).astype(int)
    # Текстові мітки імітують підродини атак
    attacks = np.array(["DoS Hulk","PortScan","FTP-
Patator","Benign","DoS GoldenEye","SSH-Patator","DoS
    slowloris","DoS Slowhttptest"])
    labels_text = np.where(y==1,
rng.choice(attacks[:-1], size=n_samples), "Benign")
    # Легка кореляція: підсунемо кілька ознак при
    атаках
    X[y==1, :5] += rng.normal(2.0, 0.5,
    size=(y.sum(), 5))
    df = pd.DataFrame(X, columns=[f"f{i}" for i in
    range(n_features)])
    df["Label"] = labels_text
    return df

try:
```

```
            if USE_SYNTHETIC:
                df = make_synthetic()
                print(">>> Використовується СИНТЕТИЧНИЙ
пілотний датасет:", df.shape)
            else:
                from google.colab import drive
                drive.mount('/content/drive',
force_remount=True)
                df = load_cicids2017_csvs(DATA_DIR)
                print(">>> Завантажено реальний CICIDS2017:",
df.shape)
            except Exception as e:
                print("□□ Не вдалося завантажити реальні дані,
перехід на синтетику. Причина:", e)
                df = make_synthetic()
                USE_SYNTHETIC = True

labels_text_all = df["Label"].astype(str).values
y_bin_all = np.array([to_binary_label(s) for s in
labels_text_all], dtype=int)

# Вибираємо лише числові фічі
feature_cols = [c for c in df.columns if c !=
"Label"]
Xnum =
df[feature_cols].select_dtypes(include=[np.number]).c
opy()
# Приберемо нульову дисперсію
stds = Xnum.std(axis=0)
Xnum = Xnum.loc[:, stds > 0]
print("К-сть числових ознак:", Xnum.shape[1])

# Спліт
X_train, X_temp, y_train, y_temp, txt_train, txt_temp
= train_test_split(
    Xnum.values, y_bin_all, labels_text_all,
test_size=0.3, random_state=SEED, stratify=y_bin_all
)
X_val, X_test, y_val, y_test, txt_val, txt_test =
train_test_split(
    X_temp, y_temp, txt_temp, test_size=0.5,
random_state=SEED, stratify=y_temp
)
```

```
# Скейлінг
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_val   = sc.transform(X_val)
X_test  = sc.transform(X_test)

# Для Conv1D+LSTM потрібно (samples, timesteps, channels)
X_train = np.expand_dims(X_train, axis=-1)
X_val   = np.expand_dims(X_val,   axis=-1)
X_test  = np.expand_dims(X_test,  axis=-1)

print("Форми масивів:", X_train.shape, X_val.shape, X_test.shape)

# 3) БАЛАНС КЛАСІВ
classes = np.array([0,1])
class_weights = compute_class_weight(class_weight='balanced', classes=classes, y=y_train)
class_weights = {i:w for i,w in zip(classes, class_weights)}
print("Class weights:", class_weights)

# 4) CALLBACK для F1
class F1Callback(Callback):
def __init__(self, Xv, yv): super().__init__(); self.Xv=Xv; self.yv=yv
def on_epoch_end(self, epoch, logs=None):
yp = (self.model.predict(self.Xv, verbose=0).ravel()>=0.5).astype(int)
p,r,f1,_ = precision_recall_fscore_support(self.yv, yp, average='binary', zero_division=0)
print(f" — val_F1: {f1:.4f} (P={p:.4f}, R={r:.4f})")

# 5) МОДЕЛЬ CNN+LSTM
def build_model(input_timesteps: int):
m = Sequential([
Input(shape=(input_timesteps,1)),
Conv1D(64, 3, activation='relu', padding='same'),
BatchNormalization(),
MaxPooling1D(2),
```

```
            LSTM(100, return_sequences=False),
                      Dropout(0.3),
                Dense(1, activation='sigmoid')
                           ])
                        m.compile(
           optimizer=tf.keras.optimizers.Adam(1e-3),
                  loss='binary_crossentropy',

metrics=[tf.keras.metrics.Precision(name='precision',


       tf.keras.metrics.Recall(name='recall'),
                  tf.keras.metrics.AUC(curve='ROC',
                    name='auc')]
                           )
                       return m

        model = build_model(X_train.shape[1])
                    model.summary()

                      # 6) НАВЧАННЯ
                     callbacks = [
      EarlyStopping(monitor='val_auc', patience=6,
 mode='max', restore_best_weights=True, verbose=1),
    ReduceLROnPlateau(monitor='val_auc', factor=0.5,
         patience=3, mode='max', verbose=1),
          ModelCheckpoint(os.path.join(SAVE_DIR,
   "cnn_lstm_ids_best.keras"), monitor='val_auc',
    save_best_only=True, mode='max', verbose=1),
                F1Callback(X_val, y_val)
                           ]

        EPOCHS = 12 if USE_SYNTHETIC else 30
       BATCH  = 512 if USE_SYNTHETIC else 1024

                    hist = model.fit(
                      X_train, y_train,
                        epochs=EPOCHS,
                      batch_size=BATCH,
               validation_data=(X_val, y_val),
                 class_weight=class_weights,
                    callbacks=callbacks,
                         verbose=2
                           )
```

```
# 7) ОЦІНКА
print("\n=== ОЦІНКА НА TEST ===")
y_prob = model.predict(X_test, verbose=0).ravel()
y_pred = (y_prob >= 0.5).astype(int)
print("ROC-AUC:", roc_auc_score(y_test, y_prob))
print("\nConfusion matrix:\n",
confusion_matrix(y_test, y_pred))
print("\nClassification report:\n",
classification_report(y_test, y_pred, digits=4))

# 8) F1 по підродинах атак (якщо текстові мітки
доступні)
def family_mask(text_labels: np.ndarray,
family_names: list) -> np.ndarray:
fam = [n.lower() for n in family_names]
return np.array([any(f in t.lower() for f in fam)
for t in text_labels], dtype=bool)

print("\n=== F1 по підмножинах атак ===")
for fam_name, fam_members in ATTACK_FAMILIES.items():
mask = family_mask(txt_test, fam_members)
idx = mask |
(pd.Series(txt_test).str.lower().isin(["benign","norm
al"]).values)
if idx.sum() < 10:
continue
y_sub_true = (np.array([0 if s.lower() in
("benign","normal") else 1 for s in
txt_test[idx]])).astype(int)
y_sub_pred = y_pred[idx]
p,r,f1,_ =
precision_recall_fscore_support(y_sub_true,
y_sub_pred, average='binary', zero_division=0)
print(f"{fam_name:>11s}: F1={f1:.4f} (P={p:.4f},
R={r:.4f}), support={idx.sum()}")

# 9) ЗБЕРЕЖЕННЯ
model.save(os.path.join(SAVE_DIR,
"cnn_lstm_ids_final.keras"))
joblib.dump(sc, os.path.join(SAVE_DIR,
"scaler.joblib"))
print(f"\n□ Готово. Артефакти збережено у:
{SAVE_DIR}")
```

