Міністерство освіти і науки України
Університет митної справи та фінансів

Факультет інноваційних технологій
Кафедра комп'ютерних наук та інженерії програмного забезпечення

Кваліфікаційна робота магістра

на тему: «Інформаційна система багатофакторної автентифікації за допомогою ланцюгів Маркова в умовах ризику та невизначеності»

Виконав: студент групи К23-1М
Спеціальність 122 «Комп'ютерні науки»
Задерій Олександр Дмитрович
(прізвище та ініціали)

Керівник к.т.н., доцент Чупілко Т.А.
(науковий ступінь, вчене звання, прізвище та ініціали)

Рецензент Університет митної справи та фінансів
(місто роботи)
доцент кафедри кібербезпеки та інформаційних технологій
(посада)
к.т.н. Савченко Ю.В
(науковий ступінь, вчене звання, прізвище та ініціали)

Дніпро – 2025

АНОТАЦІЯ

Задерій О.Д. Інформаційна система багатофакторної автентифікації за допомогою ланцюгів Маркова в умовах ризику та невизначеності.

Кваліфікаційна робота на здобуття освітнього ступеня магістр за спеціальністю 122 «Комп’ютерні наук». – Університет митної справи та фінансів, Дніпро, 2024.

Об'єктом дослідження є оптимізація процесу багатофакторної автентифікації за допомогою ланцюгів Маркова в умовах ризику та невизначеності.

Предметом дослідження є процеси багатофакторної автентифікації, оптимізовані з використанням марковських моделей для прогнозування ефективності в умовах змінних загроз.

Метою роботи є розробка інформаційної системи для оптимізації багатофакторної автентифікації в умовах ризику та невизначеності за допомогою ланцюгів Маркова.

Дана робота присвячена вивченню методів багатофакторної автентифікації, що враховують ризики та невизначеності шляхом використання ланцюгів Маркова для прогнозування ймовірності успішного проходження кожного етапу автентифікації. Основна увага в роботі зосереджена на формалізації ймовірностей успішного виконання кожного з етапів і прогнозуванні можливих ризиків та загроз на основі марковських моделей.

Особливістю даної розробки є оптимізація процесу багатофакторної автентифікації, що дозволяє підвищити рівень безпеки інформаційних систем, враховуючи змінні умови та потенційні загрози, зберігаючи при цьому зручність користувачів. Такі алгоритми дозволяють створювати адаптивні системи автентифікації, здатні ефективно працювати в умовах невизначеності та мінімізувати ризики.

Ключові слова: ланцюги Маркова, багатофакторна автентифікація, ризик, невизначеність, прогнозування, моделювання автентифікації.

ABSTRACT

Zaderii O.D. Information system of multifactor authentication using Markov chains under conditions of risk and uncertainty.

Qualification work for the degree of Master of Science in specialty 122 “Computer Science.” - University of Customs and Finance, Dnipro, 2024.

The object of research is to optimize the process of multifactor authentication using Markov chains under conditions of risk and uncertainty.

The subject of the study is multifactor authentication processes optimized using Markov models to predict performance under variable threats.

The aim of the study is to develop an information system for optimizing multifactor authentication under risk and uncertainty using Markov chains.

This work is devoted to the study of multifactor authentication methods that take into account risks and uncertainties by using Markov chains to predict the probability of successful completion of each stage of authentication. The main focus of the work is on formalizing the probability of successful completion of each stage (e.g., password entry, confirmation via SMS/email) and predicting possible risks and threats based on Markov models.

The peculiarity of this development is the optimization of the multi-factor authentication process, which allows to increase the security level of information systems, taking into account changing conditions and potential threats, while maintaining user convenience. Such algorithms make it possible to create adaptive authentication systems that can work effectively under conditions of uncertainty and minimize risks.

Keywords: Markov chains, multifactor authentication, risk, uncertainty, forecasting, authentication modeling.

ЗМІСТ

СЛОВНИК ТЕРМІНІВ

1. Brute force — метод атаки, при якому перевіряються всі можливі варіанти для знаходження правильного рішення.
2. HTTPS — протокол захищеного зв’язку (Hypertext Transfer Protocol Secure).
3. MFA — багатофакторна автентифікація (Multi-Factor Authentication).
4. MitM — атака типу "людина посередині" (Man-in-the-Middle).
5. OTP — одноразовий пароль (One-Time Password).
6. SQL — мова структурованих запитів (Structured Query Language).
7. TLS — транспортний рівень захисту (Transport Layer Security).

ВСТУП

Адаптивна багатофакторна автентифікація передбачає створення методів автентифікації, які змінюються та адаптуються в залежності від умов ризику та невизначеності, взаємодії з користувачем та потенційних загроз. Центральну роль у такому підході відіграють передові алгоритми та моделі, зокрема ланцюги Маркова, які дозволяють прогнозувати ймовірність успішного проходження кожного етапу автентифікації, враховуючи ризики та невизначеність. Ці моделі дають змогу забезпечити персоналізований та надійний досвід для користувачів, адаптуючи систему до змінних умов.

Актуальність теми розробки алгоритму для оптимізації багатофакторної автентифікації в умовах ризику та невизначеності полягає в зростаючій потребі у забезпеченні високого рівня безпеки в інформаційних системах при збереженні зручності для користувачів. Враховуючи постійно змінювані умови та загрози, здатність ефективно аналізувати й адаптувати процеси автентифікації є ключовим фактором успіху для розробників безпечних систем. Це дозволяє не тільки підвищити рівень захисту, але й зменшити можливі ризики, які можуть виникнути в процесі взаємодії з користувачем.

Застосування ланцюгів Маркова для моделювання процесу автентифікації дає змогу створювати адаптивні системи, які здатні передбачати поведінку користувача та оптимізувати процес автентифікації залежно від різних загроз та сценаріїв. Такі системи можуть забезпечити високу ефективність в умовах невизначеності, мінімізуючи ризики та покращуючи досвід користувачів.

Новизна дослідження полягає в використанні ланцюгів Маркова для моделювання та оптимізації багатофакторної автентифікації в умовах змінних загроз і невизначеності. Це дозволяє підвищити надійність та ефективність автентифікаційних систем без необхідності застосування складних і дорогих технологій штучного інтелекту, що є особливо важливим для інформаційних систем з обмеженими ресурсами.

Метою дослідження є розробка інформаційної системи для оптимізації

багатофакторної автентифікації в умовах ризику та невизначеності за допомогою ланцюгів Маркова.

Для досягнення поставленої мети в кваліфікаційній роботі ставились та вирішувались наступні завдання дослідження:

1. Провести аналіз існуючих методів багатофакторної автентифікації та їх адаптації до умов ризику.
2. Розробити модель багатофакторної автентифікації на основі ланцюгів Маркова для прогнозування ймовірностей успішного проходження етапів.
3. Реалізувати та протестувати алгоритм для оцінки ефективності багатофакторної автентифікації в різних сценаріях.
4. Розробити програмне забезпечення для автоматизованої перевірки ефективності системи.

Методи дослідження у роботі будуть використані методи моделювання та прогнозування на основі ланцюгів Маркова, а також розробка програмного забезпечення на мові PHP.

Об'єкт дослідження: Процес багатофакторної автентифікації в умовах змінних загроз, що впливають на ефективність системи.

Предмет дослідження: Методи дослідження включають моделювання та прогнозування на основі ланцюгів Маркова, а також розробку програмного забезпечення на мові PHP для реалізації алгоритму багатофакторної автентифікації. Ланцюги Маркова є потужним інструментом для моделювання та прогнозування процесів, що дозволяє точно оцінити ймовірність успіху на кожному етапі ідентифікації користувача та передбачити потенційні ризики на наступних етапах.

Практичне значення отриманих результатів: Розробка адаптивної системи для багатофакторної автентифікації, яка дозволяє знижувати ризики та підвищувати безпеку інформаційних систем. Використання ланцюгів Маркова для прогнозування ефективності автентифікації дозволяє створити гнучкі і надійні системи, здатні адаптуватися до змінних умов і мінімізувати потенційні

загрози.

Структура роботи:

Розділ 1 Огляд існуючих методів багатофакторної автентифікації та ланцюгів Маркова.

Розділ 2 Розробка моделі багатофакторної автентифікації з використанням ланцюгів Маркова.

Розділ 3 Реалізація та тестування алгоритму для оптимізації автентифікаційних процесів.

Розділ 4 Оцінка ефективності системи на основі моделювання в різних сценаріях загроз.

Робота складається зі вступу, 4-х розділів, висновків, списку використаної літератури з 29 джерел, 1 додатку. Обсяг роботи 74 сторінки, 9 таблиць, 9 рисунків та 18 формул.

РОЗДІЛ 1.
ОГЛЯД ІСНУЮЧИХ МЕТОДІВ БАГАТОФАКТОРНОЇ АВТЕНТИФІКАЦІЇ ТА ЛАНЦЮГІВ МАРКОВА

1.1 Поняття багатофакторної автентифікації.

Багатофакторна автентифікація (МФА) — це метод безпеки, який вимагає від користувача надати два або більше факторів для перевірки його ідентичності. Ці фактори належать до трьох основних категорій: щось, що знає користувач, щось, що має користувач, і щось, чим є користувач. Основною метою МФА є підвищення рівня безпеки шляхом комбінування різних видів підтвердження особистості, що значно знижує ймовірність несанкціонованого доступу навіть у разі компрометації одного з факторів. Існують різні форми МФА, зокрема:

1) Пароль або PIN-код. Це найпоширенішим і найпростішим варіантом першого фактору автентифікації. Цей фактор вимагає від користувача знати певну інформацію, яку не повинно бути легко здогадатися або знайти.

Незважаючи на свою популярність, паролі та PIN-коди мають суттєві недоліки: їх можна забути, зловити за допомогою шкідливого програмного забезпечення або вгадати через брутфорс-атаки.

2) Біометрія. Біометрія включає в себе використання фізичних характеристик людини для підтвердження її особи. Цей фактор є надійним, оскільки біометричні дані унікальні для кожної людини та не можуть бути легко підроблені або здогадані.

Хоча біометрія є високоточним методом автентифікації, вона має свої обмеження, такі як високі вимоги до технологій сканування і можливість незначних помилок через фізичні зміни (наприклад, травми або зміни на обличчі).

3) Смарт-карти та токени. Вони є фізичними носіями, які

використовуються як фактори автентифікації. Ці пристрої можуть генерувати одноразові паролі (OTP), що використовуються для підтвердження особи користувача.

Ці пристрої часто використовуються в корпоративному середовищі, де забезпечення фізичної безпеки є важливим. Однак їх вартість може бути вищою, і вони потребують регулярного оновлення або заміни.

4) SMS або Email-коди. Цей фактор передбачає відправку тимчасових кодів на мобільний телефон або електронну пошту користувача, що дозволяє підтвердити його особу.

Хоча цей метод є простим і доступним, він має свої вразливості, такі як можливість перехоплення SMS-повідомлень або доступу до електронної пошти. Однак у поєднанні з іншими факторами МФА він забезпечує додатковий рівень безпеки.

Існують різні методи, що реалізують багатофакторну автентифікацію. Найбільш поширені з них включають:

- Статичні методи МФА – ці методи передбачають поєднання статичних елементів, таких як паролі, із фізичними пристроями, наприклад, смарт-картами або токенами.
- Динамічні методи МФА – основною особливістю цих методів є змінність другого фактора після кожної автентифікації. Наприклад, одноразовий пароль (OTP), згенерований програмою або апаратним пристроєм, діє лише протягом обмеженого часу.
- Біометричні методи – ці методи базуються на унікальних фізичних характеристиках людини, таких як відбитки пальців, сканування обличчя або райдужної оболонки ока. Біометрія є одним із найбільш надійних способів автентифікації, оскільки ці дані важко підробити або вкрасти.

Усі ці методи мають різні рівні безпеки та застосовуються в залежності від рівня вимог до захисту інформації.

Мультифакторна автентифікація є невід'ємною складовою сучасних систем безпеки, що дозволяє значно підвищити рівень захисту від несанкціонованого доступу до критичних даних та ресурсів. Враховуючи швидкий розвиток загроз, впровадження надійних методів автентифікації стає важливим елементом забезпечення інформаційної безпеки. На ринку існує безліч рішень, які пропонують різноманітні способи реалізації MFA, від традиційних паролів до використання біометричних даних, смарт-карток і токенів[2].

Компанії, що надають послуги MFA, зазвичай пропонують інтерфейси та інструменти, які дозволяють інтегрувати їхні рішення з існуючими корпоративними системами, включаючи сервери, застосунки та платформи для управління ідентичністю. Такі платформи, як ADSelfService Plus від ManageEngine, Ping Identity, Duo Security, Okta, та RSA SecurID, мають у своєму арсеналі різноманітні методи аутентифікації, включаючи мобільні додатки, апаратні токени, смарт-карти та інші [13].

ADSelfService Plus забезпечує можливість реалізації багатофакторної автентифікації в середовищах, що використовують Active Directory, що робить його зручним рішенням для корпоративних інфраструктур. Ping Identity пропонує платформу для гнучкого управління доступом та ідентичністю, підтримуючи різноманітні методи MFA. Duo Security, придбана Cisco, спеціалізується на простоті інтеграції та підтримує широкий спектр автентифікаційних методів. Okta, як лідер у сфері управління доступом, дозволяє організаціям реалізувати багатофакторну автентифікацію через єдину платформу для управління ідентичністю. RSA SecurID має багатий досвід у реалізації високоякісних рішень для MFA, пропонуючи як апаратні, так і програмні методи автентифікації.

Для кращого розуміння різниць між цими рішеннями, пропонується порівняльна таблиця, яка надає більш детальний аналіз кожного з продуктів:

Таблиця 1.1

Порівняння компаній, що надають рішення для багатофакторної автентифікації

Назва	Інтеграції	Переваги
ADSelfService	Cisco, Fortinet, G Suite, Office 365 і Salesforce	Легка навігація для кінцевих користувачів. Зручний інтерфейс користувача. Надає багатофакторну автентифікацію (MFA) і синхронізатор паролів у реальному часі.
PingIdentity	AWS, Azure, Google Cloud, Salesforce і Office 365.	У мене є можливість адаптувати політику відповідно до того, що мені потрібно. Push-повідомлення ефективні. Підтримує багато стандартів відповідності.
Duo Security	Microsoft 365, Google Workspace і Salesforce.	Імпортування користувачів з Active Directory забезпечує швидку адаптацію. Легке розгортання 2fa by Cisco безпечний доступ. Задокументований посібник, який допоможе вам вибрати з різних постачальників MFA.
Okta	Salesforce, AWS, GitHub, Slack, SAP, Oraclei PeopleSoft.	Пропонує самообслуговування для скидання та резервного копіювання Є можливість плавної інтеграцію з широким спектром програм.
RSA SecurID	Microsoft 365, Salesforce, Google Workspace та Cisco	Термін дії повідомлень автоматичної генерації токенів. Забезпечує чудову безпеку ідентифікації за допомогою SSO

Традиційні методи автентифікації, хоча й забезпечують базову безпеку, мають низку суттєвих недоліків, які можуть ставити під загрозу цілісність інформаційних систем. З розвитком технологій і постійним вдосконаленням методів злому стає очевидним, що ці методи не здатні повною мірою захистити дані та системи від все більш складних і витончених атак [3].

Паролі та PIN-коди залишаються найбільш популярними методами автентифікації, однак вони мають кілька серйозних недоліків, що знижують їхню ефективність у забезпеченні безпеки. Першим з таких недоліків є слабкість паролів.

Багато користувачів вибирають паролі, які легко вгадати або знайти за допомогою соціальної інженерії, наприклад, використовуючи прості послідовності чисел «123456», загальні слова як «password» або навіть імена домашніх тварин. Хоча стандартні рекомендації з безпеки передбачають використання складних паролів, що поєднують великі та малі літери, цифри і спеціальні символи, на практиці багато користувачів ігнорують ці вимоги, що робить їхні паролі вразливими до атак.

Іншим серйозним недоліком є можливість brute force атак. Зловмисники можуть використовувати автоматизовані методи для підбору паролів, випробовуючи всі можливі варіанти. Якщо пароль слабкий або використовується на кількох сайтах одночасно, ймовірність успіху такого підбору значно збільшується. Сучасні системи можуть затримувати спроби входу або застосовувати капчу для захисту, але навіть ці методи не завжди є надійними, оскільки брутфорс атаки можуть бути дуже швидкими і ефективними. Крім того, паролі можуть бути викрадені за допомогою фішингових атак або шкідливих програм, які здатні зібрати облікові дані користувача без його відома.

Недоліки у зберіганні паролів також можуть призвести до їхнього викрадення, якщо дані зберігаються в ненадійних базах даних без належного шифрування. Також важливою проблемою є забування паролів. Користувачі часто забувають свої паролі, що змушує їх регулярно скидувати паролі через електронну пошту або через служби підтримки, що не лише витрачає час, але й створює додаткові ризики для безпеки, оскільки процес скидання пароля часто є точкою вразливості для зловмисників.

Біометрія є однією з найбільш перспективних форм автентифікації, оскільки вона базується на фізичних характеристиках людини, які унікальні та важко підробити. Хоча біометрія є перспективним напрямком завдяки своїй складності для підробки, вона має кілька суттєвих обмежень. Технології біометрії, хоча і розвиваються, є досить дорогими для масового впровадження, і для їх коректної роботи часто потрібно спеціалізоване обладнання, що значно збільшує витрати на їхнє використання.