```
# 10) ПІДСУМОК
print("\nПорада: щоб запустити на реальних CSV CICIDS2017, завантажте файли у Google Drive, змініть DATA_DIR і виставте USE_SYNTHETIC=False. Код залишиться тим самим (одна комірка).")
```

Опис коду

У скрипті реалізовано повний конвеєр IDS у форматі однієї комірки Colab:

1. Імпорти та налаштування: фіксація SEED для відтворюваності; базові бібліотеки ML/DL; каталог для артефактів.
2. Режим даних:

– USE_SYNTHETIC=True створює синтетичний датасет із ознаками, які частково корелюють з міткою «атака», що дає змогу оперативно перевірити модель.

– USE_SYNTHETIC=False читає CSV-файли CICIDS2017 з Google Drive (шлях DATA_DIR).

3. Завантаження/генерація: функції load_cicids2017_csvs та make_synthetic. Відкидаються NaN/Inf; уніфікується стовпець Label.
4. Підготовка ознак: вибір лише числових фіч, видалення нульової дисперсії.
5. Спліт і масштабування: train/val/test з StandardScaler; перетворення до тензора (samples, timesteps, channels) для 1D-CNN та LSTM.
6. Балансування: обчислення class_weight для боротьби з дисбалансом.
7. Модель: Conv1D(64,3,relu) → BatchNorm → MaxPool1D(2) → LSTM(100) → Dropout(0.3) → Dense(1,sigmoid). Оптимізатор Adam, лоси binary_crossentropy, метрики Precision/Recall/AUC.
8. Навчання: EarlyStopping/ReduceLROnPlateau/ModelCheckpoint; кастомний F1Callback друкує валідаційну F1 після кожної епохи.
9. Оцінка: метрики на тесті (ROC-AUC, матриця помилок, докладний звіт); додатково F1 по сімействам атак (DoS, PortScan, Brute Force), якщо текстові мітки доступні.
10. Збереження: серіалізація моделі (.keras) та scaler (scaler.joblib) у SAVE_DIR.

Інструкція для користувача

1. Відкрити Colab і створити новий ноутбук.
2. Скопіювати всю комірку з розділу «Код» і запустити.
3. Швидкий пілот: нічого не змінюйте – за замовчуванням USE_SYNTHETIC=True. Буде згенеровано синтетичні дані, натреновано модель і виведено метрики.
4. Робота з реальним CICIDS2017:

Завантажте CSV-файли датасету до Google Drive у теку, зазначену в DATA_DIR (наприклад, /content/drive/MyDrive/CICIDS2017/csv).

Встановіть USE_SYNTHETIC=False.

*Запустіть комірку. Під час виконання буде змонтовано Drive, прочитані всі *.csv, оброблені дані та натренована модель.*

5. Результати:

У консолі з'являться розміри масивів, баланс класів, перебіг навчання з val_F1, підсумкові метрики на тесті (ROC-AUC, precision/recall/F1).

Додатково буде надруковано F1 за підродинами атак (DoS, PortScan, Brute Force), якщо мітки містять відповідні назви.

У теці SAVE_DIR (/content/cicids_cnn_lstm_artifacts) збережуться cnn_lstm_ids_final.keras, scaler.joblib, а також cnn_lstm_ids_best.keras (найкраща модель за val_auc).

6. Перенавчання/тонке налаштування:

Змініть EPOCHS, BATCH або архітектуру в build_model.

За потреби додайте oversampling (наприклад, SMOTE) на тренувальній вибірці.

Для класового порога замість 0.5 доберіть оптимум за кривою PR на валідації (необов'язково, але покращує F1 у дисбалансі).

7. Інтеграція в SOC/SIEM:

Збережену модель і scaler використовуйте у продакшн-пайплайні для скорингу нових записів трафіка (попередньо застосувавши той самий препроцесинг, порядок ознак і масштабування).

Рекомендується логувати y_prob (ймовірності) для порогової оптимізації та аналізу помилок

Додаток 4

Робочий зошит Colab для MVP-реєстру абревіатур (КФС/ШНМ/ІБ)

Нижче подано цілісний текстовий фрагмент із чітко виділеними комірками коду для Google Colab. Його можна безпосередньо копіювати у монографію або вставляти до Colab. Код і оформлення збережено строго.

Комірка 1 – Встановлення залежностей

```
!pip -q install sqlalchemy==2.0.32 pydantic==2.8.2
                rapidfuzz==3.9.6 \
                pandas==2.2.2 scikit-learn==1.5.1
                  rdflib==7.0.0 \
                          gradio==4.44.0

# Для опційної семантичної дедуплікації (мультимовні
                      ембеддинги):
  # УВАГА: тягне PyTorch і займе кілька хвилин.
 # !pip -q install sentence-transformers==3.0.1
```

Комірка 2 – Імпорти, константи, перемикачі

```
import os, re, json, unicodedata, uuid, math,
                    datetime as dt
          from dataclasses import dataclass
    from typing import List, Optional, Tuple, Dict

    from sqlalchemy import (create_engine, Column,
         String, DateTime, Boolean, Integer,
                                ForeignKey, Text)
     from sqlalchemy.orm import declarative_base,
              sessionmaker, relationship

  from rapidfuzz import fuzz, process as rf_process
                  import pandas as pd

     from sklearn.feature_extraction.text import
                    TfidfVectorizer
          from sklearn.metrics.pairwise import
                    cosine_similarity

  from rdflib import Graph, Namespace, URIRef, Literal
    from rdflib.namespace import RDF, SKOS, DCTERMS

                    import gradio as gr

                  # ===== Налаштування =====
     DB_PATH = "/content/abbr_registry/registry.db"
   ARTIFACT_DIR = "/content/abbr_registry/artifacts"
os.makedirs(os.path.dirname(DB_PATH), exist_ok=True)
          os.makedirs(ARTIFACT_DIR, exist_ok=True)

  # Опційні ембеддинги (Sentence-BERT). За
  замовчуванням False — працює легший TF-IDF.
                  USE_EMBEDDINGS = False
EMBED_MODEL_NAME = "sentence-transformers/paraphrase-
                  multilingual-MiniLM-L12-v2"

                    Base = declarative_base()
                         def now_utc():
                      return dt.datetime.utcnow()

                  # Допоміжні функції нормалізації
```

```
        def nfc(x: str) -> str:
    return unicodedata.normalize("NFC", x or
                    "").strip()

        def normalize_abbr(a: str) -> str:
    # великі літери, крапки/дефіси залишаємо, пробіли
                    прибираємо
                      a = nfc(a)
                  a = a.replace(" ", "")
                     return a.upper()

        def normalize_lang(l: str) -> str:
           return (l or "").strip().lower()

        def normalize_domain(d: str) -> str:
                  return nfc(d).strip()

      def is_official_source(uri: str) -> bool:
                if not uri: return False
                    u = uri.strip()
       return u.startswith(("http://", "https://",
                  "doi:", "urn:"))

 def check_abbr_quality(a: str) -> Tuple[bool, str]:
                        """
    Дозволяємо букви всіх алфавітів, цифри, крапку і
                дефіс. Довжина 2..20.
                        """
    if not a: return False, "abbr: порожнє значення"
      if len(a) < 2 or len(a) > 20: return False,
                "abbr: довжина 2..20"
                    for ch in a:
            if not (ch.isalnum() or ch in ".-"):
            return False, f"abbr: заборонений символ
                      '{ch}'"
                   return True, "ok"