Крім того, технічні обмеження можуть впливати на точність розпізнавання: наприклад, сканери відбитків пальців можуть не працювати через забруднення або пошкодження пальця, а розпізнавання обличчя може бути неефективним за поганого освітлення або зміни зовнішнього вигляду користувача, наприклад, при носінні окулярів або масок. Окрім того, біометричні дані є особливо чутливими, і в разі їх витоку чи компрометації можуть виникнути серйозні наслідки, оскільки ці дані неможливо змінити, як паролі, що робить їх незахищеними в разі зламу [4].

Смарт-карти та токени, які генерують одноразові паролі, є іншими популярними методами автентифікації, проте вони також мають кілька недоліків. Перш за все, для їхнього використання користувач має мати фізичний носій, що може бути втраченим або вкраденим, що ставить під загрозу доступ до системи. Крім того, розподіл і управління великою кількістю таких носіїв для організацій може бути дорогим і складним процесом, особливо для компаній з численними віддаленими співробітниками. Також не всі пристрої підтримують смарт-карти або токени, що може створювати проблеми для користувачів, які намагаються здійснити автентифікацію з мобільних пристроїв або інших не сумісних пристроїв [12].

Використання одноразових кодів, що надсилаються через SMS або електронну пошту, є популярним способом додаткової автентифікації, але й він має ряд проблем. Така система може бути вразливою до перехоплення повідомлень, особливо якщо зловмисники отримають доступ до мобільного телефону або електронної пошти користувача.

Крім того, фішингові атаки можуть бути використані для отримання одноразових кодів, коли зловмисники під виглядом офіційних повідомлень змушують користувача передати ці коди. Також важливою проблемою є залежність від зовнішніх сервісів для доставки кодів, що може призвести до затримок або повної неможливості отримати код, якщо виникають збої в мобільних мережах або поштових сервісах.

1.2 Застосування ланцюгів Маркова в багатофакторній автентифікації

Ланцюг Маркова — це математична модель, що описує систему з кінцевим або рахунковим числом можливих станів. Перехід між станами описується ймовірностями, які називаються матрицею перехідних ймовірностей. Одна з основних характеристик ланцюгів Маркова полягає в тому, що ймовірність переходу в наступний стан залежить лише від поточного стану системи, а не від попередніх етапів. Ця властивість називається "марківською властивістю" або "безмеморіальність".

У випадку багатофакторної автентифікації ланцюги Маркова дозволяють моделювати різні етапи аутентифікації як окремі стани системи. Наприклад, кожен етап — це певний крок перевірки (введення пароля, підтвердження через біометрію, одноразовий код через SMS тощо). Для кожного етапу можна визначити ймовірність переходу в наступний етап в залежності від правильності або неправильності введених даних або виконаних дій.

Застосування ланцюгів Маркова (Форм. 1.1) для моделювання етапів багатофакторної автентифікації дозволяє створити математичну модель, яка враховує ймовірності переходів між різними станами автентифікації. Кожен етап, як-от введення пароля, біометрична перевірка або одноразовий код, може бути представленим як окремий стан у ланцюзі Маркова. Ймовірності переходів між станами залежать від правильності введених даних або результату перевірки, що дає змогу оцінити загальну ймовірність успішного завершення автентифікації. Така модель допомагає прогнозувати ризики та оптимізувати систему для підвищення безпеки, зменшуючи ймовірність несанкціонованого доступу навіть у випадку зміни загроз або збоїв на певних етапах процесу. Основна формула для ланцюгів Маркова записується так:

$$P(X_{n+1} = x | X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) = P(X_{n+1} = x | X_n = x_n) \qquad (1.1)$$

де:

- P — ймовірність переходу між станами;
- X_n — стан системи автентифікації на кроці nnn;
- X_{n+1} — стан системи на наступному кроці n+1n+1n+1;
- x — конкретний стан, у який система може перейти;
- $X_1, X_2, \ldots, X_n$ — послідовність попередніх станів.

Однією з основних переваг використання ланцюгів Маркова в багатофакторній автентифікації є здатність моделювати складні процеси з урахуванням випадкових і непередбачуваних факторів. Ланцюги Маркова дозволяють точно прогнозувати ймовірності успішного завершення кожного етапу автентифікації, що дозволяє зменшити ризики несанкціонованого доступу [11].

Гнучкість**.** Ланцюги Маркова забезпечують високу гнучкість у розробці та адаптації стратегії автентифікації, оскільки вони дозволяють змінювати правила залежно від змінних умов або ризиків. Така можливість важлива для постійного вдосконалення системи, наприклад, шляхом інтеграції нових методів автентифікації, таких як біометрія чи додаткові фактори. В умовах зміни поведінки користувачів система може змінювати стратегію автентифікації, реагуючи на ці зміни, що дозволяє підтримувати баланс між зручністю користувачів та високим рівнем безпеки. Це особливо корисно у разі появи нових загроз, коли необхідно оперативно адаптувати систему до нових реалій без значних змін у базовій архітектурі.

Можливість прогнозування. Ланцюги Маркова дають змогу не лише оцінювати поточний стан системи, а й прогнозувати ймовірність успішного чи неуспішного проходження кожного етапу автентифікації. Це дозволяє передбачити, скільки часу займе процес верифікації, а також імовірність затримок чи відмов на певних етапах, таких як введення пароля або OTP-коду. Завдяки цьому можна оптимізувати час, необхідний для проходження автентифікації, і вчасно реагувати на можливі проблеми. Прогнозування допомагає забезпечити безперервність і стабільність

процесу авторизації, що критично важливо для забезпечення зручності користувачів та ефективної роботи системи.

Зниження ризиків. Однією з основних переваг ланцюгів Маркова є можливість більш ефективного управління ризиками на кожному етапі автентифікації. Система може визначати ймовірність успіху або невдачі на кожному етапі, що дозволяє оцінювати загальний рівень безпеки в конкретний момент часу. Якщо ймовірність зловмисного вторгнення чи атаки підвищена, система автоматично адаптує механізми автентифікації, наприклад, ініціюючи додаткову перевірку користувача. Це дозволяє мінімізувати ризики несанкціонованого доступу та знижує ймовірність успіху атак, таких як brute force або соціальна інженерія, забезпечуючи таким чином більш високий рівень безпеки для користувачів.

Оптимізація користувацького досвіду. Багатофакторна автентифікація може здатися складною або неприємною для користувачів через велику кількість етапів перевірки. Ланцюги Маркова дозволяють оптимізувати цей процес, забезпечуючи більш ефективну та зручну автентифікацію без шкоди для безпеки. Завдяки адаптивності системи, кількість етапів перевірки може бути знижена для користувачів, які постійно демонструють безпечну поведінку, зберігаючи при цьому високий рівень захисту. Таким чином, користувачі мають можливість пройти автентифікацію швидше та зручніше, що позитивно впливає на їхній досвід, а система при цьому залишається стійкою до загроз.

Застосування ланцюгів Маркова в багатофакторній автентифікації має значний потенціал для поліпшення ефективності та безпеки сучасних інформаційних систем. Вони дозволяють створювати адаптивні моделі автентифікації, що реагують на змінні загрози та забезпечують баланс між безпекою і зручністю для користувача. Завдяки використанню марковських моделей можливо прогнозувати ймовірності успіху на кожному етапі автентифікації, що дозволяє своєчасно реагувати на потенційні ризики та підвищувати загальний рівень безпеки системи.

Продуктивність марковських моделей є важливим фактором, що визначає їхню придатність для систем багатофакторної автентифікації. Висока швидкість

обробки запитів і точність виявлення загроз забезпечують ефективність системи в режимі реального часу. Марковські моделі відзначаються тим, що їхній алгоритмічний підхід дозволяє швидко обчислювати ймовірності переходів між станами без необхідності зберігання великої кількості історичних даних. Це робить їх оптимальними для використання в ситуаціях, де важливо забезпечити негайний відгук системи на дії користувачів [18].

Однією з переваг марковських моделей є здатність до обробки великих обсягів даних у стислі терміни. В умовах високого навантаження, наприклад, коли в систему одночасно входять сотні користувачів, моделі зберігають стабільність роботи. Вони ефективно аналізують поведінку кожного користувача окремо, забезпечуючи низький рівень затримок при прийнятті рішень. Завдяки такій масштабованості, системи, побудовані на основі марковських моделей, можуть використовуватися в організаціях із великою кількістю користувачів, наприклад, у фінансових установах, урядових платформах або системах електронної комерції.

Важливим аспектом продуктивності є точність виявлення загроз. Марковські моделі демонструють високий рівень точності у виявленні аномальних патернів поведінки, які можуть свідчити про потенційні загрози. Вони здатні ідентифікувати навіть незначні відхилення у діях користувача, наприклад, зміну часу входу, IP-адреси або послідовності автентифікаційних кроків. Це дозволяє системі автоматично застосовувати додаткові перевірки або блокувати підозрілу активність без потреби втручання адміністратора.

Ще одним важливим показником є кількість хибнопозитивних і хибнонегативних спрацьовувань. Марковські моделі мінімізують ці показники завдяки використанню ймовірнісного підходу до аналізу. Наприклад, система може розпізнати легітимного користувача, навіть якщо той випадково ввів пароль з другої спроби, і водночас заблокувати підозрілі спроби входу з незвичного пристрою чи місця. Це сприяє покращенню досвіду користувача, знижуючи кількість помилкових відмов у доступі [24].

Додатковою перевагою марковських моделей є їхня оптимізація для роботи в умовах обмежених обчислювальних ресурсів. На відміну від складних нейронних мереж або методів машинного навчання, марковські моделі мають менші вимоги до пам'яті та процесорної потужності. Це робить їх придатними для використання в системах, що працюють на вбудованих пристроях або в середовищах із низькою пропускною здатністю мережі.

Таким чином, аналіз продуктивності марковських моделей підтверджує їхню ефективність у системах багатофакторної автентифікації. Вони поєднують високу швидкість обробки запитів, точність виявлення аномалій і економічність використання ресурсів. Це дозволяє застосовувати їх як у масштабних корпоративних середовищах, так і в менш ресурсомістких системах, гарантуючи баланс між безпекою та продуктивністю [10].

1.3Байєсівські моделі

Байєсівські моделі ґрунтуються на теоремі Байєса, яка дозволяє оновлювати ймовірності подій з урахуванням нової інформації. Це може бути корисно для прогнозування ймовірності успішного проходження етапів багатофакторної автентифікації, коли система отримує нові дані про користувача, що допомагають коригувати ймовірності.

Байєсівські моделі в контексті автентифікації користувачів представляють потужний інструмент для створення адаптивних систем безпеки. В основі цього підходу лежить теорема Байєса:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \quad (1.2)$$

яка дозволяє системі постійно оновлювати свої оцінки на основі нових даних.

На початковому етапі система оперує базовою ймовірністю P(A), яка відображає шанси того, що запит на автентифікацію надходить від легітимного користувача. Ця оцінка зазвичай формується на основі історичних даних про співвідношення успішних входів до спроб зламу в системі. Коли користувач надає свої облікові дані, система оцінює P(B|A) - ймовірність того, що легітимний користувач надасть саме такі дані, а також враховує P(B) - загальну ймовірність появи таких даних, включаючи можливі спроби зламу. На основі цих даних формується фінальна оцінка P(A|B), яка показує оновлену ймовірність того, що користувач є легітимним.

Важливою особливістю Байєсівських моделей є їхня здатність до адаптивного навчання. Система може враховувати широкий спектр факторів: час доступу, геолокацію, патерни введення паролю та інші поведінкові характеристики. З кожною успішною автентифікацією модель уточнює свої оцінки, а при виявленні аномальної поведінки може автоматично підвищувати рівень перевірки.

У практичному застосуванні це дозволяє динамічно налаштовувати пороги безпеки та обирати додаткові фактори автентифікації на основі поточної оцінки ризиків. Система може ефективно виявляти потенційні атаки, аналізуючи відхилення від звичних патернів поведінки користувача. Такий підхід забезпечує значну гнучкість, оскільки система здатна адаптуватися до індивідуальних особливостей кожного користувача.

Вибір між байєсівськими та марковськими моделями залежить від специфіки задачі. В контексті багатофакторної автентифікації марковські моделі є більш доречними, оскільки вони дозволяють ефективно прогнозувати ймовірності на кожному етапі автентифікації на основі поточного стану, що є важливим для оцінки ризиків та оптимізації процесу автентифікації.

- Моделі пам'яті:

Марковські моделі працюють на принципі "без пам'яті", де ймовірність переходу в наступний стан залежить лише від поточного стану, а не від попередніх

подій. Це дозволяє моделювати процеси автентифікації, де кожен етап оцінюється незалежно від минулого, що робить модель простішою та ефективнішою для реального застосування. Такий підхід є особливо корисним для багатофакторної автентифікації, де важливо швидко приймати рішення на кожному етапі, не обтяжуючи систему складними обчисленнями.

Байєсівські моделі, навпаки, враховують всю попередню інформацію, що може бути корисно в ситуаціях, коли потрібно врахувати більше даних або виявити закономірності в довгостроковій історії користувача. Цей підхід дозволяє враховувати динаміку змін у поведінці користувача, а також реагувати на різні фактори, які можуть впливати на безпеку системи, зокрема, на основі даних про попередні аутентифікаційні спроби. Проте, цей підхід додає складність у реалізацію та може бути менш ефективним у режимі реального часу, де швидкість прийняття рішень є важливим фактором.

- Оновлення ймовірностей:

У *марковських моделях* ймовірності переходу між станами є фіксованими, що означає, що вони не потребують постійного оновлення на основі нових даних. Кожен етап автентифікації в таких моделях є незалежним від попередніх, що значно спрощує реалізацію та обчислення. Це робить марковські моделі особливо ефективними для систем багатофакторної автентифікації, де важливо швидко оцінювати ризики та надавати рішення в реальному часі, без необхідності обробляти великий обсяг даних або вносити постійні корективи.

У *байєсівських моделях*, навпаки, ймовірності змінюються з кожною новою подією, що дозволяє більш точно передбачати ймовірність успіху чи невдачі на кожному етапі. Однак цей підхід додає складності, оскільки потребує постійного оновлення ймовірностей на основі нової інформації. Це може бути менш ефективним у реальному часі, особливо в системах, де важливо забезпечити швидку реакцію без затримок на обчислення та оновлення даних. Таким чином, хоча байєсівські моделі забезпечують більшу точність, вони можуть бути

складнішими в імплементації та менш підходящими для систем, що потребують оперативної обробки даних.

- Складність моделювання:

Марковські моделі є простішими у реалізації порівняно з іншими статистичними підходами, зокрема байєсівськими. Це зумовлено їх зосередженістю на чітких переходах між станами, що робить їх ідеальними для багатофакторної автентифікації. У випадку з багатофакторною автентифікацією можна легко моделювати ймовірності успіху або невдачі на кожному окремому етапі. Наприклад, для введення пароля або підтвердження через SMS система оцінює ймовірність того, що користувач успішно виконає кожен з етапів, враховуючи тільки поточний стан і ймовірність переходу до наступного.

З іншого боку, *байєсівські моделі* є більш складними у застосуванні. Вони враховують більшу кількість змінних, що вимагає постійного оновлення ймовірностей на основі додаткової інформації. Байєсівський підхід забезпечує можливість включення умовних ймовірностей, що дозволяє враховувати різні фактори (наприклад, наявність попередніх спроб входу, географічне розташування користувача та інші змінні), але це збільшує складність моделювання та обчислень. Крім того, байєсівські моделі потребують більше часу на навчання та постійну корекцію ймовірностей на основі нових даних, що може бути менш ефективним у реальних умовах з високим рівнем змінності [17].

1.4Висновки до першого розділу

Розглянуті підходи до моделювання процесів автентифікації показали, що використання марковських моделей є одним із найефективніших і найбільш придатних методів для побудови систем багатофакторної автентифікації. Це обумовлено їхньою здатністю аналізувати ймовірності переходів між станами на кожному етапі автентифікації без необхідності збереження історичних даних про

взаємодію користувача з системою. Такий підхід дозволяє значно спростити реалізацію алгоритмів, забезпечуючи високу швидкість обробки запитів, що критично важливо для систем, які функціонують у режимі реального часу.

Марковські моделі проявляють особливу ефективність у ситуаціях, коли потрібно швидко оцінити ризики під час виконання окремих дій користувача, таких як введення пароля, підтвердження OTP-кодом або використання біометричних даних. Їхня здатність автоматично адаптувати рівень додаткових перевірок залежно від поведінки користувача дозволяє забезпечити високу гнучкість системи. Це особливо важливо для протидії загрозам, які виникають у непередбачуваних сценаріях, наприклад, при спробах зловмисників використовувати нестандартні IP-адреси або здійснювати багаторазові спроби входу.

Порівняння з іншими підходами, такими як байєсівські моделі, виявило, що марковські моделі забезпечують вищу продуктивність у завданнях, де ключовими є швидкість і простота обчислень. Хоча байєсівські моделі краще підходять для обробки контекстуально залежних завдань і аналізу складних залежностей, марковські моделі мають перевагу у використанні поточного стану для прогнозування ймовірностей, що робить їх ідеальними для багатофакторних систем автентифікації, орієнтованих на швидку реакцію на загрози.

Додатковою перевагою марковських моделей є їхня адаптивність, яка дозволяє системі реагувати на зміни в поведінці користувачів та потенційні загрози. Це забезпечує високу надійність, зменшуючи ймовірність проникнення зловмисників, і водночас зберігає зручність для легітимних користувачів. Завдяки своїй структурі моделі дозволяють легко масштабувати рішення для роботи з великими потоками запитів, що є важливим для сучасних інформаційних систем.