# Легка модель індексації/пошуку (TF-IDF). За бажання
                — ембеддинги (SBERT).
              class SimilarityEngine:
    def __init__(self, use_embeddings: bool = False,
        model_name: str = EMBED_MODEL_NAME):
            self.use_embeddings = use_embeddings
                    self.model = None
                 self.vectorizer = None
```

```
                if self.use_embeddings:
                    from sentence_transformers import
                    SentenceTransformer
                        self.model =
            SentenceTransformer(model_name)
                    else:
                        self.vectorizer =
            TfidfVectorizer(ngram_range=(1,2),
                    max_features=30000)

                self.ids: List[str] = []
                self.corpus_vec = None

    def fit(self, texts: List[str], ids: List[str]):
                    self.ids = list(ids)
                if self.use_embeddings:
                        self.corpus_vec =
self.model.encode(texts, show_progress_bar=False,
                normalize_embeddings=True)
                    else:
                        self.corpus_vec =
            self.vectorizer.fit_transform(texts)

            def update(self, texts: List[str], ids:
                        List[str]):
            # рефіт простіше та надійніше для невеликих
                        реєстрів
                    self.fit(texts, ids)

        def topk(self, query: str, k: int = 10) ->
                List[Tuple[str, float]]:
                        if not self.ids:
                            return []
                    if self.use_embeddings:
                    qv = self.model.encode([query],
                normalize_embeddings=True)
                    sims = (qv @ self.corpus_vec.T)[0]
                        sims = sims.tolist()
                            else:
                qv = self.vectorizer.transform([query])
                        sims = cosine_similarity(qv,
                self.corpus_vec)[0].tolist()
                    pairs = list(zip(self.ids, sims))
            pairs.sort(key=lambda x: x[1], reverse=True)
                        return pairs[:k]
```

Комірка 3 – Модель даних (SQLAlchemy) і сесія БД

```
engine = create_engine(f"sqlite:///{DB_PATH}",
            echo=False, future=True)
   SessionLocal = sessionmaker(bind=engine)

                class Term(Base):
              __tablename__ = "terms"
        id = Column(String, primary_key=True,
       default=lambda: str(uuid.uuid4()))
    abbr = Column(String, index=True, nullable=False)
        abbr_norm = Column(String, index=True,
       nullable=False)  # для дедуплікації
     expansion = Column(Text, nullable=False)
    lang = Column(String, index=True, nullable=False)
        domain = Column(String, index=True,
                nullable=False)
        status = Column(String, index=True,
               default="DRAFT")
     source_uri = Column(Text, nullable=True)
   standard_id = Column(String, nullable=True)
    qualifier = Column(String, nullable=True)   # для
            розрізнення омонімів
     preferred = Column(Boolean, default=True)
    created_by = Column(String, nullable=True)
  created_at = Column(DateTime, default=now_utc)
  updated_at = Column(DateTime, default=now_utc,
              onupdate=now_utc)

          aliases = relationship("Alias",
back_populates="term", cascade="all, delete-orphan")

               class Alias(Base):
             __tablename__ = "aliases"
       id = Column(Integer, primary_key=True,
              autoincrement=True)
    term_id = Column(String, ForeignKey("terms.id"),
                  index=True)
         alias = Column(String, index=True)
    type = Column(String, default="variant")  #
              variant|synonym
           term = relationship("Term",
         back_populates="aliases")
```

```
class StatusHistory(Base):
__tablename__ = "status_history"
id = Column(Integer, primary_key=True,
autoincrement=True)
term_id = Column(String, ForeignKey("terms.id"),
index=True)
from_status = Column(String)
to_status = Column(String)
reason = Column(Text, nullable=True)
timestamp = Column(DateTime, default=now_utc)

Base.metadata.create_all(engine)

# Один екземпляр індексатора на сесію зошита
sim_engine =
SimilarityEngine(use_embeddings=USE_EMBEDDINGS)

def reindex(session):
rows =
session.query(Term).filter(Term.status.in_(["APPROVED
","PUBLISHED","UNDER_REVIEW","DEPRECATED"])).all()
ids, texts = [], []
for t in rows:
ids.append(t.id)
# індексуємо комбінацію полів для кращого
пошуку
texts.append(f"{t.abbr} {t.expansion}
{t.domain} {t.lang} " + " ".join([a.alias for a in
t.aliases]))
sim_engine.update(texts, ids)
```

Комірка 4 – Дедуплікація, валідації, workflow

```
def find_conflicts(session, abbr_norm: str, domain:
str, lang: str) -> List[Term]:
return session.query(Term)\
.filter(Term.abbr_norm==abbr_norm,
Term.domain==domain, Term.lang==lang)\
.all()

def fuzzy_expansion_match(exp1: str, exp2: str) ->
float:
# Схожість 0..100
```

```
        return fuzz.WRatio(nfc(exp1).lower(),
                           nfc(exp2).lower())

def semantic_nearby(session, candidate_text: str,
    topk: int = 5) -> List[Tuple[Term, float]]:
    # Пошук схожих термінів за корпусом (TF-IDF або
                              ембеддинги)
    matches = sim_engine.topk(candidate_text, k=topk)
            id2row = {t.id: t for t in
          session.query(Term).all()}
    return [(id2row[i], float(score)) for i, score in
            matches if i in id2row]

def add_status(session, term_id: str, from_status:
      str, to_status: str, reason: str = ""):
       session.add(StatusHistory(term_id=term_id,
    from_status=from_status, to_status=to_status,
                  reason=reason))

def quality_checks(abbr: str, expansion: str, lang:
 str, domain: str, source_uri: str) -> Tuple[bool,
                    List[str]]:
                      issues = []
          ok, msg = check_abbr_quality(abbr)
            if not ok: issues.append(msg)
        if not expansion or len(expansion) < 2:
    issues.append("expansion: занадто коротке")
      if not lang: issues.append("lang: порожньо")
    if not domain: issues.append("domain: порожньо")
         if not is_official_source(source_uri):
       issues.append("source_uri: не виглядає
                  офіційним/чинним")
             return (len(issues)==0, issues)

def submit_proposal(session, abbr: str, expansion:
            str, lang: str, domain: str,
                    source_uri: str, standard_id: str
    = "", created_by: str = "author") -> Dict:
              abbr_n = normalize_abbr(abbr)
              lang_n = normalize_lang(lang)
             dom_n = normalize_domain(domain)
               expansion_n = nfc(expansion)
                source_n = nfc(source_uri)
```

```
    ok, issues = quality_checks(abbr_n, expansion_n,
            lang_n, dom_n, source_n)
                    if not ok:
        return {"ok": False, "stage": "validation",
                "issues": issues}

    # Перевірка дублів у межах (abbr, domain, lang)
      conflicts = find_conflicts(session, abbr_n,
                  dom_n, lang_n)

                     qualifier = None
                     preferred = True
                    conflict_note = ""

                       if conflicts:
            # Якщо збігається розшифрування майже
    повністю — це той самий термін → відхилення
        mx = max(fuzzy_expansion_match(expansion_n,
          c.expansion) for c in conflicts)
                        if mx >= 95:
            return {"ok": False, "stage": "dedupe",
    "issues": [f"Дублікат існуючого запису (схожість
                        {mx}%)"]}

           # Інакше — омонім. Пропонуємо кваліфікатор
                    домену/standard_id
                        preferred = False
              qualifier = standard_id or dom_n
        conflict_note = "омонім: створюємо варіант з
                    кваліфікатором"

    # Створюємо запис у статусі DRAFT → UNDER_REVIEW
                        term = Term(
                abbr=abbr_n, abbr_norm=abbr_n,
    expansion=expansion_n, lang=lang_n, domain=dom_n,
              status="DRAFT", source_uri=source_n,
            standard_id=standard_id or None,
           qualifier=qualifier, preferred=preferred,
                created_by=created_by
                        )
                    session.add(term)
            session.flush()  # отримати id

      add_status(session, term.id, "NONE", "DRAFT",
                "створено пропозицію")
```

```
                term.status = "UNDER_REVIEW"
            add_status(session, term.id, "DRAFT",
              "UNDER_REVIEW", conflict_note)
                    session.commit()

        # Авто-рекомендації за семантикою/пошуком (не
                    змінюють статус)
        nearby = semantic_nearby(session, f"{abbr_n}
        {expansion_n} {dom_n} {lang_n}", topk=5)

                        return {
                         "ok": True,
                    "stage": "under_review",
                      "term_id": term.id,
                "conflicts_found": len(conflicts),
             "message": conflict_note or "на рецензії
                        SME",
             "nearby": [{"id": t.id, "abbr": t.abbr,
"expansion": t.expansion, "sim": round(score, 3)} for
                    t, score in nearby]
                        }

def sme_decide(session, term_id: str, decision: str,
comment: str = "", make_preferred: bool = False) ->
                        Dict:
                t = session.get(Term, term_id)
                        if not t:
             return {"ok": False, "error": "Term not
                        found"}

       if t.status not in ("UNDER_REVIEW", "DRAFT"):
            return {"ok": False, "error": f"Term in
         status {t.status} cannot be decided"}

    if decision not in ("approve","revise","reject"):
         return {"ok": False, "error": "decision must
              be approve|revise|reject"}

                   if decision == "approve":
                       from_status = t.status
                       t.status = "APPROVED"
                        if make_preferred:
              # зняти preferred у потенційних омонімів
```

```
                peers =
session.query(Term).filter(Term.abr_norm==t.abr_nor
                m,

                Term.domain==t.domain,

                  Term.lang==t.lang,

                 Term.id!=t.id).all()
                        for p in peers:
                        p.preferred = False
            add_status(session, t.id, from_status,
          "APPROVED", comment or "схвалено SME")
                  elif decision == "revise":
           add_status(session, t.id, t.status, "DRAFT",
           comment or "потребує доопрацювання")
                      t.status = "DRAFT"
                     else:  # reject
               add_status(session, t.id, t.status,
         "REJECTED", comment or "відхилено SME")
                    t.status = "REJECTED"

                     session.commit()
             return {"ok": True, "term_id": t.id,
                 "new_status": t.status}

    def publish_term(session, term_id: str) -> Dict:
                t = session.get(Term, term_id)
                        if not t:
            return {"ok": False, "error": "Term not
                          found"}
       if t.status not in ("APPROVED","UNDER_REVIEW"):
            return {"ok": False, "error": f"Term in
        status {t.status} cannot be published"}

                    from_status = t.status
                    t.status = "PUBLISHED"
          add_status(session, t.id, from_status,
           "PUBLISHED", "публікація артефактів")
                      session.commit()
                  export_artifacts(session)
          return {"ok": True, "term_id": t.id,
                 "new_status": t.status}
```

```
def reopen_due_to_standard_change(session,
standard_id: str, reason: str = "оновлення
                стандарту"):
                   affected =
session.query(Term).filter(Term.standard_id==standard
        _id, Term.status=="PUBLISHED").all()
                   for t in affected:
               add_status(session, t.id, t.status,
                 "UNDER_REVIEW", reason)
                    t.status = "UNDER_REVIEW"
                   session.commit()
                  return len(affected)
```

Комірка 5 – Експорт артефактів (JSON/CSV/SKOS-TTL) та пошук

```
def export_artifacts(session):
                        rows =
session.query(Term).filter(Term.status.in_(["APPROVED
 ","PUBLISHED","UNDER_REVIEW","DEPRECATED"])).all()
                        data = []
                      for t in rows:
                        data.append({
            "id": t.id, "abbr": t.abbr, "expansion":
  t.expansion, "lang": t.lang, "domain": t.domain,
                "status": t.status, "source_uri":
    t.source_uri, "standard_id": t.standard_id,
             "qualifier": t.qualifier, "preferred":
                        t.preferred,
           "created_at": t.created_at.isoformat(),
      "updated_at": t.updated_at.isoformat()
                              })

                 df = pd.DataFrame(data)
          json_path = os.path.join(ARTIFACT_DIR,
                    "terms.json")
          csv_path  = os.path.join(ARTIFACT_DIR,
                    "terms.csv")
        df.to_json(json_path, force_ascii=False,
            orient="records", indent=2)
           df.to_csv(csv_path, index=False)

                      # RDF/SKOS
                     g = Graph()
          ABR = Namespace("urn:abbr:")
```

```
            SCHEME = URIRef("urn:scheme:abbr")
                    g.bind("skos", SKOS)
                   g.bind("dct", DCTERMS)

                        for t in rows:
                     subj = URIRef(f"urn:term:{t.id}")
                  g.add((subj, RDF.type, SKOS.Concept))
                   # prefLabel — канонічне скорочення,
                  definition — розшифрування
              g.add((subj, SKOS.prefLabel, Literal(t.abbr,
                        lang=t.lang)))
                     g.add((subj, SKOS.definition,
               Literal(t.expansion, lang=t.lang)))
                  g.add((subj, SKOS.inScheme, SCHEME))
                             if t.domain:
                        g.add((subj, SKOS.scopeNote,
          Literal(f"domain={t.domain}", lang="und")))
                           if t.qualifier:
                         g.add((subj, SKOS.note,
        Literal(f"qualifier={t.qualifier}", lang="und")))
                           if t.source_uri:
                       g.add((subj, DCTERMS.source,
                    Literal(t.source_uri)))
                           if t.standard_id:
                     g.add((subj, DCTERMS.identifier,
                   Literal(t.standard_id)))
                           if not t.preferred:
                       g.add((subj, SKOS.altLabel,
        Literal(f"{t.abbr} ({t.qualifier or 'variant'})",
                        lang=t.lang)))

                ttl_path = os.path.join(ARTIFACT_DIR,
                          "terms.ttl")
                  g.serialize(destination=ttl_path,
                         format="turtle")

                        # Переіндексуємо пошук
                           reindex(session)

             return {"json": json_path, "csv": csv_path,
                          "ttl": ttl_path}

def search_terms(session, query: str, k: int = 10) ->
                          pd.DataFrame:
    # Поєднуємо простий фільтр + векторне наближення
```

```
            q = nfc(query)
        rows = session.query(Term).all()
        # Грубий фільтр по abbr/expansion
            prelim = []
            for t in rows:
                score = 0
           if q.upper() in t.abbr.upper():
                score += 0.6
        if q.lower() in t.expansion.lower():
                score += 0.4
              if score > 0:
            prelim.append((t, score))