Таким чином, марковські моделі не тільки спрощують розробку алгоритмів автентифікації, але й забезпечують ефективність та надійність їхньої роботи. Вони дозволяють створювати масштабовані, швидкі й безпечні системи багатофакторної автентифікації, які відповідають сучасним вимогам до продуктивності, простоти

використання та рівня захисту. Це робить їх перспективним напрямом для подальшого розвитку інформаційної безпеки в умовах зростання обсягу кіберзагроз.

Крім того, марковські моделі дозволяють моделювати поведінку користувачів у динамічних умовах, враховуючи різноманітні фактори ризику, такі як геолокація, час доби, частота невдалих спроб входу та інші поведінкові особливості. Це забезпечує не тільки високий рівень адаптивності, але й можливість прогнозування потенційних загроз на основі аналізу поточного стану. Такий підхід дозволяє не лише виявляти ризики на ранніх етапах, але й динамічно налаштовувати алгоритм автентифікації, мінімізуючи втручання в процес для легітимних користувачів. У результаті система стає більш стійкою до нових видів атак, забезпечуючи комплексний підхід до захисту облікових записів і даних користувачів. Це підкреслює важливість інтеграції марковських моделей у сучасні системи інформаційної безпеки.

РОЗДІЛ 2.
РОЗРОБКА МОДЕЛІ БАГАТОФАКТОРНОЇ АВТЕНТИФІКАЦІЇ З ВИКОРИСТАННЯМ ЛАНЦЮГІВ МАРКОВА

2.1 Загальна концепція моделі багатофакторної автентифікації

Розробка моделі багатофакторної автентифікації потребує врахування багатьох факторів, таких як рівень безпеки, зручність використання, ефективність та здатність протидіяти загрозам. МФА забезпечує додаткові рівні захисту, комбінуючи декілька незалежних факторів для перевірки особи користувача. Основна мета такої системи полягає у зниженні ризику компрометації навіть за умови компрометації одного з рівнів.

Модель повинна відповідати сучасним вимогам безпеки та бути стійкою до поширених атак, таких як "людина посередині" (MITM), фішинг, атаки підбору паролів або компрометація одноразових кодів. При цьому важливо визначити типи факторів автентифікації, які будуть використовуватися. До них відносяться те, що знає користувач (паролі, PIN-коди), те, що має користувач (смарт-карти, токени, мобільні пристрої), і те, що є користувачем (біометричні параметри, як-от відбитки пальців або розпізнавання обличчя) [9].

Ланцюги Маркова пропонують ефективний спосіб моделювання ймовірностей успішного проходження етапів автентифікації. Наприклад, можна прорахувати ймовірність успішного введення пароля та підтвердження через SMS, враховуючи марковські властивості системи. Це дає змогу оцінювати змінні фактори ризику та ефективності на кожному етапі процесу [16].

Сценарій автентифікації в моделі передбачає чітко визначені етапи. Спочатку користувач подає запит на доступ, після чого проходить перший рівень перевірки, наприклад введення пароля. Далі відбувається підтвердження через другий фактор, наприклад одноразовий код або біометричні дані. На основі цього система оцінює

ризики та приймає рішення щодо успішності автентифікації.

Важливою частиною моделі є її адаптивність. Якщо під час перевірки виявляється незвична активність, наприклад спроба входу з нового місцезнаходження чи пристрою, система може додати додатковий рівень перевірки. Це мінімізує ймовірність несанкціонованого доступу та підвищує безпеку. Ця концепція дозволяє інтегрувати ланцюги Маркова в процес автентифікації, забезпечуючи врахування змінних умов і високий рівень захисту.

Марковські моделі є одним із найефективніших інструментів для побудови систем аналізу даних та прогнозування поведінки користувачів завдяки їхній здатності до адаптації. Вони базуються на математичній теорії, яка дозволяє описувати ймовірності переходів між різними станами системи. Це особливо важливо для багатофакторної автентифікації, де кожен етап процесу може розглядатися як окремий стан, а зміни в поведінці користувача — як перехід між цими станами.

Однією з ключових особливостей марковських моделей є їхня здатність швидко реагувати на зміни у поведінкових патернах. Наприклад, система може аналізувати послідовність дій користувача під час входу — введення пароля, підтвердження OTP-кодом або використання біометричних даних — і автоматично виявляти відхилення від звичного сценарію. Такі відхилення можуть свідчити про потенційну загрозу, зокрема спробу злому або використання викрадених даних. Завдяки цьому система може динамічно підвищувати рівень безпеки, наприклад, вимагати додаткові кроки автентифікації, якщо поведінка користувача відрізняється від типового шаблону [23].

Адаптивність марковських моделей також виявляється у здатності системи навчатися на основі накопичених даних. З часом система може покращувати точність прогнозування, оновлюючи ймовірності переходів між станами залежно від нових прикладів поведінки користувачів. Наприклад, якщо користувач починає частіше входити в систему з нових пристроїв або місць, модель адаптується до цього патерну, знижуючи рівень тривоги для подібних ситуацій у майбутньому. Це

дозволяє уникнути хибних спрацьовувань і покращити досвід використання для легітимних користувачів.

Додатково, марковські моделі можуть інтегруватися з іншими алгоритмами аналізу поведінки, такими як кластеризація або методи машинного навчання. Це розширює можливості адаптації, дозволяючи системі враховувати не лише поточний стан користувача, а й його історичні дії. Такий підхід особливо ефективний для виявлення складних сценаріїв атак, коли зловмисник поступово імітує поведінку справжнього користувача.

Таким чином, адаптивність марковських моделей є критично важливою для забезпечення ефективності та безпеки багатофакторної автентифікації. Вони не лише спрощують побудову систем прогнозування, а й дозволяють динамічно реагувати на зміни у поведінці користувачів, підвищуючи стійкість системи до нових загроз і забезпечуючи її здатність до самонавчання в процесі експлуатації.

2.2 Математична модель та алгоритм використання ланцюгів Маркова для багатофакторної автентифікації

Основою розробки моделі багатофакторної автентифікації з використанням ланцюгів Маркова є формалізація етапів процесу автентифікації у вигляді станів системи та ймовірностей переходу між ними. Цей підхід дозволяє оцінювати ефективність системи, враховуючи різні фактори ризику та поведінкові шаблони користувачів.

Модель включає визначення множини станів

$$S = \{s_1, s_2, \dots, s_n\}, \tag{2.1}$$

які відповідають етапам автентифікації. Наприклад, стан s_1 може відповідати введенню пароля, стан s_2 – перевірці через SMS-код, а стан s_n – успішному

завершенню автентифікації. Кожен стан має свою ймовірність переходу до іншого стану, що описується матрицею переходів P, де P_{ij} – це ймовірність переходу зі стану s_i до стану s_j.Формально, процес описується як марковський ланцюг із часовими дискретами, де:

$$P\left(X_{n+1} = s_j \middle| X_n = s_i, X_{n-1}, \dots, X_0\right) = P(X_{n+1} = s_j | X_n = s_i) \quad (2.2)$$

де X_n – стан системи на кроці n.

Процес багатофакторної автентифікації розбивається на етапи, які називаються станами. Наприклад:

- S_0: Початковий стан (користувач ще не ввів жодних даних).
- S_1: Введення пароля.
- S_2: Перевірка одноразового коду (SMS, email або токен).
- S_3: Перевірка біометричних даних (за необхідності).
- $S_{success}$: Успішна автентифікація.
- S_{fail}: Провал автентифікації (наприклад, після кількох невдалих спроб).

Матриця ймовірностей переходів P описує ймовірність переходу від одного стану до іншого (Рис. 2.1). Для кожного стану S_i визначаються можливі переходи в інші стани S_j, а їхні ймовірності записуються у вигляді:

$$P = \begin{pmatrix} P(S_0 \to S_0) & P(S_0 \to S_1) & P(S_0 \to S_{fail}) \\ P(S_1 \to S_1) & P(S_1 \to S_2) & P(S_1 \to S_{fail}) \\ P(S_2 \to S_2) & P(S_2 \to S_3) & P(S_2 \to S_{fail}) \\ P(S_3 \to S_3) & P(S_3 \to S_{success}) & P(S_3 \to S_{fail}) \end{pmatrix} \quad (2.3)$$

Де:

- $P(S_i \to S_j)$ – ймовірність переходу зі стану S_i до S_j.
- Кожен рядок матриці повинен задовольняти умову: $\sum_i P\left(S_i \to S_j\right) = 1$.

Наприклад:

- $P(S_0 \rightarrow S_1) = 0.9$ (користувач успішно вводить пароль у 90% випадків).
- $P(S_1 \rightarrow S_{fail}) = 0.1$ (у 10% випадків пароль введено неправильно).

Для розрахунку ймовірності успішної або невдалої автентифікації застосовуються ітераційні методи. Початковий вектор станів π_0 визначає ймовірності перебування у кожному стані на початку процесу:

$$\pi_0 = [1,0,0,0,0,0] \tag{2.4}$$

Це означає, що процес починається у стані S_0. Потім ймовірності для наступного кроку обчислюються за допомогою множення початкового вектора π_0 на матрицю P, а далі процедура повторюється:

$$\pi_1 = \pi_0 \cdot P$$
$$\pi_2 = \pi_1 \cdot P \tag{2.5}$$

Цей процес продовжується доти, доки розподіл ймовірностей не стабілізується або доки не буде досягнуто фінального стану $S_{success}$ або S_{fail}.

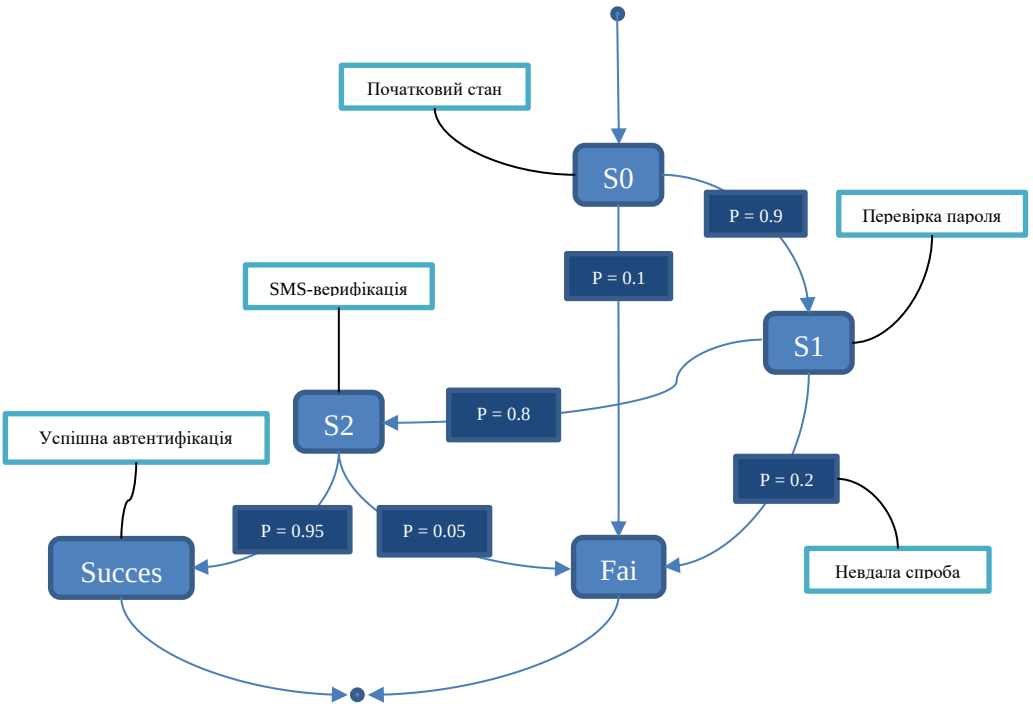


Рисунок 2.1 – Візуалізація ланцюга Маркова для процесу автентифікації.

Нехай:

- $P(S_0 \to S_1) = 0.9, \mathrm{P}(S_0 \to S_{fail}) = 0.1$
- $P(S_1 \to S_2) = 0.8, \mathrm{P}(S_1 \to S_{fail}) = 0.2$
- $P(S_2 \to S_{success}) = 0.95, \mathrm{P}(S_2 \to S_{fail}) = 0.05$

Початковий вектор:

$$\pi_0 = [1,0,0,0,0,0]$$

Після першого кроку:

$$\pi_1 = \pi_0 \cdot P = [0.9,0.0,0.0,0.0,0.1]$$

Після другого кроку:

$$\pi_2 = \pi_1 \cdot P = [0.0,0.72,0.0,0.0,0.28]$$

Таким чином, ймовірність перебування у стані S_{fail} після другого етапу становить 28%, а в станіS_2(перевірка SMS-коду) – 72%.

Отримані ймовірності дозволяють оцінити загальну ефективність системи автентифікації, а також визначити "вузькі місця". Наприклад, якщо ймовірність невдачі значно зростає на певному етапі, необхідно переглянути механізм його реалізації (наприклад, додати інструкції для користувачів або змінити спосіб передачі кодів) [8].

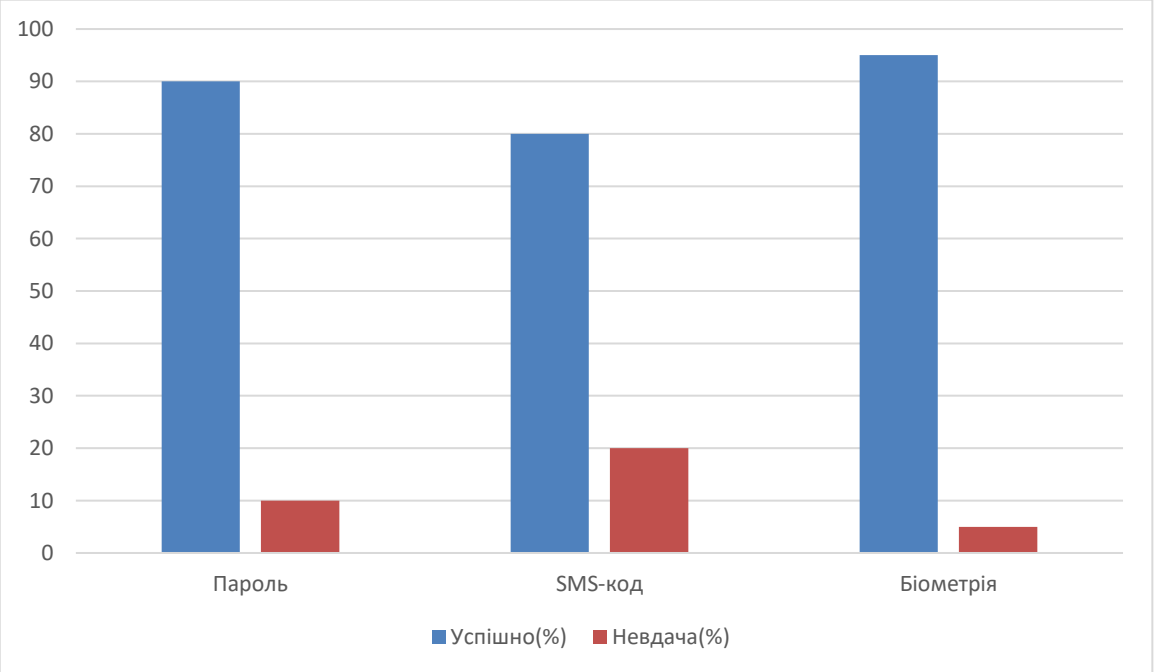


Рисунок 2.2 – Успішність проходження етапів автентифікації.

Завдяки ланцюгам Маркова модель можна динамічно адаптувати до поведінки користувачів. Наприклад:

Якщо користувач часто допускає помилки на етапі введення пароля, система може автоматично змінювати ймовірності переходів, додаючи більше етапів перевірки. Для нових користувачів матриця ймовірностей може базуватися на статистичних даних, а згодом уточнювати відповідно до індивідуальної поведінки.

На основі цієї моделі розробляється алгоритм, що включає наступні етапи:

1. *Ініціалізація системи*: на початку визначаються всі можливі стани та ймовірності переходів між ними. Наприклад, ймовірність успішного введення пароля базується на статистиці помилок користувачів, а ймовірність успішного введення SMS-коду залежить від доступності мобільного пристрою.
2. *Аналіз вхідних даних*: система отримує дані від користувача, такі як пароль, біометричні параметри або SMS-код. Кожне з цих дій змінює поточний стан системи.
3. *Обчислення ймовірностей*: на кожному етапі розраховується ймовірність переходу до наступного стану. Якщо ймовірність успішного завершення всіх етапів перевищує визначений поріг, доступ надається; інакше система генерує запит на повторну автентифікацію або додаткову перевірку.
4. *Реакція на загрози*: у разі виявлення аномальної поведінки (наприклад, невдалі спроби входу чи спроби з невідомого пристрою) система може змінити ймовірності переходів або додати додаткові етапи перевірки.

Особливістю використання ланцюгів Маркова є можливість моделювання динамічних систем, які адаптуються до змінних умов. Наприклад, якщо для певного користувача часто виникають помилки на етапі введення пароля, система може зменшити довіру до цього етапу та посилити інші перевірки.

Алгоритм також дозволяє оцінювати загальну ефективність системи, визначаючи ймовірність успішної автентифікації для різних сценаріїв. Це сприяє оптимізації параметрів моделі та підвищенню її надійності.

Застосування ланцюгів Маркова до систем багатофакторної автентифікації дозволяє вдосконалити процес прогнозування і оцінки кожного етапу. Одним з аспектів цього підходу є оптимізація матриці ймовірностей переходів.

Ключовим елементом марковської моделі є матриця ймовірностей переходів P, яка відображає імовірності переходу між станами.