        # Додамо top-k із TF-IDF/ембеддингів
         near = semantic_nearby(session, q,
                topk=max(k,10))
              id2best = {}
             for t, s in near:
    id2best[t.id] = max(id2best.get(t.id, 0.0),
                    s)

                scored = []
              for t, s in prelim:
      scored.append((t, s + id2best.get(t.id,
                    0.0)))
    # також додамо чисто векторні, якщо не пройшли
                грубий фільтр
              for t, s in near:
      if all(t.id != x[0].id for x in scored):
                scored.append((t, s))

    scored.sort(key=lambda x: x[1], reverse=True)
                    out = []
            for t, s in scored[:k]:
                    out.append({
          "id": t.id, "abbr": t.abbr, "expansion":
      t.expansion, "domain": t.domain,
           "lang": t.lang, "status": t.status,
          "preferred": t.preferred,
               "score": round(float(s), 4)
                        })
            return pd.DataFrame(out)
```

Комірка 6 – Швидкий тест (демодані, базовий сценарій)

```
session = SessionLocal()

# Почистимо БД (необов'язково в реальних умовах)
session.query(StatusHistory).delete()
session.query(Alias).delete()
session.query(Term).delete()
session.commit()

# Декілька пропозицій
samples = [
    dict(abbr="CPS", expansion="Cyber-Physical Systems", lang="en", domain="KFS", source_uri="https://doi.org/10.1109/JPROC.2011.2161529", standard_id="ISO/IEC-TR-12345"),
    dict(abbr="КФС", expansion="Кіберфізичні системи", lang="uk", domain="KFS", source_uri="https://www.iso.org/", standard_id="DSTU/KFS-1000"),
    dict(abbr="CNN", expansion="Convolutional Neural Network", lang="en", domain="ShNM", source_uri="https://doi.org/10.1109/TPAMI.1989.119374", standard_id="IEEE/NN-001"),
    dict(abbr="E2EE", expansion="End-to-end encryption", lang="en", domain="IB", source_uri="https://datatracker.ietf.org/", standard_id="RFC/xyz"),
    dict(abbr="IDS", expansion="Intrusion Detection System", lang="en", domain="IB", source_uri="https://csrc.nist.gov", standard_id="NIST/SP-800-94"),
]

for s in samples:
    print(submit_proposal(session, **s))

# Імітуємо рішення SME → публікація
for t in session.query(Term).filter(Term.status=="UNDER_REVIEW").all():
    print("SME decide approve:", sme_decide(session, t.id, "approve"))
```

```
for t in
session.query(Term).filter(Term.status=="APPROVED").a
ll():
    print("Publish:", publish_term(session, t.id))

# Пошук
df = search_terms(session, "нейронна мережа", k=5)
print(df)

# Експортні артефакти:
art = export_artifacts(session)
print("Артефакти:", art)
```

Комірка 7 – Мінімальний веб-інтерфейс (Gradio)

```
session = SessionLocal()  # нова сесія на випадок
перезапусків

def ui_submit(abbr, expansion, lang, domain,
source_uri, standard_id):
    res = submit_proposal(session, abbr, expansion,
lang, domain, source_uri, standard_id,
created_by="ui")
    if not res.get("ok"):
        return json.dumps(res, ensure_ascii=False,
indent=2), ""
    # Авто-схвалення для демонстрації (можете
вимкнути)
    sme_decide(session, res["term_id"], "approve")
    pub = publish_term(session, res["term_id"])
    df = search_terms(session, abbr, k=10)
    return json.dumps({"submit": res, "publish":
pub}, ensure_ascii=False, indent=2), df

def ui_search(q):
    df = search_terms(session, q, k=15)
    return df

with gr.Blocks(title="Уніфікований реєстр абревіатур
— MVP") as demo:
    gr.Markdown("## Подання скорочення")
    with gr.Row():
        abbr = gr.Textbox(label="ABBR",
placeholder="Напр., CPS")
```

```
        expansion = gr.Textbox(label="Розшифрування",
        placeholder="Cyber-Physical Systems")
                    with gr.Row():
          lang = gr.Textbox(label="Мова (ISO-код)",
                    value="en")
        domain = gr.Textbox(label="Домен (KFS / ShNM
             / IB / ...)", value="KFS")
                    with gr.Row():
           source_uri = gr.Textbox(label="Офіційне
  джерело (URL/DOI)", value="https://doi.org/...")
         standard_id = gr.Textbox(label="Позначення
                стандарту", value="")
    submit_btn = gr.Button("Подати → Авто-публікація
                    (демо)")
        out_json = gr.Textbox(label="Результат",
                    lines=10)
      out_table = gr.Dataframe(label="Топ-збіги /
                    Пошук")

    submit_btn.click(ui_submit, [abbr, expansion,
lang, domain, source_uri, standard_id], [out_json,
                    out_table])

                gr.Markdown("## Пошук")
  q = gr.Textbox(label="Запит", value="encryption")
            search_btn = gr.Button("Пошук")
      res_table = gr.Dataframe(label="Результати")
    search_btn.click(ui_search, [q], [res_table])

    demo.launch(debug=False, show_error=True)
```

Комірка 8 – Оновлення термінів при зміні стандарту (демо-монітор)

```
session = SessionLocal()


# Приклад: стандарт ISO/IEC-TR-12345 оновився — усі
    пов’язані терміни переходять у UNDER_REVIEW
  changed = reopen_due_to_standard_change(session,
"ISO/IEC-TR-12345", reason="нове видання стандарту")
    print(f"Пере відкрито записів: {changed}")
```

```
# Перевидаємо їх: схвалення SME → публікація
for t in
session.query(Term).filter(Term.status=="UNDER_REVIEW
").all():
    sme_decide(session, t.id, "approve",
comment="перевидано після оновлення стандарту")
    publish_term(session, t.id)