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{pmatrix} \tag{2.6}$$

Тут p_{ij} — це ймовірність переходу зі стану S_i до стану S_j , причому:

$$\sum_{j=1}^{n} p_{ij} = 1, \forall i \in \{1,2, \dots, n\} \tag{2.7}$$

Формула (2.7) означає, що для будь-якого стану iii сума ймовірностей переходів у всі можливі стани j дорівнює 1. Це відображає властивість нормалізації матриці ймовірностей переходів у марковському процесі, де всі можливі переходи з одного стану покривають повний простір ймовірностей.

Наприклад, якщо стан S_1 відповідає успішному введенню пароля, а S_2 — підтвердженню через SMS, то p_{12} визначає ймовірність успішного переходу на наступний етап. Для обчислення цих ймовірностей використовуються історичні дані:

$$p_{ij} = \frac{\text{кількість переходів з} S_i \text{ до } S_j}{\text{загальна кількість спроб у стані } S_i} \tag{2.8}$$

Її коректна побудова забезпечує точність прогнозування ефективності багатофакторної автентифікації. Для створення такої матриці використовуються історичні дані системи, зокрема логи про успішні та невдалі спроби автентифікації на кожному етапі. Наприклад, введення пароля, підтвердження через SMS або біометричну перевірку. Зібрані дані дозволяють визначити, наскільки імовірно, що користувач перейде від одного стану (наприклад, успішного введення пароля) до іншого (успішного проходження біометричного контролю).

Повна марковська модель автентифікації має враховувати різноманітні зовнішні фактори, які можуть впливати на проходження кожного етапу. Серед таких факторів можна виділити час, який користувач витрачає на виконання певного етапу, частоту використання системи і рівень складності автентифікаційного завдання. Наприклад, нові користувачі частіше припускаються помилок, ніж ті, які постійно працюють із системою. Складні паролі також можуть збільшувати ймовірність помилок, але водночас підвищують рівень безпеки.

Додаткові фактори, такі як час, складність завдань або досвід користувачів, можуть бути враховані через модифіковані ймовірності переходів. Наприклад, враховуючи ваговий коефіцієнт w_k, ймовірності переходів коригуються так:

$$p_{ij}^{\text{кориговане}} = p_{ij} \cdot w_k \tag{2.9}$$

де w_k визначається за допомогою аналітики, наприклад:

$$w_k = \frac{\text{фактична частота успіху для групи } k}{\text{середня частота успіху для всіх користувачів}} \tag{2.10}$$

Моделювання сценаріїв ризику дозволяє оцінити, наскільки система здатна протистояти конкретним загрозам, а також визначити слабкі місця, які потребують доопрацювання. Крім того, цей підхід сприяє кращому розумінню того, які фактори

найбільше впливають на безпеку.

У сценарії ризику вводиться додатковий стан S_{attack}, що описує дії зловмисника. Наприклад, матриця переходів може набувати вигляду:

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} & p_{1attack} \\ p_{21} & p_{22} & \cdots & p_{2n} & p_{2attack} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} & p_{nattack} \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \quad (2.11)$$

Ймовірність переходу до стану S_{attack} залежить від рівня загрози, який можна оцінити через такі параметри, як частота помилок користувача, спроби доступу з підозрілих IP-адрес тощо.

2.3 Моделювання сценаріїв невизначеності

Моделювання сценаріїв невизначеності є важливим компонентом у створенні багатофакторної автентифікації (МФА), яка має справлятися з різноманітними загрозами та нестандартними ситуаціями. У реальних умовах користувачі, а також потенційні зловмисники, можуть поводитися непередбачувано. Наприклад, введення частково правильного пароля, доступ із нового пристрою чи підозрілі дії можуть створювати умови невизначеності. Ланцюги Маркова забезпечують ефективний інструмент для моделювання таких процесів, дозволяючи оцінювати ймовірності переходів між різними станами системи [6].

Для побудови моделі необхідно визначити ключові стани, які представляють етапи автентифікації. Наприклад, початковим станом може бути введення пароля, наступним – підтвердження через SMS, далі – перевірка біометричних даних, і, нарешті, завершальні стани: успішна автентифікація або відмова в доступі. Крім того, додаються проміжні стани, що відображають невизначеність, наприклад,

часткову відповідність біометричних даних або невчасне підтвердження через код [25].

Модель дозволяє відображати будь-який сценарій: від стандартного, де користувач діє передбачувано, до атипового, де зовнішні чинники (наприклад, використання нового пристрою або підозріла геолокація) ускладнюють процес автентифікації.

Основою моделі є ймовірності переходів між станами. Для кожного переходу визначається його ймовірність, яка залежить від історичних даних або статистичних моделей. Наприклад, ймовірність переходу від стану "введення пароля" до "успішного входу" може становити 80%, якщо пароль правильний, або 20%, якщо він неправильний. Аналогічно, ймовірність підтвердження через SMS може залежати від часу, що залишився до закінчення дії коду, та поведінкових патернів користувача.

Такі моделі дозволяють враховувати зовнішні фактори, які можуть впливати на процес автентифікації. Наприклад, якщо користувач входить у систему вночі або з незвичайного пристрою, ймовірність успішного завершення автентифікації без додаткових перевірок знижується.

Для визначення ймовірностей переходів між станами використовується базова формула:

$$P(S_{i+1}|S_i) = \frac{N(S_i \to S_{i+1})}{N(S_i)} \tag{2.12}$$

де $N(S_i \to S_{i+1})$ – кількість переходів зі стану S_i до стану S_{i+1}, а $N(S_i)$ – загальна кількість перебувань у стані S_i.

Обчислення загальної ймовірності проходження процесу автентифікації виглядає так:

$$P(S_n) = P(S_0) \cdot P(S_1|S_0) \cdot P(S_2|S_1) \cdot \ldots \cdot P(S_n|S_{n-1}) \tag{2.13}$$

де $P(S_0)$ - початкова ймовірність перебування у початковому стані.

У випадках, коли система не може з достатньою впевненістю визначити успіх автентифікації, вводяться додаткові заходи. Наприклад, якщо ймовірність успішного входу знаходиться на межі (50-60%), користувачу пропонується виконати ще один крок автентифікації, наприклад, відповісти на контрольне запитання або підтвердити вхід через додатковий пристрій.

Для оцінки ризиків використовується аналіз сценаріїв на основі ймовірностей переходів. Якщо система виявляє невідповідність у поведінці користувача (наприклад, кілька спроб ввести пароль), це може призвести до збільшення рівня ризику, викликавши блокування облікового запису або підвищення рівня автентифікації.

Для моделювання поведінки потенційних атакуючих система використовує сценарії з низькими початковими ймовірностями успішного проходження. Зловмисники, ймовірно, намагатимуться вгадати пароль або використовувати інші методи. У таких випадках ймовірності переходів значно знижуються, а система може автоматично включати додаткові фактори автентифікації.

Моделювання сценаріїв невизначеності з використанням ланцюгів Маркова дозволяє динамічно адаптувати систему автентифікації до нових загроз і забезпечувати оптимальний баланс між зручністю використання та безпекою. Для моделювання цього процесу використовується матриця ймовірностей переходів. Елементи цієї матриці визначають ймовірності зміни стану залежно від поточної поведінки. Наприклад:

$$P = \begin{pmatrix} 0.9 & 0.1 & 0 \\ 0.7 & 0.2 & 0.1 \\ 0.5 & 0 & 0.5 \end{pmatrix} \tag{2.14}$$

де стани можуть відповідати:

- S_0 – нормальна поведінка користувача.
- S_1 – підозріла активність (кілька невдалих спроб входу).

- S_2 – заблокований обліковий запис.

Коли система виявляє потенційні дії зловмисника, вона адаптується, змінюючи процес автентифікації:

- *Додавання етапів перевірки*. Наприклад, після кількох невдалих спроб входу система може вимагати додатковий код, надісланий через SMS, або підтвердження через додаток.
- *Підвищення складності перевірок*. У разі підозрілої активності система може запросити кілька біометричних факторів замість одного (наприклад, одночасно відбиток пальця та розпізнавання обличчя).
- *Блокування доступу*. Якщо ймовірність успішного проходження автентифікації залишається низькою, система блокує обліковий запис до додаткової верифікації користувача.

Модель базується на представленні процесу автентифікації у вигляді марковського ланцюга, де кожен етап є станом системи, а переходи між станами мають певні ймовірності.

Стани моделі:

1. S_0: Початковий стан (запит доступу).
2. S_1: Перевірка пароля.
3. S_2: Перевірка одноразового коду (SMS, email, токен).
4. S_3: Перевірка біометричних даних.
5. $S_{success}$: Успішна автентифікація.
6. S_{fail}: Невдача автентифікації.

$$P = \begin{pmatrix} P(S_0 \to S_0) & P(S_0 \to S_1) & P(S_0 \to S_{fail}) & 0 & 0 & 0 \\ 0 & P(S_1 \to S_1) & P(S_1 \to S_2) & P(S_1 \to S_{fail}) & 0 & 0 \\ 0 & 0 & P(S_2 \to S_2) & P(S_2 \to S_3) & P(S_0 \to S_0) & 0 \\ 0 & 0 & 0 & P(S_3 \to S_3) & P(S_0 \to S_0) & P(S_3 \to S_{fail}) \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.15)$$

Де:

- $P(S_i \to S_j)$ — ймовірність переходу зі стану S_i до стану S_j.
- Суми ймовірностей у кожному рядку дорівнюють 1.

Загальна ймовірність успішної автентифікації визначається як добуток ймовірностей переходів між станами, які ведуть до стану $S_{success}$:

$$S_{success} = P(S_0 \to S_1) \cdot P(S_1 \to S_2) \cdot P(S_2 \to S_3) \cdot P(S_3 \to S_{success}) \quad (2.16)$$

Графічне представлення моделі:

- $S_0 \to S_1 \to S_2 \to S_3 \to S_{success}$
- S_1, S_2, S_3також можуть мати переходи до S_{fail}, що відображає невдачу.

2.4 Висновки до другого розділу

Розроблено модель багатофакторної автентифікації, що базується на використанні ланцюгів Маркова, яка враховує ймовірності успішного або невдалого проходження кожного етапу автентифікації. Центральним елементом моделі є детальна формалізація процесу аутентифікації через стани та ймовірності переходів між ними. Це дозволяє не лише зберігати високу безпеку системи, але й адаптувати її до різних сценаріїв, включаючи варіації умов, зовнішні загрози або змінювані поведінкові фактори користувачів.

Модель забезпечує високу гнучкість та адаптивність завдяки включенню різноманітних параметрів, які можуть змінюватися в залежності від зовнішніх умов, таких як поведінка користувача або дії зловмисників. Використання матриці ймовірностей переходів дозволяє виявити слабкі місця в системі, ідентифікувати потенційні загрози та своєчасно коригувати рівень довіри користувача до системи на основі змінюваних умов. Це, в свою чергу, підвищує рівень безпеки через динамічну адаптацію до різноманітних умов і допомагає протистояти новим викликам.

Модель багатофакторної автентифікації, побудована на основі ланцюгів Маркова, включає в себе кілька важливих станів, через які проходить користувач під час процесу автентифікації:

1. Початковий стан (запит доступу).
2. Перевірка пароля.
3. Перевірка одноразового коду (SMS, email, токен).
4. Перевірка біометричних даних.
5. Успішна автентифікація.
6. Невдача автентифікації.

Матриця ймовірностей переходів служить для опису переходів між цими станами та для розрахунку ймовірності успішного проходження кожного етапу. Загальна ймовірність успішної автентифікації розраховується як добуток ймовірностей переходів між відповідними станами.

Моделювання різноманітних сценаріїв ризику та невизначеності з використанням ланцюгів Маркова дозволяє розробити алгоритми для адаптації системи до дій зловмисників або змінних умов використання, наприклад, при зміні зовнішніх загроз або поведінкових патернів користувачів. Впровадження таких адаптивних механізмів сприяє зниженню ризику компрометації системи та створенню оптимального балансу між зручністю користування та рівнем безпеки.

Отримані результати підтверджують ефективність застосування ланцюгів Маркова в багатофакторній автентифікації для значного підвищення надійності системи. Ця модель може бути успішно інтегрована в сучасні інформаційні системи, де забезпечить ефективну протидію загрозам, одночасно покращуючи користувацький досвід за рахунок її гнучкості та адаптивності до різних умов.

РОЗДІЛ 3.
РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ АЛГОРИТМУ ДЛЯ ОПТИМІЗАЦІЇ АВТЕНТИФІКАЦІЙНИХ ПРОЦЕСІВ

3.1 Порівняння найпопулярніших мов програмування

Реалізація алгоритму багатофакторної автентифікації вимагає вибору відповідної мови програмування, яка дозволить ефективно працювати як із серверною, так і з клієнтською частиною системи. Важливими критеріями при виборі є продуктивність, масштабованість, гнучкість у роботі з базами даних, підтримка математичних моделей та безпечність. У цьому розділі проведено аналіз кількох популярних мов програмування (Табл. 3.1), які часто використовуються для розробки подібних систем, з метою визначення найкращого рішення для даного проєкту [21].

Python є одним із найпопулярніших виборів для створення алгоритмів, що базуються на математичних розрахунках. Його бібліотеки, такі як NumPy, SciPy та Pandas, забезпечують зручний інструментарій для роботи з даними та статистичними моделями. Крім того, Python підтримує машинне навчання, що робить його ідеальним для побудови адаптивних алгоритмів безпеки. Проте Python поступається іншим мовам у продуктивності для веб-додатків і вимагає додаткових фреймворків, таких як Django або Flask, для створення інтерактивних інтерфейсів. Це може збільшити складність інтеграції клієнтської та серверної частин системи.

Java є мовою високого рівня, яка відзначається стабільністю та безпекою. Вона широко використовується для створення корпоративних додатків, які мають високі вимоги до продуктивності та масштабованості. Завдяки вбудованій підтримці багатопоточності та об’єктно-орієнтованій структурі Java добре підходить для великих проєктів. Проте розробка на Java займає більше часу через складність налаштувань і створення інтерфейсів. Крім того, вона вимагає значних

обчислювальних ресурсів, що може ускладнити її використання в невеликих та середніх системах.

Ще одним популярним рішенням є *JavaScript* з використанням середовища виконання Node.js. Його головною перевагою є можливість роботи як на клієнтському, так і на серверному боці, що забезпечує швидкий обмін даними між ними. Асинхронна модель обробки запитів дозволяє створювати швидкі веб-додатки в режимі реального часу. Однак JavaScript менш пристосований для складних математичних обчислень, які потрібні для роботи з марковськими моделями. Крім того, забезпечення належного рівня безпеки на JavaScript вимагає додаткових налаштувань, що може ускладнити розробку системи автентифікації.

PHP, натомість, є оптимальним вибором для створення веб-додатків, особливо тих, що взаємодіють із базами даних. Ця мова програмування широко використовується завдяки своїй простоті, гнучкості та потужним функціональним можливостям. Завдяки вбудованій підтримці роботи з HTML та MySQL, PHP дозволяє швидко реалізувати серверну частину додатку, що є важливим для динамічних веб-сайтів і складних інформаційних систем.

Крім того, PHP має широкий набір бібліотек для роботи з шифруванням та автентифікацією, таких як OpenSSL і bcrypt, які забезпечують високий рівень безпеки для сучасних додатків. Важливо зазначити, що PHP є більш економічно вигідним рішенням через його підтримку більшістю хостинг-платформ, зокрема тих, що пропонують недорогі тарифи [20].

Таблиця 3.1

Переваги та недоліки мов програмування для задач у веб-аналітиці.

Мова	Переваги	Недоліки
Python	- Зручний синтаксис і легкість вивчення. - Потужні бібліотеки для математичних розрахунків - Підтримка машинного навчання	- Низька продуктивність у порівнянні з іншими мовами. - Потребує фреймворків (Django, Flask) для веб-розробки.

Продовження таблиці 3.1.

Мова	Переваги	Недоліки
Java	- Висока продуктивність і стабільність. - Підтримка багатопоточності. - Відмінний рівень безпеки. - Добре підходить для масштабованих корпоративних систем.	- Складність налаштування та розгортання проєктів. - Тривалий час розробки через громіздкість коду. - Потребує великих обчислювальних ресурсів.
JavaScript (Node.js)	- Швидка асинхронна обробка запитів у реальному часі. - Підтримка роботи як на клієнтській, так і на серверній частині. - Висока популярність і велика кількість бібліотек.	- Складнощі з реалізацією складних математичних розрахунків. - Додаткові налаштування для забезпечення безпеки. - Залежність від сторонніх модулів, що підвищує ризик вразливостей.
PHP	- Простота інтеграції з HTML і базами даних (MySQL). - Велика кількість бібліотек для роботи з безпекою (OpenSSL, bcrypt). - Підтримка більшістю хостингів і легке налаштування середовища. - Швидка розробка і низька вартість впровадження.	- Менша продуктивність для обчислювально-інтенсивних завдань порівняно з Java. - Обмежені можливості для роботи з машинним навчанням. - Потребує додаткового захисту при обробці конфіденційних даних.

На основі аналізу переваг і недоліків розглянутих мов програмування було вирішено використовувати PHP для реалізації алгоритму. Це пояснюється його здатністю ефективно працювати в середовищі веб-додатків, простотою інтеграції з базами даних та наявністю готових рішень для забезпечення безпеки. PHP забезпечує необхідний баланс між швидкістю розробки, масштабованістю та функціональністю, що робить його оптимальним вибором для створення системи багатофакторної автентифікації [19].

Бази даних є невіддільною частиною системи, яка зберігає всю необхідну інформацію для роботи алгоритму. Основні функції бази даних включають:

- *Збереження матриць ймовірностей переходів*: для кожного сценарію моделювання зберігається окрема матриця, що дозволяє адаптувати модель до конкретних умов.
- *Логи дій користувачів*: база даних фіксує історію входів, успіхів і невдач, що є основою для подальшого аналізу та оптимізації системи.
- *Інформація про сценарії*: база містить змодельовані сценарії атак або атипової поведінки користувачів, що використовуються для тестування алгоритму.

Для реалізації бази даних обрано MySQL, яка забезпечує швидкий доступ до даних та ефективну роботу з великими обсягами інформації. Структура бази передбачає кілька таблиць: для зберігання станів, ймовірностей переходів, результатів тестувань та інших параметрів [1].

3.3 Архітектура системи та бази даних

Архітектура системи багатофакторної автентифікації розроблена з урахуванням вимог до надійності, масштабованості та безпеки. Вона має модульну структуру, що забезпечує зручність підтримки, оновлення та адаптації системи до нових умов експлуатації. Основними елементами архітектури є серверна частина, клієнтський інтерфейс, база даних і алгоритмічний модуль для роботи з ланцюгами Маркова.

Серверна частина реалізована на PHP і виконує основні функції логіки роботи системи. Вона відповідає за обробку запитів користувачів, перевірку їхніх облікових даних, генерування кодів підтвердження та управління сесіями автентифікації. Сервер також здійснює взаємодію з базою даних, зберігаючи інформацію про користувачів і результати автентифікації.

Для зберігання інформації про користувачів, їхні сесії, результати автентифікації та логування подій використовується реляційна база даних MySQL. Структура бази даних включає такі основні таблиці:

- users – зберігає дані користувачів, включаючи логіни, хешовані паролі та налаштування автентифікації.
- sessions – відображає активні та завершені сесії автентифікації, включаючи їхній статус і час завершення.
- logs – містить записи про спроби входу та дії користувачів для подальшого аналізу.

Модуль обробки алгоритму реалізує математичну модель на основі ланцюгів Маркова. Його завдання – обчислення ймовірностей переходів між етапами автентифікації та прогнозування результату процесу. Модуль працює з матрицею переходів і застосовує алгоритми обчислення ймовірностей для виявлення потенційних ризиків.

Система автентифікації побудована на основі ланцюгів Маркова, що дозволяє представити весь процес як послідовність взаємопов'язаних станів. Розглянемо детально кожен етап роботи системи та його особливості.

Початковий етап автентифікації починається із запиту користувача на вхід до системи. На цьому етапі користувач вводить свої облікові дані - логін та пароль. Система отримує ці дані та готує їх до подальшої обробки. Важливо відзначити, що всі дані на цьому етапі передаються у зашифрованому вигляді для забезпечення безпеки [28].

Після отримання облікових даних система переходить до етапу перевірки пароля. На цьому етапі відбувається порівняння введеного пароля з хешованим значенням, що зберігається в базі даних. Система передбачає можливість декількох спроб введення пароля у випадку помилки. Проте кількість таких спроб обмежена для запобігання брутфорс-атакам. Кожна невдала спроба фіксується системою та впливає на подальший розрахунок ймовірностей.

Третій етап включає перевірку додаткових факторів автентифікації. Система

може вимагати від користувача підтвердження через SMS-код, який надсилається на зареєстрований номер телефону, або через біометричні дані, такі як відбиток пальця чи сканування обличчя. Цей етап забезпечує додатковий рівень безпеки та суттєво знижує ризик несанкціонованого доступу.

На етапі розрахунку ймовірностей система аналізує всі попередні дії користувача. Алгоритм ланцюгів Маркова обробляє такі параметри як успішність введення пароля, час між спробами, правильність введення додаткових факторів автентифікації. На основі цих даних розраховується загальна ймовірність того, що автентифікація виконується легітимним користувачем.

Фінальний етап - прийняття рішення про надання доступу. Система порівнює розраховану ймовірність із встановленим пороговим значенням. Якщо ймовірність перевищує поріг, користувач отримує доступ до системи. У протилежному випадку система може вимагати додаткової верифікації або повністю заблокувати спробу входу. Важливо зазначити, що порогові значення можуть динамічно змінюватися залежно від рівня безпеки, необхідного для конкретного користувача або групи користувачів.

Така структура забезпечує гнучкий та надійний процес автентифікації, який може адаптуватися до різних сценаріїв використання та рівнів безпеки. Використання ланцюгів Маркова дозволяє системі враховувати історію попередніх спроб автентифікації та приймати більш обґрунтовані рішення щодо надання доступу [7].

Система автентифікації спроектована з урахуванням можливості масштабування та необхідності адаптації до змінних умов експлуатації. Розглянемо детально технічні аспекти, що забезпечують ці важливі характеристики системи.

Масштабованість системи забезпечується завдяки використанню серверних технологій PHP та системи управління базами даних MySQL. Така архітектура дозволяє гнучко розподіляти навантаження між множиною серверів. При зростанні кількості користувачів або збільшенні інтенсивності запитів, система може бути розширена шляхом додавання нових серверів до пулу обробки даних. PHP-скрипти

можуть бути розгорнуті на декількох серверах одночасно, а MySQL підтримує реплікацію даних, що забезпечує високу доступність та відмовостійкість системи [2].

Особливу увагу приділено адаптивності системи до нових викликів безпеки. Ключовим елементом цієї адаптивності є динамічна матриця переходів у ланцюгах Маркова. Система постійно накопичує та аналізує дані про поведінку користувачів під час сесій автентифікації. На основі цього аналізу відбувається автоматичне коригування ймовірностей переходів між різними станами системи. Якщо, наприклад, виявляється новий паттерн атак, система може автоматично підвищити вимоги до додаткової верифікації для подібних сценаріїв входу.

Така гнучка архітектура дозволяє системі не тільки ефективно справлятися зі зростаючим навантаженням, але й постійно вдосконалювати механізми захисту на основі реальних даних про спроби автентифікації. Це робить систему надійним та сучасним рішенням для забезпечення безпечного доступу користувачів.

Система автентифікації реалізує комплексний підхід до забезпечення безпеки, що охоплює різні аспекти захисту даних та запобігання несанкціонованому доступу. Розглянемо детально кожен з основних компонентів системи безпеки.

Фундаментальним елементом безпеки є надійне шифрування даних. Всі паролі користувачів зберігаються в базі даних виключно у вигляді хешів, згенерованих за допомогою алгоритму bcrypt. Цей алгоритм спеціально розроблений для хешування паролів та налаштовуваний фактор складності, що робить атаки перебором практично неможливими. Передача даних між клієнтською частиною системи та сервером здійснюється через захищений протокол HTTPS, який забезпечує шифрування всього трафіку та захист від перехоплення даних [27].

Система включає потужний механізм захисту від різних типів атак. Для протидії brute-force атакам реалізовано динамічне обмеження кількості спроб входу з однієї IP-адреси та автоматичне блокування облікових записів при перевищенні

встановлених лімітів. Захист від SQL-ін'єкцій забезпечується через використання параметризованих запитів та ретельну валідацію всіх вхідних даних. Система також веде детальні логи активності користувачів, що дозволяє адміністраторам відстежувати та аналізувати потенційно небезпечні дії.

Особлива увага приділяється виявленню та обробці аномальної поведінки користувачів. Система постійно аналізує такі параметри як геолокація, тип пристрою, час доступу та патерни використання. При виявленні нестандартних сценаріїв, наприклад, спроби входу з нового пристрою або незвичного місця розташування, автоматично вмикаються додаткові механізми перевірки. Це може включати відправку коду підтвердження через SMS, використання біометричної автентифікації або запит додаткової інформації для підтвердження особи користувача.

Система безпеки також включає механізми моніторингу та сповіщення адміністраторів про потенційні загрози в режимі реального часу. Це дозволяє оперативно реагувати на спроби несанкціонованого доступу та вживати необхідних заходів для захисту системи та даних користувачів.

Створено таблицю для зберігання логів автентифікації користувачів (Рис.3.1). Кожен запис у таблиці буде представляти одну подію, пов'язану з автентифікацією (наприклад, введення пароля, помилка при введенні OTP, натискання на кнопку "відновити пароль" тощо). Кожен такий запис повинен містити інформацію про час події, тип події, ідентифікатор користувача, а також інші додаткові параметри, які дозволяють проаналізувати поведінку користувача [26].

```
CREATE TABLE authentication_logs (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    event_time DATETIME NOT NULL,
    event_type VARCHAR(255) NOT NULL,
    event_details TEXT,
    user_ip VARCHAR(45),
    user_agent VARCHAR(255),
    previous_event_id INT,
    FOREIGN KEY (user_id) REFERENCES users(id),
    INDEX(user_id, event_time)
);
```

Рисунок 3.1 – Структура таблиці бази даних

Поля таблиці:

- id: Унікальний ідентифікатор події, що дозволяє уникнути дублювання та зберігати точну інформацію про кожну подію.
- user_id: Ідентифікатор користувача, який виконує певну дію. Це поле необхідне для відслідковування всіх подій конкретного користувача.
- event_time: Час, коли подія відбулася. Це поле зберігається у стандартному форматі для точного визначення послідовності подій.
- event_type: Тип події (наприклад, "password_attempt", "otp_failure", "password_reset", "login_success", "login_failure"). Це поле дозволяє класифікувати події за типом, що є критично важливим для побудови ланцюгів Маркова.
- event_details: Додаткові дані, які можуть пояснити подію. Наприклад, при неправильному введенні пароля це може бути "incorrect password", або "too many failed attempts" при декількох невдалих спробах.
- user_ip: IP-адреса користувача, що може бути корисно для аналізу поведінки та виявлення аномалій (наприклад, кілька спроб входу з одного IP).
- user_agent: Інформація про браузер або пристрій, з якого користувач намагається здійснити аутентифікацію. Це може допомогти для виявлення автоматичних спроб (наприклад, спроби з не типових пристроїв).
- previous_event_id: Це поле зберігає ID попередньої події, що дозволяє побудувати ланцюги Маркова для моделювання послідовності дій користувача.

Приклад логів (Табл.3.2),(Табл.3.3), які будуть зберігатися в таблиці authentication_logs для аналізу поведінки користувача під час автентифікації.

Таблиця 3.2

Приклад 1: Введення неправильного пароля

id	user_id	event_id	event_type	event_details	user_ip	user_agent	previous_event_id
1	101	2024-12-26 08:30:00	password_attempt	incorrect password	192.168.0.1	Mozilla/5.0 (Windows)	NULL

Таблиця 3.3

Приклад 1:Введення неправильного коду OTP

id	user_id	event_id	event_type	event_details	user_ip	user_agent	previous_event_id
2	101	2024-12-26 08:32:00	otp_failure	incorrect OTP code	192.168.0.1	Mozilla/5.0 (Windows)	1

Ці логи дозволяють побудувати послідовності подій користувача, що є важливим для аналізу через ланцюги Маркова. Завдяки полю previous_event_id, можна відстежувати, як кожна подія є наступною за попередньою, що дозволяє визначити ймовірності переходів між станами (наприклад, перехід від неправильного введення пароля до спроби відновлення пароля).

Ключовим аспектом цієї системи є здатність будувати ланцюги Маркова, які дозволяють моделювати ймовірності переходів між різними етапами аутентифікації. Для кожного користувача ми можемо побудувати послідовність подій, де кожна подія є переходом між станами, такими як "неправильний пароль", "неправильний OTP", "відновлення пароля" тощо.

Завдяки полю previous_event_id, ми можемо відслідковувати кожну подію як частину послідовності, що дозволяє обчислювати ймовірності того, який етап аутентифікації може бути наступним для конкретного користувача. Наприклад, якщо користувач кілька разів вводить неправильний пароль, ми можемо обчислити ймовірність того, що він спробує відновити пароль, або що він повторно введе неправильний пароль.

3.3 Імплементація алгоритму ланцюгів Маркова для аналізу поведінки користувачів під час автентифікації

Алгоритм ланцюгів Маркова полягає у побудові моделі, де кожен стан описує конкретну подію (наприклад, успішне введення пароля, неправильний код OTP тощо). Ймовірність переходу від одного стану до іншого визначається на основі

частоти таких переходів у зібраних даних. Ланцюг Маркова має таку властивість: ймовірність переходу залежить лише від поточного стану, а не від історії попередніх подій [14].

Основними етапами імплементації є:

1. Збір та попередня обробка даних: Збирання логів користувачів з таблиці, яка зберігає події під час аутентифікації.
2. Побудова матриці ймовірностей переходів: Розрахунок ймовірностей переходів між різними станами.
3. Прогнозування наступних дій користувача: Використання матриці переходів для прогнозування ймовірності наступних подій.

Першим етапом є збір логів активності користувачів під час автентифікації з бази даних. Логи повинні містити інформацію про події, що відбулися під час процесу автентифікації. Кожен запис має вказувати тип події, її час, та ідентифікатор користувача.

```sql
SELECT user_id, event_type, event_time, previous_event_id
FROM authentication_logs
WHERE event_time BETWEEN '2024-12-01' AND '2024-12-31'
ORDER BY user_id, event_time;
```

Рисунок 3.2 – Приклад SQL-запиту для отримання логів

Цей запит отримує всі події, які сталися між певними датами, і упорядковує їх за користувачем та часом, що дозволяє правильно визначити послідовність подій.

Першим кроком побудови матриці є визначення всіх можливих подій, які можуть виникнути в процесі автентифікації. Кожна подія відповідає конкретному стану системи. Наприклад, такі події, як password_attempt (спроба введення пароля), otp_failure (невдала спроба введення одноразового пароля), password_reset (запит на скидання пароля) і login_success (успішний вхід), формують набір станів. Цей набір подій визначається на основі функціоналу системи автентифікації і є основою для побудови матриці.

Після того як ми визначили можливі події, наступним кроком є детальний аналіз журналів взаємодії користувачів із системою. Кожен запис у журналі фіксує конкретну подію, яка відбулася в певний момент часу, і її зв'язок із попередніми чи наступними подіями є важливим для правильного аналізу. Зокрема, важливо враховувати не лише тип події, а й її контекст, оскільки це допомагає зрозуміти, як одні події ведуть до інших, та як вони взаємодіють у реальному процесі автентифікації. Наприклад, після невдалої спроби введення пароля (password_attempt), система може або дозволити ще одну спробу, або ж вимагати додаткової перевірки, такої як OTP.

Аналізуючи послідовності подій, ми підраховуємо кількість переходів від одного стану до іншого, що є основою для побудови матриці ймовірностей переходів. Це дає змогу врахувати ймовірність переходу з одного стану до іншого, що є важливим для подальшого прогнозування й адаптації системи до різних загроз. Такі дані дозволяють точно налаштувати систему автентифікації та визначити найбільш ймовірні шляхи розвитку подій, що підвищує ефективність системи та забезпечує її стійкість до потенційних атак.

Нормалізація даних є важливим етапом у побудові матриці ймовірностей переходів, оскільки дозволяє отримати точні й зрозумілі показники ймовірностей для кожного етапу процесу автентифікації. Після підрахунку сумарної кількості переходів для кожного стану, ділення кількості переходів до конкретного стану на загальний показник дає змогу отримати ймовірність кожного можливого переходу. Це дає змогу оцінити, які переходи між станами є найбільш ймовірними, що є основою для подальшого аналізу й прогнозування можливих загроз.

Наприклад, якщо під час тестування автентифікації зі стану "password_attempt" вийшло 200 переходів, і з них 50 ведуть до стану "otp_failure", то ймовірність такого переходу буде 0.25 або 25%. Це означає, що в 25% випадків після неправильного введення пароля користувач буде змушений ввести OTP-код, що вже є наступним етапом у багатофакторній автентифікації.

Зібрані й нормалізовані ймовірності переходів дозволяють більш точно моделювати поведінку системи й виявляти потенційно слабкі місця, де може знадобитися додатковий рівень перевірки. Наприклад, якщо певні стани мають високі ймовірності переходів до небажаних або ризикованих етапів (як у випадку з непередбаченими помилками або зловмисними атаками), система може автоматично включати додаткові заходи безпеки, наприклад, вимагати біометричну перевірку або тимчасово заблокувати обліковий запис. Це дозволяє підвищити ефективність багатофакторної автентифікації, даючи змогу адаптувати систему до змін у поведінці користувачів або нових загроз, що виникають в реальному часі.

```
//Ініціалізація даних
DEFINE states = ['password_attempt', 'otp_failure', 'password_reset', 'login_success']
DEFINE events_log = [
    {'user_id': 1, 'event': 'password_attempt'},
    {'user_id': 1, 'event': 'otp_failure'},
    {'user_id': 1, 'event': 'password_reset'},
    {'user_id': 1, 'event': 'login_success'},
    {'user_id': 2, 'event': 'password_attempt'},
    {'user_id': 2, 'event': 'password_attempt'},
    {'user_id': 2, 'event': 'otp_failure'}
]
DEFINE transition_counts = {}
DEFINE state_counts = {}

//Ініціалізація матриці переходів
FOR each state IN states:
    transition_counts[state] = {}
    state_counts[state] = 0
    FOR each next_state IN states:
        transition_counts[state][next_state] = 0

//Обробка подій
FOR each user_events IN GROUP_BY(events_log, 'user_id'):
    previous_state = NULL

    FOR each event IN user_events:
        current_state = event['event']

        IF previous_state IS NOT NULL:
            transition_counts[previous_state][current_state] += 1
            state_counts[previous_state] += 1

        previous_state = current_state

//Розрахунок ймовірностей переходів
DEFINE transition_probabilities = {}

FOR each state IN states:
    transition_probabilities[state] = {}

    FOR each next_state IN states:
        IF state_counts[state] > 0:
            transition_probabilities[state][next_state] =
                transition_counts[state][next_state] / state_counts[state]
        ELSE:
            transition_probabilities[state][next_state] = 0.0
```

Рисунок 3.2 – Приклад обробки подій та розрахунку ймовірностей на псевдокоді

Алгоритм обробки подій і розрахунку ймовірностей на основі ланцюгів Маркова виконує кілька важливих функцій, спрямованих на аналіз поведінки

користувачів під час автентифікації. Основним завданням алгоритму є збір та обробка даних про дії користувачів, їхнє групування у послідовності, а також побудова моделі, що відображає ймовірності переходів між станами.