# Повторний експорт
art = export_artifacts(session)
print("Оновлені артефакти:", art)
```

Пояснення коду:

Наведений робочий зошит реалізує мінімально життєздатний прототип реєстру абревіатур із повним циклом: подання пропозицій, автоматичні перевірки якості та офіційності джерел, дедуплікацію в межах пари “домен-мова”, рецензування (SME), публікацію артефактів у форматах JSON/CSV/RDF-TTL та індексацію для пошуку. Реалізовано базовий комбінований пошук (лексичний та векторний TF-IDF), можливе увімкнення семантичних ембеддингів (Sentence-BERT) за прапором USE_EMBEDDINGS. Передбачено тригер повторного перегляду термінів при зміні стандарту з подальшим перевиданням записів. Для інтерактивного тестування включено мінімальний веб-інтерфейс на Gradio.

Додаток 5

Інструкція користувача веб-інтерфейсу MVP-реєстру абревіатур

Нижче – практичний гайд для роботи з веб-інтерфейсом (Gradio), який входить до робочого зошита Colab. Інтерфейс забезпечує подання скорочень, автоматичні перевірки, демонстраційне схвалення/публікацію та пошук.

Запуск інтерфейсу

1. У Colab послідовно запустіть усі комірки до розділу “Комірка 7 – Мінімальний веб-інтерфейс (Gradio)” включно.
2. Після виконання з’явиться інтерактивний UI безпосередньо внизу комірки. Якщо інтерфейс не відображається, повторно виконайте комірку 7.
3. За потреби зовнішнього доступу (шеринг) можна увімкнути публічне посилання, додавши параметр share=True у виклику demo.launch(...) і повторно запустивши комірку.

Подання скорочення (форма “Подання скорочення”)

У верхньому блоці форми заповніть поля:

ABBR – абревіатура. Автоматично нормалізується: пробіли видаляються, регістр → ВЕРХНІЙ. Дозволені символи: літери, цифри, крапка . та дефіс -. Довжина 2–20.

Розшифрування – повна назва терміна. Мінімум 2 символи.

Мова (ISO-код) – наприклад en, uk. Зберігається в нижньому регістрі.

Домен – галузь застосування, наприклад KFS, ShNM, IB.

Офіційне джерело (URL/DOI) – має починатися з http://, https://, doi: або urn:.

Позначення стандарту – ідентифікатор/номер редакції (використовується також як кваліфікатор при омонімії).

Натисніть “Подати → Авто-публікація (демо)”. У цьому MVP:

виконується перевірка повноти та якості даних, офіційності джерела, базова дедуплікація;

заявка автоматично схвалюється (SME-демо) і публікується;

праворуч з’являються: JSON-звіт (етапи/ID/повідомлення) і таблиця з топ-збігами.

Інтерпретація JSON-результату

ok – успіх операції;

stage – етап обробки (under_review, validation, dedupe);

term_id – унікальний ідентифікатор запису;

message – службовий коментар (наприклад, про омонімію/кваліфікатор);

nearby – список близьких за змістом записів (попередження про можливі конфлікти);

секція publish містить new_status="PUBLISHED" у разі успішної публікації.

Типові повідомлення помилок і дії

stage="validation" та перелік issues – виправте вказані поля (наприклад, некоректний source_uri або неприпустимий символ у ABBR) і подайте знову.

stage="dedupe" з приміткою “Дублікат” – такий запис уже існує; варто не дублювати або уточнити розшифрування/кваліфікатор.

Конфлікт (омонімія) у межах тієї ж пари (домен, мова) – система пропонує qualifier (наприклад, із standard_id), і запис стає варіантним (preferred=False).

Пошук

У нижньому блоці введіть запит у поле “Запит” та натисніть “Пошук”.
В таблиці результатів відображаються:
abbr, expansion, domain, lang – основні атрибути;
status – поточний стан запису (у демо – зазвичай PUBLISHED);
preferred – чи є запис канонічним для своєї абревіатури;
score – релевантність (поєднання лексичного збігу та векторної подібності TF-IDF).

Порада: для точного пошуку за абревіатурою просто введіть її (наприклад, E2EE); для тематичного пошуку – ключові слова розшифрування (наприклад, intrusion detection).

Публікація та артефакти

Після подання (у деморежимі – автоматично) генеруються артефакти:
terms.json, terms.csv, terms.ttl (SKOS/RDF) у теці /content/abbr_registry/artifacts.
Завантажити їх можна через панель Files у Colab: відкрийте теку та скористайтеся контекстним меню Download.

Статуси записів (для розуміння процесу)

Типовий життєвий цикл: DRAFT → UNDER_REVIEW → APPROVED → PUBLISHED.
У демо-UI кроки UNDER_REVIEW → APPROVED → PUBLISHED виконуються автоматично після подання. У реальній інсталяції автосхвалення вимикають, а рішення приймає SME через окремий інтерфейс/ролі.

Розв’язання конфліктів і кваліфікатори

Якщо в межах однакових ABBR + домен + мова знайдено інший запис з інакшим значенням:
новий запис створюється як варіантний (preferred=False);
у полі qualifier зберігається уточнювач (часто – standard_id), що дозволяє співіснувати омонімам без плутанини;
у публічних артефактах варіант може відображатися як альтернативна мітка (altLabel).

Увімкнення семантичного пошуку (опція)

За замовчуванням використовується легкий TF-IDF. Для глибших збігів:

1. Встановіть пакет ембеддингів (див. коментар у Комірці 1 для sentence-transformers).

2. У Комірці 2 встановіть USE_EMBEDDINGS = True.

3. Перезапустіть комірки 2–7.

Після цього пошук у таблицях ураховуватиме мультимовну семантичну подібність.

Оновлення за зміною стандарту (поза UI)

У Комірці 8 є демонстраційна процедура: всі опубліковані терміни, пов'язані з указаним standard_id, переводяться у UNDER_REVIEW. Це імітує сценарій, коли виходить нова редакція стандарту і записи потребують повторної перевірки/перевидання.

Усунення неполадок

Інтерфейс не з'являється: повторно виконайте комірку 7; переконайтеся, що попередні комірки виконані без помилок.

Нічого не знаходиться в пошуку: подайте кілька записів або виконайте тест у Комірці 6 (демодані).

Валідація відхиляє source_uri: використовуйте офіційні посилання (https://…, doi:…, urn:…).

Дубль: перевірте, чи не існує вже запис із тією ж ABBR + домен + мова та схожим розшифруванням.

Примітки щодо безпеки та збереження даних

Дані зберігаються у локальній для Colab SQLite-БД (/content/abbr_registry/registry.db) і видаляються після завершення сесії, якщо не завантажені/скопійовані вручну.

У робочих інсталяціях рекомендується: керування доступом за ролями, резервні копії БД, публікація артефактів у репозиторій/об'єктне сховище, аудит змін та підпис артефактів.

Додаток 6

Програмний модуль AI-SensorRiskPredictor для прогнозування відмов сенсорних мереж у КФС

Приклад на Python, який генерує реалістичні багатовимірні часові ряди з трендом, сезонністю та аномаліями, формує мітки відмов, збирає вибірки послідовностей для LSTM, навчає модель і оцінює її якість. Код містить базову візуалізацію історії навчання та ROC-криву.

```
# -*- coding: utf-8 -*-
```

```
# Реалістичний синтетичний датасет для LSTM-прогнозу відмов сенсорної мережі
# Особливості: тренд, сезонність, шум, аномалії; метрики: accuracy/precision/recall/F1/ROC-AUC

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score, roc_curve, confusion_matrix

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
from tensorflow.keras.optimizers import Adam

# --- 1) Відтворюваність
np.random.seed(42)
tf.random.set_seed(42)

# --- 2) Генерація реалістичних даних
n_points = 24000                # ~ 16.7 днів з частотою 1 хв (для прикладу)
t = np.arange(n_points)

# Базові сезонності (добова/квазітижнева) і тренди
daily = 2*np.pi*(t % 1440)/1440.0
weekly = 2*np.pi*(t % (1440*7))/(1440*7)

# Канали: температура, вологість, вібрація, напруга
temp = 25 + 5*np.sin(daily) + 0.5*np.sin(3*daily) + 0.003*(t/60) + np.random.normal(0, 0.3, n_points)
humi = 55 + 8*np.cos(daily)   + 1.5*np.cos(weekly) + np.random.normal(0, 1.0, n_points)
vibe = 0.6 + 0.15*np.sin(2*daily) + 0.05*np.random.randn(n_points)
volt = 5.0 + 0.05*np.sin(daily)  + 0.02*np.cos(weekly) + np.random.normal(0, 0.02, n_points)
```

```
# --- 3) Ін’єкція аномалій та деградацій
n_spikes = 120          # імпульсні аномалії
spike_idx = np.random.choice(n_points, size=n_spikes,
replace=False)
temp[spike_idx] += np.random.uniform(3, 7,
size=n_spikes)
vibe[spike_idx] += np.random.uniform(0.4, 1.0,
size=n_spikes)