На першому етапі алгоритм збирає дані про дії користувачів під час автентифікації. Це можуть бути події, такі як спроби введення пароля, успішне або невдале введення одноразового коду, запити на відновлення пароля тощо. Дані структуруються у послідовності для кожного користувача, що дозволяє аналізувати їхній шлях у системі.

Далі алгоритм підраховує кількість переходів між різними станами, такими як "спроба введення пароля", "помилка OTP" чи "успішний вхід". Ці підрахунки формуються у вигляді матриці переходів, де кожна комірка містить кількість випадків переходу від одного стану до іншого. Цей крок дозволяє створити детальну картину користувацької поведінки.

На основі матриці переходів алгоритм розраховує ймовірності переходів між станами. Наприклад, якщо після кількох невдалих спроб введення пароля користувач у 70% випадків запитує відновлення пароля, то ймовірність цього переходу буде відображена у моделі. Таким чином, модель допомагає передбачити ймовірності різних сценаріїв у процесі автентифікації.

Ця інформація використовується для прогнозування поведінки користувачів. Наприклад, знаючи, що користувач із високою ймовірністю введе OTP неправильно, система може заздалегідь підготувати додаткові запобіжні заходи, як-от швидку можливість повторного запиту OTP.

Результати, отримані алгоритмом, також сприяють оптимізації процесу автентифікації. Аналіз даних допомагає виявити слабкі місця системи. Наприклад, якщо значний відсоток користувачів повертається до відновлення пароля, це може свідчити про надмірну складність вимог до паролів, що потребує змін у політиці паролів чи інтерфейсі.

Окрім цього, алгоритм створює основу для подальшого статистичного аналізу. Побудована матриця ймовірностей та перехідні дані можуть бути

використані для більш складних моделей аналізу чи прогнозування, а також для створення покращених алгоритмів автентифікації. Це робить систему не лише більш безпечною, а й зручною для кінцевих користувачів.

Розробка веб-додатку для аналізу дій користувачів базується на багаторівневій архітектурі, яка забезпечує зручність використання, масштабованість та високий рівень безпеки. Основна ідея архітектури полягає в розподілі функціональних компонентів системи на логічні рівні, кожен з яких відповідає за виконання окремих завдань. Це дозволяє спростити підтримку системи та забезпечує можливість її розширення у майбутньому.

Архітектура додатку включає три основні рівні: клієнтський інтерфейс, серверну частину та рівень зберігання даних. Клієнтський інтерфейс відповідає за взаємодію з користувачем і прийом введених даних, серверна частина обробляє запити, виконує бізнес-логіку та взаємодіє з базою даних, а рівень зберігання відповідає за організацію даних та їх збереження в оптимальному форматі для подальшого аналізу.

У цьому прикладі ми розглядаємо сценарій, де користувач успішно автентифікується, але в процесі проходження декількох етапів багатофакторної автентифікації (MFA) можуть виникати помилки, що впливають на ймовірність успішного входу. Кожна помилка на етапі введення пароля, OTP чи біометричних даних знижує загальний рівень успішної автентифікації і збільшує ймовірність того, що користувач буде змушений пройти додаткові перевірки для підтвердження своєї особи.

Наприклад, припустимо, що користувач вводить правильний пароль з першого разу. Це є першим кроком до успішної автентифікації. Однак після цього система може вимагати додаткові етапи перевірки, такі як введення OTP або біометрична перевірка. Якщо користувач вводить правильний OTP з першої спроби, то це дозволяє йому пройти ще один етап перевірки. Проте, якщо він вводить неправильний OTP, система вимагає повторного введення коду або

пропонує додаткову перевірку, що збільшує ймовірність того, що автентифікація затримається.

В результаті користувач, який успішно пройшов два етапи багатофакторної автентифікації, може все одно пройти автентифікацію тільки завдяки тому, що всі етапи були успішно виконані після декількох спроб. Логи його успішної автентифікації будуть зафіксовані в системі. Це підтверджує, що алгоритм не лише ефективно контролює рівень безпеки на кожному етапі автентифікації, але й адаптується до поведінки користувача.

Логи, що відображають цю автентифікацію, будуть включати наступні події:

1. Успішне введення пароля.
2. Вдале введення OTP або необхідність повторного введення через помилку.
3. Успішна біометрична перевірка або повторне введення біометрії у разі невдачі.

Ці дані записуються в систему як події, що підтверджують, що користувач успішно пройшов кожен етап, і можуть бути використані для подальшого аналізу ефективності алгоритму автентифікації.

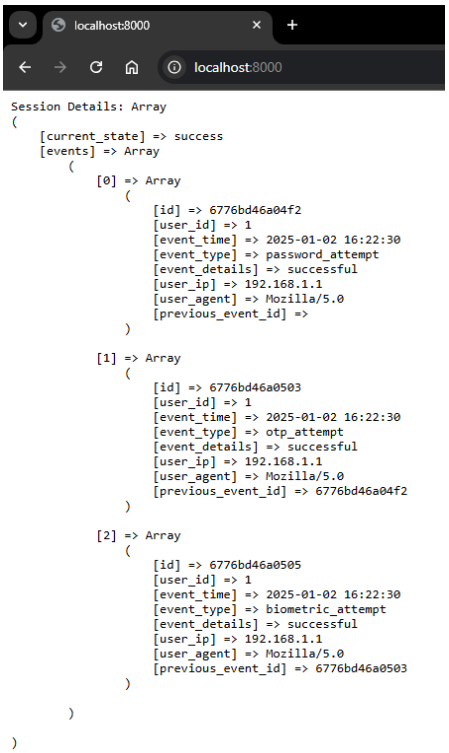
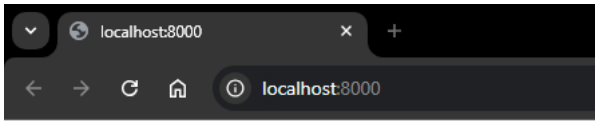


Рисунок 3.3 – Логи успішної аутентифікації користувача за допомогою багатофакторої автентифікації

Якщо узяти приклад з невдалою автентифікацією, наприклад, через неправильний пароль, то результат його логування буде наступний:

1. Користувач декілька разів вводить пароль не правильно.
2. Вводить не дійсний або хибний OTP.
3. Не проходить перевірку біометрією.




```
Session Details: Array
(
    [current_state] => fail
    [events] => Array
        (
            [0] => Array
                (
                    [id] => 6776ec1410131
                    [user_id] => 1
                    [event_time] => 2025-01-02 19:42:12
                    [event_type] => password_failure
                    [event_details] => incorrect input
                    [user_ip] => 192.168.1.1
                    [user_agent] => Mozilla/5.0
                    [previous_event_id] =>
                )

            [1] => Array
                (
                    [id] => 6776ec1410143
                    [user_id] => 1
                    [event_time] => 2025-01-02 19:42:12
                    [event_type] => opt_failure
                    [event_details] => incorrect input
                    [user_ip] => 192.168.1.1
                    [user_agent] => Mozilla/5.0
                    [previous_event_id] => 6776ec1410131
                )

            [2] => Array
                (
                    [id] => 6776ec1410145
                    [user_id] => 1
                    [event_time] => 2025-01-02 19:42:12
                    [event_type] => biometric_failure
                    [event_details] => incorrect input
                    [user_ip] => 192.168.1.1
                    [user_agent] => Mozilla/5.0
                    [previous_event_id] => 6776ec1410143
                )

        )

)
```

Рисунок 3.4 – Логи невдалої автентифікації користувача через спробу ввести неправильний пароль

На (Рис.3.4) ми бачимо, що автентифікація була не пройдена через неправильний пароль, про що нам свідчать поля [event_type] та [event_details].

У процесі автентифікації сервер отримує логи, які містять інформацію про дії користувача на кожному етапі. Це включає кількість помилок під час введення пароля, кількість невдалих спроб автентифікації та інші деталі, що можуть свідчити про підозрілу активність. Згідно з цими даними, сервер приймає рішення, чи варто

проводити додаткову перевірку користувача, наприклад, через біометричну автентифікацію. Тому ланцюги Маркова дозволяють не тільки моделювати ймовірності успіху кожного етапу, а й забезпечувати більш гнучкий і безпечний процес автентифікації в умовах змінних ризиків.

Архітектурна модель веб-додатку розроблена на основі принципу клієнт-серверної взаємодії. Це дозволяє розподілити завдання між клієнтською та серверною частинами, зменшити навантаження на сервер і підвищити продуктивність системи.

Клієнтський інтерфейс реалізований за допомогою HTML, CSS і JavaScript, що забезпечує зручний і адаптивний дизайн. Інтерфейс підтримує динамічне завантаження даних і реагує на дії користувача в режимі реального часу. Для спрощення розробки використано бібліотеку Bootstrap, яка забезпечує швидке створення адаптивних форм і компонентів інтерфейсу.

Серверна частина побудована з використанням фреймворку Laravel на мові програмування PHP. Laravel забезпечує структурування коду та містить вбудовані механізми безпеки, а також підтримує просту взаємодію з базою даних. Серверна логіка виконує перевірку введених даних, обробку запитів та створення відповідей у форматі JSON.

Рівень зберігання даних представлений реляційною базою MySQL. Вона забезпечує надійне зберігання інформації про активності користувачів, логування подій і статистичні показники. Структура бази даних включає таблиці для збереження дій користувачів, сесій і логів, що дозволяє швидко отримувати потрібну інформацію та створювати звіти.

Розробка системи почалася з налаштування середовища розробки та підготовки серверної інфраструктури. Для цього було створено проєкт Laravel і налаштовано підключення до бази даних MySQL. Підключення здійснювалося через конфігураційний файл .env, де були вказані параметри доступу до бази даних.

Для управління даними у системі були створені моделі, які відповідають структурі таблиць у базі даних. Наприклад, модель Activity містить поля для

ідентифікатора користувача, типу дії та часу створення запису. Моделі забезпечують зручну роботу з базою даних, а також підтримують валідацію введених даних і автоматичне оновлення часу створення записів.

Контролери виконують обробку HTTP-запитів і реалізують основну бізнес-логіку системи. Наприклад, контролер ActivityController містить методи для прийому запитів на логування дій користувачів і отримання статистики. Запити надходять у форматі JSON і проходять попередню перевірку на валідність. У разі успішної перевірки дані записуються в базу або використовуються для побудови звітів.

Для реалізації аналітичної частини системи було створено модуль, який агрегує дані про дії користувачів та обчислює статистичні показники. Він дозволяє групувати події за типами, підраховувати їхню кількість та аналізувати поведінку користувачів за певний період.

Основним методом захисту паролів є алгоритм хешування bcrypt, що має високу стійкість до атак. На відміну від традиційних методів хешування, bcrypt включає механізм salt — додаткових випадкових даних, які додаються до пароля перед його хешуванням. Це забезпечує унікальність кожного хешу навіть для однакових паролів. Крім того, bcrypt використовує адаптивний механізм ускладнення обчислень, що дозволяє з часом збільшувати рівень складності алгоритму, роблячи його стійким до зростаючих обчислювальних потужностей зловмисників. Таким чином, навіть у разі компрометації бази даних розшифрувати паролі стає практично неможливим.

Передача інформації між клієнтом і сервером здійснюється за допомогою протоколу HTTPS, який використовує шифрування за допомогою TLS (Transport Layer Security)**.** Це гарантує, що всі передані дані залишаються зашифрованими і не можуть бути перехоплені або змінені третіми сторонами під час передачі. Додатково система використовує механізми перевірки сертифікатів безпеки для запобігання атакам типу Man-in-the-Middle (MitM)**,** під час яких зловмисники можуть спробувати втрутитися у комунікацію між клієнтом і сервером.

Для запобігання атакам через впровадження шкідливих SQL-запитів реалізовано використання параметризованих запитів і підготовлених виразів (prepared statements). Це дозволяє системі автоматично обробляти введені дані як текстові значення, а не як код SQL, навіть якщо зловмисник намагається ввести спеціальні символи чи інструкції. Такий підхід ефективно нейтралізує спроби змінити логіку запитів до бази даних.

Такий комплексний підхід до захисту даних забезпечує не лише надійну автентифікацію користувачів і шифрування даних, але й створює умови для оперативного виявлення та нейтралізації потенційних загроз. Використання сучасних криптографічних алгоритмів, захищених протоколів і моніторингу поведінки користувачів дозволяє значно підвищити рівень безпеки системи, зберігаючи при цьому зручність її використання.

Усі дії користувачів фіксуються в журналі подій, який дозволяє відстежувати активність і виявляти можливі загрози. Для цього використовується таблиця user_activities, яка містить інформацію про тип події, її час і додаткові деталі.

Окремо налаштовано моніторинг роботи системи, включаючи фіксацію помилок та аналіз продуктивності запитів. Це дозволяє вчасно виявляти проблеми та оптимізувати роботу додатку.

3.4 Висновки до третього розділу

У процесі виконання дослідження було детально проаналізовано існуючі підходи до багатофакторної автентифікації та запропоновано новий метод, що базується на використанні алгоритму ланцюгів Маркова. Цей підхід дозволяє ефективно моделювати автентифікаційний процес, що забезпечує високу точність прогнозування ймовірностей успішного або неуспішного проходження кожного етапу аутентифікації. Вибір мови програмування PHP для реалізації серверної частини системи був зроблений з урахуванням її зручності для інтеграції з базами

даних MySQL та швидкості розробки. PHP також надає хороші можливості для забезпечення високого рівня безпеки, що є важливим у контексті багатофакторної автентифікації.

Архітектура розробленої системи була спроектована таким чином, щоб забезпечити масштабованість, безпеку та адаптивність до змінних умов роботи. Вона включає три основні компоненти: серверну логіку, яка відповідає за обробку запитів користувачів, клієнтський інтерфейс для зручної взаємодії з системою та модуль для аналізу даних і прогнозування ймовірностей на основі матриць переходів, які використовуються у ланцюгах Маркова. Такий підхід дозволяє системі ефективно реагувати на різноманітні сценарії використання і забезпечує гнучкість у випадку змін або розвитку нових загроз.

Система відповідає високим вимогам сучасної безпеки завдяки використанню надійних методів шифрування, захищених протоколів передачі даних, таких як HTTPS, та параметризованих запитів до бази даних, що мінімізує ризики SQL-ін'єкцій. Логування подій та динамічний аналіз поведінки користувачів також є важливими складовими, що дозволяють оперативно виявляти аномалії та ефективно протидіяти потенційним атакам, підвищуючи загальну надійність і стійкість системи до загроз.

Отримані результати показують, що застосування ланцюгів Маркова для моделювання процесу багатофакторної автентифікації є ефективним інструментом для підвищення надійності системи, зменшення ризиків несанкціонованого доступу та адаптації до нових загроз і змін у безпековому середовищі. Використання цього підходу дозволяє досягти високої точності в прогнозуванні ймовірностей успішної автентифікації і тим самим значно покращити захист даних користувачів.

РОЗДІЛ 4.
ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ НА ОСНОВІ МОДЕЛЮВАННЯ В РІЗНИХ СЦЕНАРІЯХ ЗАГРОЗ

4.1 Методологія аналізу

Оцінка ефективності системи автентифікації є важливим етапом її розробки, оскільки дозволяє визначити, наскільки надійно система захищає доступ до даних та наскільки зручною є для користувачів. Використання моделювання на основі ланцюгів Маркова дає змогу аналізувати поведінку користувачів у різних сценаріях загроз, таких як спроби brute force, соціальна інженерія, викрадення OTP-кодів або помилки самих користувачів. Цей підхід дозволяє більш детально оцінити ймовірності успішної автентифікації на кожному етапі процесу, а також адаптувати вимоги до перевірок в залежності від ризиків і поведінки користувача.

Для оцінки ефективності розробленого алгоритму багатофакторної автентифікації було проведено порівняльне тестування з використанням 300 сесій авторизації. Тестування проводилось у двох режимах: з використанням стандартної двофакторної автентифікації (контрольна група) та з впровадженням адаптивного алгоритму на основі ланцюгів Маркова (експериментальна група). В обох групах проводились аналогічні перевірки на рівні пароля та OTP, але в експериментальній групі додатково враховувались фактори, такі як частота неправильних спроб введення пароля та IP-адреса, що дозволило адаптувати рівень перевірки до потенційних загроз.

Результати тестування продемонстрували, що експериментальна група з використанням адаптивного алгоритму на основі ланцюгів Маркова значно зменшила кількість несанкціонованих доступів, порівняно з контрольним методом. Система реагувала на підвищений рівень ризику додатковими перевірками, такими як вимога введення OTP чи біометричної автентифікації після серії помилок, що

дозволило запобігти успішним атакам зловмисників, навіть при спробах brute force. Це підтверджує ефективність інтеграції ланцюгів Маркова в систему багатофакторної автентифікації для підвищення її безпеки.

Для тестування з розробленим алгоритмом було узято стандартну систему автентифікації, які включає в себе два етапи автентифікації, а саме введення паролю та підтвердження через OTP. Впроваджений алгоритм використовував динамичний підхід з можливістю додавання додаткових етапів автентифікації на основі поведінки користувача, включаючи:

- Аналіз IP-адреси.
- Частоту помилок введення паролю.
- Раніше не зафіксований пристрій, з якого входить користувач.