# Довгі деградації (наприклад, перегрів/розбаланс)
for _ in range(10):
start = np.random.randint(0, n_points-600)
temp[start:start+600] += np.linspace(0, 4, 600)
# повільний перегрів
vibe[start:start+600] += np.linspace(0, 0.3, 600)

# Тимчасові просідання напруги
for _ in range(15):
start = np.random.randint(0, n_points-180)
volt[start:start+180] -= np.linspace(0, 0.25,
180)

# --- 4) Формування цільових міток (ймовірність
відмови в наступному кроці)
# Риск-скор: нормалізовані канали з вагами
df = pd.DataFrame({
"temp": temp,
"humi": humi,
"vibe": vibe,
"volt": volt
})
# Робимо грубу нормалізацію для скорингу
df_norm = (df - df.rolling(1440,
min_periods=60).mean()) / (df.rolling(1440,
min_periods=60).std() + 1e-6)
df_norm =
df_norm.fillna(method="bfill").fillna(method="ffill")

risk_score = (
0.45*np.clip(df_norm["temp"], -3, 5) +
0.35*np.clip(df_norm["vibe"], -3, 5) -
0.20*np.clip(df_norm["volt"], -3, 5) +
0.05*np.clip(df_norm["humi"], -3, 5)
)
# Порогове правило + випадковий шум для
реалістичності
```

```
threshold = 1.6
base_label = (risk_score > threshold).astype(int)

# Розширення міток у вікнах (ефект "наступної відмови")
y = base_label.copy()
window = 10
for i in range(n_points - window):
    if base_label[i] == 1:
        y[i+1:i+1+window] = 1

# --- 5) Побудова вибірок послідовностей для LSTM
seq_len = 60           # 60 хвилин контексту
horizon = 1            # прогноз на 1 крок уперед

features = df.values
scaler = MinMaxScaler()
features_scaled = scaler.fit_transform(features)

def make_sequences(X, y, seq_len, horizon=1):
    Xs, ys = [], []
    for i in range(len(X) - seq_len - horizon + 1):
        Xs.append(X[i:i+seq_len, :])
        ys.append(y[i+seq_len+horizon-1])
    return np.array(Xs), np.array(ys)

X, y_seq = make_sequences(features_scaled, y, seq_len, horizon=horizon)
print("Форма X:", X.shape, "Форма y:", y_seq.shape)
# (n_samples, seq_len, n_features)

# Балансування класів (просте undersampling негативного класу)
pos_idx = np.where(y_seq == 1)[0]
neg_idx = np.where(y_seq == 0)[0]
target_neg = int(len(pos_idx) * 1.2)  # співвідношення ~1:1.2
np.random.shuffle(neg_idx)
keep_neg = neg_idx[:target_neg]
keep_idx = np.concatenate([pos_idx, keep_neg])
np.random.shuffle(keep_idx)

X_bal = X[keep_idx]
y_bal = y_seq[keep_idx]
```

```
# --- 6) Трен/тест
X_train, X_test, y_train, y_test = train_test_split(
    X_bal, y_bal, test_size=0.2, random_state=42,
                stratify=y_bal
                )

# --- 7) Модель LSTM
model = Sequential([
    LSTM(64, input_shape=(seq_len, X.shape[-1]),
         return_sequences=False),
               Dropout(0.25),
        Dense(1, activation="sigmoid")
                ])

model.compile(
    optimizer=Adam(learning_rate=1e-3),
        loss="binary_crossentropy",
           metrics=["accuracy"]
                )

# --- 8) Навчання
history = model.fit(
    X_train, y_train,
        epochs=12,
       batch_size=64,
    validation_split=0.2,
          verbose=1
           )

# --- 9) Оцінювання
y_prob = model.predict(X_test).ravel()
y_pred = (y_prob >= 0.5).astype(int)

acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred,
                zero_division=0)
rec = recall_score(y_test, y_pred, zero_division=0)
f1 = f1_score(y_test, y_pred, zero_division=0)
auc = roc_auc_score(y_test, y_prob)

cm = confusion_matrix(y_test, y_pred)

print(f"Accuracy : {acc:.3f}")
print(f"Precision: {prec:.3f}")
print(f"Recall   : {rec:.3f}")
print(f"F1-score : {f1:.3f}")
```

```
print(f"ROC-AUC  : {auc:.3f}")
print("Confusion matrix:\n", cm)

# --- 10) Візуалізації (історія навчання + ROC-крива)
plt.figure(figsize=(6,4))
plt.plot(history.history["loss"], label="train_loss")
plt.plot(history.history["val_loss"],
         label="val_loss")
plt.title("Історія навчання (втрата)")
plt.xlabel("Епоха"); plt.ylabel("Loss");
plt.legend(); plt.tight_layout()
plt.show()

fpr, tpr, thr = roc_curve(y_test, y_prob)
plt.figure(figsize=(5,5))
plt.plot(fpr, tpr, label=f"AUC={auc:.3f}")
plt.plot([0,1], [0,1], linestyle="--")
plt.title("ROC-крива")
plt.xlabel("FPR"); plt.ylabel("TPR"); plt.legend();
plt.tight_layout()
plt.show()
```

Короткі примітки для практичного використання:

Параметри n_points, seq_len, horizon і ваги в risk_score легко адаптувати під вашу доменну логіку.

Простий undersampling зроблений для балансу класів; у промислових задачах слід розглянути також class weights або спеціальні методи oversampling (наприґклад, SMOTE для послідовностей).

Для підвищення розрахункових можливостей додайте раннє зупинення, збереження найкращої моделі та періодичне донавчання на нових даних.

Можливо зробити приклад із розгортанням як REST-сервісу (FastAPI) і прикладом інтеграції до контуру моніторингу.

Додаток 7

Код для Google Colab – одна комірка, яка:

1. ***встановлює залежності;***
2. ***генерує дані, тренує LSTM, зберігає модель і скейлер;***
3. ***підіймає Flask-API (/health, /predict) у фоні;***
4. ***робить тестовий запит до API;***
5. ***(необов'язково) має функцію для відправки результатів у Splunk HEC.***

Просто вставте в одну комірку Colab і запустіть

```
# ==== ЄДИНИЙ ЗАГАЛЬНИЙ КОД ДЛЯ COLAB (LSTM + Flask API) ====
# 0) Залежності (Flask, joblib; TF та sklearn є в Colab)
!pip -q install flask==3.0.0 joblib==1.3.2

# 1) Імпорти та базові налаштування
import os, json, joblib, threading, time, warnings, requests
import numpy as np
from pathlib import Path
from flask import Flask, request, jsonify

import tensorflow as tf
from tensorflow.keras.models import Sequential, load_model
from tensorflow.keras.layers import LSTM, Dense, Dropout

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler

warnings.filterwarnings("ignore")
np.random.seed(42)
tf.random.set_seed(42)

# 2) Шляхи збереження артефактів (можна змінити на Google Drive)
BASE_DIR = Path("/content")                # за потреби змініть на `/content/drive/MyDrive/…`
MODEL_DIR = BASE_DIR / "models"
MODEL_DIR.mkdir(parents=True, exist_ok=True)
MODEL_PATH = MODEL_DIR / "lstm_anomaly_model.h5"
SCALER_PATH = MODEL_DIR / "scaler.pkl"
# 3) Синтетичні дані (приклад). Підставте свої за потреби.
n_samples, n_timesteps, n_features = 4000, 20, 10
X = np.random.rand(n_samples, n_timesteps, n_features).astype("float32")
y = np.random.randint(0, 2, size=n_samples).astype("int32")

# 4) Трен/тест розбиття ПО ІНДЕКСАХ (щоб фітити скейлер ТІЛЬКИ на train)
```

```
idx_train, idx_test =
train_test_split(np.arange(n_samples), test_size=0.2,

random_state=42, stratify=y)
X_train, y_train = X[idx_train], y[idx_train]
X_test,  y_test  = X[idx_test],  y[idx_test]

# 5) Масштабування: fit на TRAIN, transform усюди
scaler = MinMaxScaler()
X_train_2d = X_train.reshape(-1, n_features)
X_test_2d  = X_test.reshape(-1,  n_features)
scaler.fit(X_train_2d)
X_train_scaled =
scaler.transform(X_train_2d).reshape(len(idx_train),
n_timesteps, n_features)
X_test_scaled  = scaler.transform(X_test_2d
).reshape(len(idx_test),  n_timesteps, n_features)

# 6) Модель LSTM
model = Sequential([
LSTM(64, input_shape=(n_timesteps, n_features)),
Dropout(0.25),
Dense(1, activation="sigmoid")
])
model.compile(optimizer="adam",
loss="binary_crossentropy", metrics=["accuracy"])
history = model.fit(X_train_scaled, y_train,
epochs=6, batch_size=64,
validation_split=0.2, verbose=1)

# 7) Збереження артефактів
model.save(MODEL_PATH.as_posix())
joblib.dump({"scaler": scaler, "n_timesteps":
n_timesteps, "n_features": n_features}, SCALER_PATH)

loss, acc = model.evaluate(X_test_scaled, y_test,
verbose=0)
print(f"[INFO] Test accuracy: {acc:.3f}")
print(f"[INFO] Saved MODEL to:  {MODEL_PATH}")
print(f"[INFO] Saved SCALER to: {SCALER_PATH}")

# 8) Flask-API: завантаження з диска, підготовка
послідовностей, передбачення
app = Flask(__name__)
```

```
# Завантаження артефактів (імітуємо реальне прод-
                    оточення)
_loaded_model = load_model(MODEL_PATH.as_posix())
        _meta = joblib.load(SCALER_PATH)
        _loaded_scaler = _meta["scaler"]
       SEQ_LEN = int(_meta["n_timesteps"])
       N_FEAT  = int(_meta["n_features"])

              @app.get("/health")
                 def health():
                  return jsonify({
                    "status": "ok",
           "model_path": MODEL_PATH.as_posix(),
          "scaler_path": SCALER_PATH.as_posix(),
                  "seq_len": SEQ_LEN,
                  "n_features": N_FEAT
                        })

         def _prepare_sequence(events):
    """Подія = {"features": [...]} довжини N_FEAT;
       будуємо послідовність SEQ_LEN."""
      feats = [e["features"] for e in events if
                 "features" in e]
       X = np.asarray(feats, dtype="float32")

       if X.ndim != 2 or X.shape[1] != N_FEAT:
      raise ValueError(f"Очікувано features розміру
       (*, {N_FEAT}), отримано {X.shape}")

        # лише transform тим самим скейлером!
          X = _loaded_scaler.transform(X)

          # паддінг/підрізання до SEQ_LEN
               if len(X) >= SEQ_LEN:
                  X = X[-SEQ_LEN:, :]
                       else:
       pad = np.zeros((SEQ_LEN - len(X), N_FEAT),
                dtype="float32")
                 X = np.vstack([pad, X])
        return X.reshape(1, SEQ_LEN, N_FEAT)

             @app.post("/predict")
                 def predict():
       payload = request.get_json(force=True,
              silent=True) or {}
         events = payload.get("events", [])
```

```
    threshold = float(payload.get("threshold", 0.8))
    if not isinstance(events, list) or len(events) ==
0:
        return jsonify({"error": "Очікується
непорожній список 'events'"}), 400
    try:
        X = _prepare_sequence(events)
        score = float(_loaded_model.predict(X,
verbose=0).ravel()[0])
        is_anomaly = score >= threshold
        return jsonify({"anomaly_score": score,
"is_anomaly": is_anomaly, "threshold": threshold})
    except Exception as e:
        return jsonify({"error": str(e)}), 400

def _run_app():
    # use_reloader=False важливо для середовища
ноутбука
    app.run(host="0.0.0.0", port=5000, debug=False,
use_reloader=False)

# 9) Стартуємо сервер у фоні цієї ж комірки і робимо
тестовий запит
thread = threading.Thread(target=_run_app,
daemon=True)
thread.start()
time.sleep(1.5)  # коротка пауза, щоб сервер піднявся

# Тест /health
try:
    print("[TEST] /health ->",
requests.get("http://127.0.0.1:5000/health",
timeout=5).json())
except Exception as e:
    print("[WARN] Не вдалося звернутися до /health:",
e)
# Тест /predict (штучна послідовність з 20 подій; API
сам піджене до SEQ_LEN)
events = [{"features": [0.1]*N_FEAT} for _ in
range(20)]
try:
    r =
requests.post("http://127.0.0.1:5000/predict",
                  json={"events": events,
"threshold": 0.7}, timeout=10)
```

```
        print("[TEST] /predict ->", r.status_code,
                        r.json())
                except Exception as e:
            print("[WARN] Не вдалося звернутися до
                        /predict:", e)

# 10) (Необов'язково) функція для Splunk HEC —
                залишена для зручності
            def send_to_splunk(result: dict,
                               hec_url: str,
                              hec_token: str,
                          sourcetype: str = "_json",
                           verify_tls: bool = True):
            headers = {"Authorization": f"Splunk
                        {hec_token}"}
         payload = {"event": result, "sourcetype":
                        sourcetype}
      resp = requests.post(hec_url, headers=headers,
                data=json.dumps(payload),
                               verify=verify_tls,
                        timeout=10)
             return resp.status_code, resp.text

          print("\n[READY] Flask API працює на
       http://127.0.0.1:5000  (ендпойнти: /health,
                        /predict)")
```

Як користуватись:

- За потреби збереження на Google Drive змонтуйте диск та змініть BASE_DIR на вашу теку в Drive.
- У продуктивному середовищі використовуйте власні дані замість синтетичних, а також додайте аутентифікацію до API та TLS.

Наукове видання

ТЕРМІНОЛОГІЧНИЙ АНАЛІЗ ТА КЛАСИФІКАЦІЯ ПОНЯТЬ У НЕЙРОМЕРЕЖЕВИХ МЕТОДАХ УПРАВЛІННЯ РИЗИКАМИ КІБЕРФІЗИЧНИХ СИСТЕМ

Монографія

Прокопович-Ткаченко Дмитро Ігорович

Підписано до друку 03.09.2025. Формат 60×84 1/16. Папір офсетний.
Ум. друк. арк. 21,94. Облік.-вид. арк. 19,50. Тираж 300 прим.
Замовлення № 1.
Дніпро: Університет митної справи та фінансів (свідоцтво про видавничу діяльність ДК № 6198 від 24.05.2018 р.).
49000, м. Дніпро, вул. Володимира Вернадського