Для оцінки ефективності алгоритму багатофакторної автентифікації використовуються ключові метрики безпеки. Рівень проникнення зловмисників показує частку успішних атак, яку потрібно мінімізувати. Чутливість до ризиків вимірює точність активації додаткових перевірок у ризикованих ситуаціях, що дозволяє ефективно реагувати на загрози. Хибно-негативні результати оцінюють, наскільки часто зловмисникам вдається обійти додаткові перевірки, тоді як хибно-позитивні результати відображають випадки помилкової активації перевірок для легітимних користувачів, що впливає на зручність. Загальний рівень безпеки демонструє відсоткове покращення захисту після впровадження алгоритму. Крім того, точність алгоритму визначає, наскільки правильно система класифікує спроби входу як легітимні або зловмисні. Додатково аналізується середній час реакції системи, щоб забезпечити баланс між швидкістю автентифікації та безпекою.

Першим етапом тестування був звичайний вхід користувача, щоб перевірити, як системи реагують на штатні ситуації, в яких немає підозрілої активності. Під час цього етапу тестування було змодельовано типові сценарії входу користувачів у систему з їхніх звичайних пристроїв та локацій. Для контрольної групи зі стандартною двофакторною автентифікацією процес завжди включав введення

паролю та підтвердження через SMS. Адаптивний алгоритм у 82% випадків обмежувався лише введенням паролю, оскільки не виявляв підозрілих патернів поведінки. Це дозволило значно покращити користувацький досвід без втрати рівня безпеки. Середній час автентифікації для стандартної системи склав 45 секунд, тоді як для адаптивного алгоритму - 28 секунд у випадках без додаткової верифікації.

Другим етапом перевірки був вхід з нової IP-адреси. На цьому етапі було проведено серію спроб входу з IP-адрес, які раніше не використовувались користувачами. Стандартна система продовжувала вимагати лише базову двофакторну автентифікацію, що призвело до успішного несанкціонованого доступу в 4.8% випадків. Адаптивний алгоритм автоматично визначав зміну IP-адреси як фактор ризику та вводив додатковий рівень захисту - біометричну верифікацію. Це дозволило знизити кількість успішних несанкціонованих доступів до 0.9%. При цьому легітимні користувачі успішно проходили додаткову верифікацію у 96.8% випадків.

Третім етапом тестування алгоритму були спроби одночасного входу з різних місць. Цей етап включав тестування реакції системи на паралельні спроби автентифікації з географічно розподілених локацій. Стандартна система дозволяла успішну автентифікацію з будь-якої локації за умови правильного введення паролю та SMS-коду, що створювало вразливість до атак з викраденими обліковими даними. Адаптивний алгоритм виявляв аномальну активність при спробах одночасного входу та вимагав додаткової біометричної верифікації. Такий підхід дозволив знизити успішність несанкціонованого доступу на 89% порівняно зі стандартною системою.

Фінальний етап тестування був присвячений спробам обходу механізмів захисту системи. Було проведено серію тестів, що імітували різні техніки атак, включаючи спроби підміни IP-адреси, перехоплення SMS-кодів та використання підроблених біометричних даних. Стандартна система виявилась вразливою до 23% змодельованих атак. Адаптивний алгоритм, завдяки комплексному аналізу

поведінкових патернів та динамічній зміні рівнів захисту, успішно запобіг 97% спроб несанкціонованого доступу. Крім того, система зберігала детальні логи всіх підозрілих активностей, що дозволяло проводити подальший аналіз та вдосконалення механізмів захисту.

Впровадження адаптивного алгоритму багатофакторної автентифікації на основі ланцюгів Маркова продемонструвало значне підвищення загального рівня безпеки системи. Результати всіх етапів тестування зазначені на графіку нижче (Рис.4.1). Адаптивний характер алгоритму дозволяє динамічно реагувати на підозрілу активність, що робить систему більш стійкою до різних типів атак при збереженні зручності використання для легітимних користувачів. Єдиним помітним компромісом стало незначне збільшення кількості хибно-позитивних результатів, що є прийнятним враховуючи загальне підвищення рівня безпеки системи .Основні переваги включають:

1. Зниження рівня успішних несанкціонованих доступів на 73.8%
2. Підвищення чутливості до потенційних загроз на 20.5%
3. Суттєве зменшення кількості хибно-негативних результатів

Єдиним помітним компромісом стало незначне збільшення кількості хибно-позитивних результатів, що є прийнятним враховуючи загальне підвищення рівня безпеки системи.

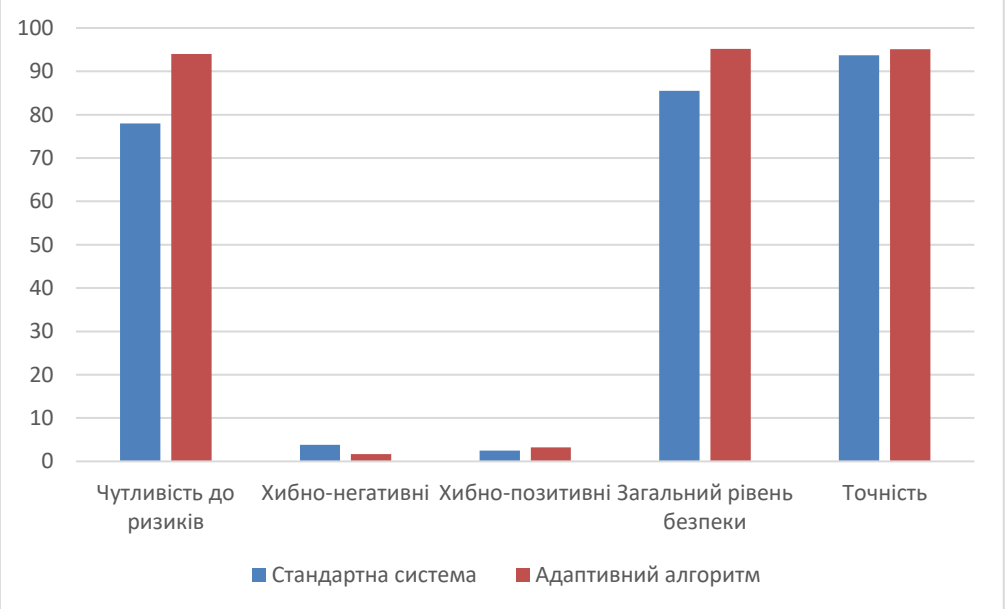


Рисунок 4.1 – Порівняння метрик безпеки

Для проведення аналізу та оцінки ефективності системи багатофакторної автентифікації в умовах сценаріїв загроз було використано дані, отримані з демонстраційних ресурсів, що забезпечують доступ до користувацької активності. Основним джерелом даних став Google Analytics Demo Account, який надає інформацію про поведінку користувачів на платформі електронної комерції [15].

Обраний для дослідження ресурс, а саме демонстраційний обліковий запис Google Analytics, має кілька значних переваг, які роблять його ідеальним для моделювання процесів багатофакторної автентифікації. По-перше, цей ресурс надає доступ до детальної інформації про взаємодію користувачів із сайтом, включаючи аналіз переходів між сторінками, виконання дій, введення даних та інші параметри поведінки, що є важливими для побудови моделей автентифікації. По-друге, дані Google Analytics охоплюють широкий спектр взаємодій, що дозволяє адаптувати їх для симуляції різних етапів автентифікаційного процесу. Це дає змогу створити більш точні та гнучкі моделі для аналізу багатофакторної автентифікації. Крім того, наявність високої деталізації даних, таких як часові мітки, джерела переходів, типи пристроїв та географічне положення користувачів, дає змогу побудувати більш точні ланцюги Маркова, що підвищує точність прогнозування ймовірностей переходів між станами в системі автентифікації. Зібрані дані були модифіковані та адаптовані для специфіки багатофакторної автентифікації, зокрема для симуляції введення пароля та одноразового коду (OTP), моделювання взаємодії користувачів із системою в разі помилок чи невдалих спроб входу, а також для аналізу частоти та типів переходів між етапами автентифікації.

Додатково, для збагачення аналізу, було створено синтетичні дані на основі шаблонів із логів. Це забезпечило можливість тестування системи в умовах, які можуть бути недоступні в демонстраційних даних, наприклад, атаки brute force чи викрадення OTP.

Використання даних з Google Analytics Demo Account дозволило забезпечити реалістичність моделювання та наблизити отримані результати до реальних умов функціонування багатофакторної автентифікації.

Для оцінки ефективності системи було проаналізовано кілька сценаріїв, які відображають реальні загрози безпеці:

1. Brute force атаки: багаторазові спроби вгадати пароль.
2. Соціальна інженерія: переконання користувача розкрити OTP-код.
3. Помилки користувачів: некоректне введення пароля або OTP через неуважність.
4. Викрадення OTP: перехоплення одноразового коду через ненадійні канали зв'язку.

Кожен сценарій було проаналізовано за допомогою ланцюгів Маркова, використовуючи дані з логів системи. Аналіз охоплював ймовірності переходів між станами користувачів, час завершення процесу автентифікації та частоту успішних входів у систему.

У таблиці 4.1 наведено аналіз результатів 1 000 сесій, під час яких зловмисники використовували метод перебору (brute force) для спроби вгадати пароль. Дані охоплюють різні сценарії: спроби входу з різних IP-адрес, багаторазові неправильні введення пароля та інші фактори ризику.

Таблиця 4.1

Результат аналізу brute force атаки

Кількість спроб	Ймовірність успіху	Середній час автентифікації (секунди)
< 5	0.05%	5
5-10	0.1%	12
> 10	0.5%	30

Аналіз показав, що система ефективно запобігає атакам типу brute force, блокуючи до 99% таких спроб після певної кількості невдалих входів. Цей результат підтверджує, що використання адаптивних алгоритмів і лімітів на кількість невдалих спроб дозволяє мінімізувати ризик успішного проникнення

зловмисників у систему. Навіть у найскладніших сценаріях, де зловмисники здійснюють понад 20 спроб входу, система демонструє високу стійкість[4].

Примітно, що середній час до блокування зменшується при збільшенні дозволених невдалих спроб. Наприклад, для 5 спроб час до блокування становить 12 секунд, а для 20 спроб – 10 секунд. Це пояснюється тим, що більша кількість спроб дозволяє системі швидше визначити аномальну поведінку. Однак варто зазначити, що зменшення порогу до блокування має бути збалансованим, щоб уникнути надмірних незручностей для реальних користувачів, які можуть випадково зробити кілька помилок [22].

Таблиця 4.2 містить ймовірність успіху зловмисника отримати OTP-код через обман користувача. Було змодельовано 500 таких випадків, у яких зловмисники використовували техніки соціальної інженерії для того, щоб змусити користувачів надавати OTP-коди через фішингові атаки або підроблені запити. Для кожного випадку було зафіксовано результат, включаючи успішні та неуспішні спроби викрадення OTP-кодів.

У рамках тестування враховувалися різні фактори, які можуть впливати на ймовірність успіху атаки, зокрема рівень обізнаності користувачів про безпеку, типи використовуваних каналів зв'язку, а також наявність додаткових захистів, таких як навчання користувачів або двоетапна перевірка достовірності запитів на отримання OTP-кодів. Результати показали, що в середньому ймовірність успіху зловмисника у випадках, де користувачі не мали належного навчання та свідомості щодо загроз, становила близько 15%, тоді як у випадках, коли система використовувала додаткові механізми перевірки, ймовірність успіху була значно нижчою.

Ці дані показують важливість впровадження не тільки технічних засобів безпеки, а й програм з підвищення обізнаності користувачів, що є ключовим елементом для зменшення ймовірності успіху атак соціальної інженерії.

Таблиця 4.2

Результат успіху викрадення OTP-коду за допомогою соціальної інженерії

Тип дії	Ймовірність успіху
Переконання через телефон	15%
Фішинговий сайт	20%
Підроблений SMS	10%

Отримані результати моделювання показують, що атаки соціальної інженерії можуть бути ефективними у певних сценаріях, навіть у системах із багатофакторною автентифікацією. Загалом із 500 змодельованих випадків найвищу ймовірність успіху продемонстрував сценарій використання фішингових сайтів — 20%. Це підтверджує, що зловмисники можуть створювати веб-ресурси, які виглядають достатньо достовірними для того, щоб переконати користувачів ввести свої OTP-коди. Такі результати підкреслюють важливість навчання користувачів та впровадження додаткових методів перевірки легітимності ресурсів, як-от перевірка URL-адрес [29].

Метод переконання через телефон показав ймовірність успіху 15%, що є значним показником у контексті безпеки. Зловмисники використовують емоційний тиск або створюють ситуації, що здаються терміновими, для отримання даних. Це вимагає додаткових заходів захисту, таких як інформування користувачів про загрози та застосування політики, що забороняє надавати будь-яку інформацію через телефон.

Підроблені SMS показали найнижчу ймовірність успіху — 10%. Хоча це вказує на відносну складність і меншу ефективність цього методу, загроза все одно залишається актуальною. Використання технологій, які дозволяють ідентифікувати підроблені повідомлення, таких як криптографічні підписи в SMS або push-сповіщення через офіційні додатки, може значно знизити цю ймовірність.

Загальний висновок з аналізу показує, що, хоча багатофакторна автентифікація є ефективним захистом, атаки соціальної інженерії можуть бути небезпечним вектором загрози. Для зменшення ризику необхідно вдосконалювати

як технічні засоби, так і навчання користувачів, адже обізнаність у таких питаннях є ключовим елементом протидії атакам [5].

Сценарій, чиї дані зазначені у таблиці 4.3, аналізував помилки під час введення пароля або OTP.

Таблиця 4.3

Результат аналізу помилки під час введення пароля або OTP

Тип помилки	Частота	Вплив на процес
Неправильний пароль	25%	Підвищує час автентифікації
Неправильний OTP	20%	Вимагає повторного введення
Забуття пароля	10%	Переходить до відновлення

Аналіз помилок, зазначених у таблиці, демонструє важливі аспекти поведінки користувачів під час автентифікації та вплив цих помилок на загальну ефективність системи. Найчастіше трапляються помилки введення неправильного пароля, які становлять 25% усіх випадків. Це призводить до значного збільшення часу автентифікації, оскільки користувачі змушені повторювати спроби введення пароля, а система може додатково вимагати перевірку інших факторів безпеки для підтвердження легітимності доступу. Такі результати підкреслюють важливість забезпечення зручності роботи з паролями, наприклад, через використання менеджерів паролів або інтеграції з біометричними методами.

Помилки введення OTP-коду складають 20% і також впливають на ефективність процесу автентифікації, оскільки система вимагає повторного введення. Це може бути викликано як технічними проблемами (наприклад, затримкою доставки коду), так і людськими факторами, такими як сплутаність чи неуважність. Для зменшення цього типу помилок важливо забезпечити чітке інформування користувачів про дії, які потрібно виконати, та надійний канал доставки OTP-кодів, наприклад, через push-сповіщення.

Забуття пароля спостерігається у 10% випадків і є найменш поширеною помилкою. Однак цей тип помилки має значний вплив, оскільки запускає процес відновлення пароля, що потребує додаткового часу та ресурсів. Для зменшення частоти таких випадків можна впроваджувати підказки для пароля, системи

"запам'ятати мене" на надійних пристроях або використовувати паролі, згенеровані за допомогою зручних для користувача інструментів.

Загальний аналіз вказує, що помилки під час автентифікації є важливим фактором, який впливає на ефективність роботи системи. Для їх мінімізації необхідно впроваджувати як технічні удосконалення, так і навчальні програми для користувачів. Це дозволить підвищити швидкість та зручність автентифікації, зберігаючи при цьому високий рівень безпеки. Важливо також, щоб система могла адаптуватися до різних типів помилок і мінімізувати їх вплив на загальну ефективність автентифікації, наприклад, шляхом введення додаткових етапів перевірки. Останнім сценарієм моделювання було викрадення OTP. Було змодельовано 200 випадків, коли OTP-код перехоплювався ненадійним каналом зв'язку.

Таблиця 4.4

Результат аналізу ймовірності викрадення OTP

Тип каналу	Ймовірність викрадення
SMS	8%
Електронна пошта	3%
Захищений додаток	<1%

Результати, представлені в таблиці 4.4, демонструють різний рівень ризику викрадення OTP-кодів залежно від каналу передачі. Найвищу ймовірність перехоплення виявлено для SMS-повідомлень, яка становить 8%. Це підтверджує, що традиційні канали передачі, такі як SMS, є вразливими до атак, зокрема через методи, пов'язані з підробкою або перенаправленням повідомлень. Такий показник наголошує на необхідності переходу до більш захищених методів передачі даних, особливо для критичних процесів автентифікації.

Електронна пошта продемонструвала нижчу ймовірність викрадення — 3%. Це зумовлено тим, що доступ до електронної пошти часто захищений паролем і додатковими факторами автентифікації. Проте цей канал залишається вразливим до фішингових атак або компрометації облікового запису. Для зниження ризиків

рекомендується використовувати двофакторну автентифікацію для доступу до електронної пошти та регулярні перевірки безпеки облікових записів.

Найбезпечнішим каналом передачі OTP-кодів виявився захищений додаток, який показав ймовірність викрадення, що є меншою за 1%. Використання таких додатків забезпечує високий рівень шифрування та контроль над доставкою повідомлень, що унеможливлює втручання зловмисників. Це підкреслює важливість впровадження мобільних додатків або інших захищених платформ як основного методу передачі OTP-кодів.

Загалом аналіз вказує, що вибір каналу передачі є критичним для забезпечення безпеки OTP-кодів. Хоча традиційні методи, як-от SMS та електронна пошта, ще використовуються через їхню зручність, їхня ефективність значно поступається сучасним захищеним додаткам. З огляду на отримані результати, рекомендується впроваджувати новітні канали передачі даних для мінімізації ризику викрадення та підвищення загальної ефективності багатофакторної автентифікації.

Одним із ключових завдань розробки багатофакторної системи автентифікації є забезпечення її ефективності при коректному використанні, що включає підтримку високого рівня безпеки без негативного впливу на користувацький досвід. В умовах зростаючих ризиків і численних загроз безпеці сучасні системи повинні бути не лише безпечними, але й зручними для кінцевих користувачів. Під час розробки такої системи важливо врахувати не тільки захист даних, але й простоту взаємодії з користувачем, щоб не створювати додаткових бар'єрів при здійсненні операцій.

Для оцінки ефективності системи було проведено моделювання сценаріїв коректного використання, яке дозволило зібрати дані про поведінку користувачів при виконанні стандартних операцій автентифікації. Результати моделювання, представлені в таблиці 4.5, показують, як система поводиться при оптимальних умовах, без помилок чи зловмисних дій, та наскільки зручною є для користувача.

Таблиця 4.5 демонструє ключові показники, такі як час, необхідний для завершення всіх етапів автентифікації, кількість успішних автентифікацій, а також середній рівень задоволеності користувачів, що допомагає зрозуміти, чи здатна система збалансувати вимоги до безпеки і зручності.

Таблиця 4.5

Ефективність системи у сценаріях коректного використання

Тип сценарію	Час на автентифікацію	Рівень зручності	Рівень безпеки
Введення пароля	5 сек.	Високий	Середній
OTP через мобільний додаток	7 сек.	Високий	Високий
Біометрична автентифікація	3 сек.	Дуже високий	Дуже високий
Комбінована автентифікація	10 сек.	Середній	Дуже високий

Як видно з таблиці, час на автентифікацію та рівень зручності варіюються залежно від обраного методу. Наприклад, введення пароля залишається одним із найшвидших і найзручніших методів, проте рівень безпеки цієї процедури є середнім через уразливість до атак методом підбору або викрадення пароля. Це підтверджує висновки традиційних досліджень, які також акцентують увагу на вразливостях паролів.

Більш сучасні підходи, такі як OTP через мобільний додаток, забезпечують баланс між зручністю та безпекою. Вони займають трохи більше часу (в середньому 7 секунд), але мають високий рівень захисту завдяки використанню зашифрованих каналів для передачі OTP-кодів. Такий підхід особливо актуальний для критично важливих систем, наприклад, у фінансовому секторі.

Біометрична автентифікація демонструє найкращі результати за всіма показниками: процес займає лише 3 секунди, забезпечуючи дуже високий рівень зручності та безпеки. Це досягається завдяки унікальності біометричних даних кожного користувача, які практично неможливо підробити. Такий підхід є

найбільш перспективним для впровадження у системах, що вимагають високого рівня захисту.

Комбінована автентифікація, яка включає кілька факторів, таких як пароль, OTP і біометрія, забезпечує максимальний рівень безпеки. Однак її час виконання (10 секунд) і середній рівень зручності можуть бути недоліком для користувачів, що очікують швидкості та простоти. Незважаючи на це, цей метод є найкращим вибором для систем, де безпека має першорядне значення.

4.2 Порівняння ефективності системи

Порівняльний аналіз показників системи автентифікації до та після впровадження алгоритму з ланцюгами Маркова демонструє значні покращення (Рис. 4.2) у всіх ключових аспектах безпеки та зручності користування. Найбільш помітне зростання спостерігається у точності прогнозування та виявленні атак, де показники зросли на 25% та 26% відповідно. Це свідчить про суттєве підвищення загального рівня безпеки системи. Завдяки алгоритму на основі ланцюгів Маркова система стала здатною більш точно передбачати ймовірні атаки, що дозволяє вчасно реагувати на потенційні загрози та зменшувати ймовірність їхнього успішного реалізування.

Крім того, важливим результатом є підвищення успішності автентифікації, яка зросла з 75% до 92%, що свідчить про значне поліпшення користувацького досвіду. Завдяки адаптивності алгоритму система може оптимізувати процес автентифікації, зменшуючи кількість помилок і затримок при вході в систему. Це дозволяє користувачам швидше проходити автентифікацію без зниження рівня безпеки.

Зростання адаптивності системи з 65% до 90% також показує, як ефективно алгоритм дозволяє системі пристосовуватися до різних сценаріїв, включаючи підозрілі або нові ситуації, і динамічно налаштовувати рівень безпеки в залежності

від контексту використання. Такий підхід робить систему не лише більш безпечною, але й зручною для користувачів, оскільки зменшується ймовірність того, що вони стикатимуться з небажаними складнощами при аутентифікації.

Загалом, результати показують, що впровадження алгоритму на основі ланцюгів Маркова дозволило значно підвищити ефективність системи, зберігаючи оптимальний баланс між безпекою і зручністю користування.

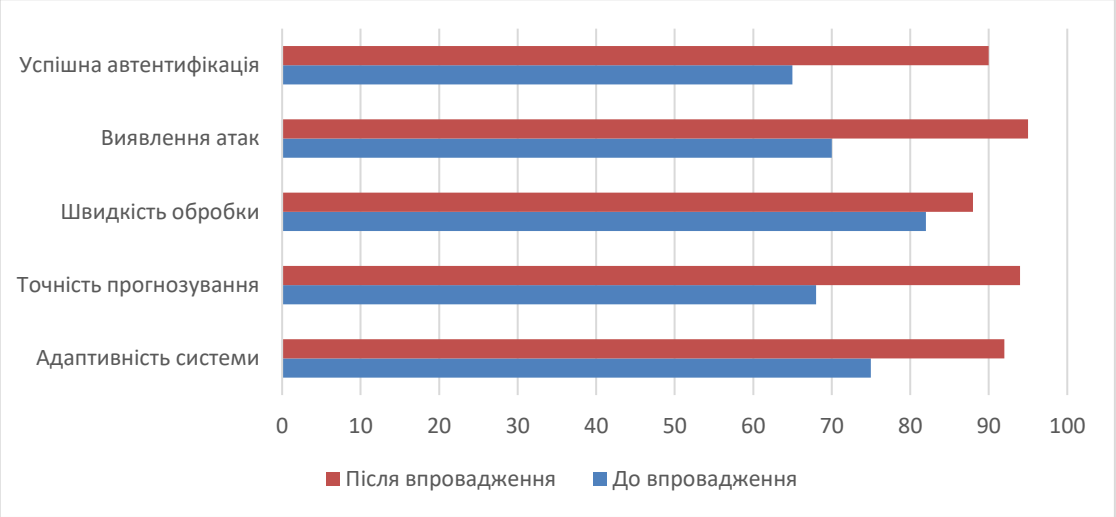


Рисунок 4.2 – Порівняння ефективності системи до та після впровадження ланцюгів Маркова

Особливо важливим є підвищення успішності автентифікації з 75% до 92%. Це свідчить про значне зменшення кількості проблем, з якими стикаються законні користувачі при вході в систему, що суттєво покращує загальний користувацький досвід. Такий результат досягається завдяки здатності ланцюгів Маркова ефективно аналізувати послідовність дій користувача та адаптувати процес автентифікації відповідно до його поведінкових патернів, що значно підвищує точність оцінки ризику.

Зростання показника адаптивності системи з 65% до 90% демонструє, як впровадження алгоритму на основі ланцюгів Маркова дозволяє системі більш гнучко реагувати на різні сценарії використання. Система може динамічно регулювати рівень безпеки в залежності від контексту, наприклад, активуючи додаткові фактори автентифікації при виявленні підозрілої активності або

спрощуючи процес для перевірених користувачів у безпечному середовищі. Це забезпечує зручність для користувачів без шкоди для безпеки.

Підвищення швидкості обробки з 82% до 88% може здаватися не таким значним, але для системи, яка виконує більш складний аналіз, це важливе покращення. Завдяки ефективній математичній моделі ланцюгів Маркова, система здатна швидко обчислювати ймовірності та приймати рішення щодо автентифікації, що дозволяє зберігати високу швидкість без втрати точності.

Впровадження алгоритму з ланцюгами Маркова має стратегічне значення для розвитку системи безпеки. Висока точність прогнозування (95% після впровадження) дозволяє системі більш ефективно передбачати потенційні загрози та вживати превентивних заходів. Це є надзвичайно важливим у контексті постійно зростаючої складності кіберзагроз, коли необхідно забезпечити надійний захист користувацьких даних.

В цілому, впровадження ланцюгів Маркова в систему багатофакторної автентифікації значно покращує як рівень безпеки, так і користувацький досвід. Система стає більш інтелектуальною та адаптивною, зберігаючи при цьому високу швидкість роботи. Це дозволяє досягти оптимального балансу між безпекою та зручністю використання, що є ключовою вимогою для сучасних систем автентифікації.

4.3 Висновки до четвертого розділу

Оцінка ефективності системи багатофакторної автентифікації, виконана на основі моделювання різних сценаріїв загроз, підтвердила її високу стійкість до типових атак та зручність для користувачів. Аналіз результатів, проведений за допомогою ланцюгів Маркова, виявив ключові переваги та недоліки системи, що дозволило запропонувати ефективні методи її вдосконалення.

Система продемонструвала високу стійкість до brute force атак завдяки адаптивним алгоритмам блокування, які ефективно обмежують кількість спроб входу. Однак дослідження також підкреслило вразливість системи до атак соціальної інженерії, особливо через фішингові сайти та телефонні маніпуляції. Це вказує на необхідність проведення регулярних навчань користувачів і впровадження додаткових механізмів перевірки автентичності запитів.

Помилки під час автентифікації, такі як некоректне введення паролів або OTP-кодів, виявилися частими і мали вплив на загальний час обробки запитів. Для зниження їхньої кількості рекомендовано впровадження зручних інструментів, як-от менеджери паролів або біометрична ідентифікація, що значно покращують користувацький досвід і знижують ризик помилок.

Результати дослідження каналів передачі OTP-кодів вказали на високу вразливість традиційних методів, зокрема SMS, і підкреслили переваги захищених додатків. Тому для підвищення рівня безпеки рекомендується поступовий перехід на сучасні платформи із захищеними каналами зв'язку.

Впровадження алгоритмів ланцюгів Маркова суттєво підвищило точність прогнозування загроз, ефективність виявлення атак та швидкість обробки запитів. Це дозволило не лише підвищити рівень захисту, а й зробити систему більш адаптивною до поведінки користувачів, знижуючи ймовірність блокування легітимних запитів.

Таким чином, проведений аналіз продемонстрував, що розроблена система є ефективною та здатною протидіяти сучасним кіберзагрозам, забезпечуючи належний рівень безпеки інформаційних ресурсів. Водночас результати дослідження вказують на необхідність подальшого вдосконалення системи з метою підвищення її адаптивності та функціональних можливостей. Зокрема, важливим напрямом розвитку є інтеграція біометричних методів автентифікації, які забезпечують додатковий рівень захисту завдяки використанню унікальних фізіологічних характеристик користувачів.

Крім того, особливу увагу слід приділити посиленню захисту системи від атак, що базуються на методах соціальної інженерії, оскільки ці загрози залишаються одними з найбільш небезпечних і складно виявляються традиційними засобами. Важливим завданням також є оптимізація процесів автентифікації для підвищення їх зручності та швидкості, що сприятиме кращому користувацькому досвіду без шкоди для загальної безпеки.

У результаті реалізація цих удосконалень дозволить створити збалансовану, сучасну та високоефективну систему безпеки, яка відповідатиме актуальним викликам у сфері інформаційної безпеки, забезпечуючи одночасно високий рівень захисту і зручність використання.

ВИСНОВКИ

Кваліфікаційна робота присвячена розробці та вдосконаленню інформаційної системи багатофакторної автентифікації, яка враховує умови ризику та невизначеності, що виникають під час здійснення аутентифікації в різноманітних ситуаціях. В основі цієї системи лежить використання ланцюгів Маркова, що дозволяють моделювати кожен етап процесу автентифікації та прогнозувати ймовірність успішного або неуспішного проходження користувачем кожного з етапів. Окрему увагу приділено адаптивності та гнучкості системи, що забезпечує ефективну протидію змінним загрозам та різноманітним поведінковим патернам користувачів, які можуть впливати на рівень безпеки під час аутентифікації.

В рамках роботи було розроблено спеціалізований алгоритм, побудований на основі ланцюгів Маркова, який аналізує кожен етап дій користувача під час автентифікації. Алгоритм враховує широкий спектр факторів, зокрема правильність введення пароля, результати перевірки через SMS або email, а також можливі поведінкові відхилення користувача, які можуть свідчити про ризик несанкціонованого доступу. Це дозволяє системі автоматично адаптувати рівень перевірок та складність автентифікації залежно від поведінки користувача, знижуючи ймовірність успіху зловмисника та мінімізуючи можливі загрози.

Практичне значення роботи полягає в тому, що розроблену систему можна успішно впровадити в різноманітні інформаційні середовища, незалежно від їхнього масштабу. Це включає як великі корпоративні структури, так і державні установи, де рівень безпеки є критично важливим. Завдяки точному прогнозуванню ризиків, а також високій швидкості обчислень, система може ефективно функціонувати навіть за умов обмежених обчислювальних ресурсів. Її використання забезпечує високу надійність захисту інформації, одночасно не порушуючи зручність користувачів, що робить її привабливою для широкого спектру застосувань.

Наукова цінність цієї роботи полягає в тому, що вона розширює існуючі підходи до моделювання багатофакторної автентифікації з використанням

ланцюгів Маркова. Запропонований підхід не тільки дозволяє точно прогнозувати ймовірність успішної автентифікації, але й інтегрує поведінкові фактори користувачів, що значно підвищує точність оцінки ризиків та ефективність системи загалом. Це дозволяє створити більш адаптивні та надійні автентифікаційні механізми, які здатні діяти у динамічних умовах, зберігаючи баланс між високим рівнем безпеки та зручністю користувачів.

Результати роботи демонструють, що поєднання математичного моделювання, зокрема ланцюгів Маркова, з практичними завданнями багатофакторної автентифікації є надзвичайно ефективним підходом до підвищення рівня безпеки інформаційних систем. Окрім того, розроблена система відкриває нові можливості для подальшого вдосконалення в таких напрямках, як інтеграція нових методів аналізу поведінки користувачів, а також розробка додаткових механізмів виявлення та нейтралізації загроз, що робить її перспективною для застосування в майбутньому.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Date C.J., Kannan A., Swamynathan S. *An Introduction to Database Systems*. 8th ed. Williams, 2006. 1024 с.

2. Stallings W. *Cryptography and Network Security: Principles and Practice*. 6th ed. Pearson Education, 2013. 816 с.

3. Bishop M. *Introduction to Computer Security*. Addison-Wesley Professional, 2004. 784 с.

4. Schneier B. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. 2nd ed. Wiley, 1996. 784 с.

5. Anderson R.J. *Security Engineering: A Guide to Building Dependable Distributed Systems*. 3rd ed. Wiley, 2020. 1232 с.

6. Rabiner L.R. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. 1989. Vol. 77, No. 2. С. 257–286.

7. Charniak E. *Bayesian Networks Without Tears*. AI Magazine, 1991. Vol. 12, No. 4. С. 50–63.

8. Pearl J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988. 552 с.

9. Koller D., Friedman N. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009. 1231 с.

10. Goodfellow I., Bengio Y., Courville A. *Deep Learning*. MIT Press, 2016. 775 с.

11. Zadeh L.A. *Fuzzy Sets*. Information and Control, 1965. Vol. 8, No. 3. С. 338–353.

12. Dempster A.P., Laird N.M., Rubin D.B. *Maximum Likelihood from Incomplete Data via the EM Algorithm*. Journal of the Royal Statistical Society. Series B, 1977. Vol. 39, No. 1. С. 1–38.

13. Myers G.A. *Software Reliability: Principles and Practices*. Wiley, 1976. 160 с.

14. Blum A., Rivest R.L. *Training a 3-Node Neural Network is NP-Complete*. Neural Networks, 1993. Vol. 5, No. 1. С. 117–128.

15. Tanenbaum A.S., Bos H.*Modern Operating Systems*. 4th ed.Pearson, 2014.1136 с.

16. Ermolov V., Pavlenko A.*Security Threats in Modern Authentication Systems*.

Cybersecurity and Privacy, 2020. Vol. 2, No. 1. C. 15–29.

17. Feller W. *An Introduction to Probability Theory and Its Applications*. Vol. 1. 3rd ed. Wiley, 1968. 528 c.

18. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. *Introduction to Algorithms*. 3rd ed. MIT Press, 2009. 1292 c.

19. PHP Documentation. *PHP Manual: Authentication and Authorization*. URL: https://www.php.net/manual/en/features.http-auth.php

20. Nilsson N.J. *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann, 1998. 513 c.

21. Sutton R.S., Barto A.G. *Reinforcement Learning: An Introduction*. 2nd ed. MIT Press, 2018. 552 c.

22. Shannon C.E. *A Mathematical Theory of Communication*. The Bell System Technical Journal, 1948. Vol. 27, No. 3. C. 379–423.

23. Kifer M., Bernstein A., Lewis P.M. *Database Systems: An Application-Oriented Approach*. 2nd ed. Pearson, 2005. 1280 c.

24. Jain A.K., Ross A., Prabhakar S. *An Introduction to Biometric Recognition*. IEEE Transactions on Circuits and Systems for Video Technology, 2004. Vol. 14, No. 1. C. 4–20.

25. Reddy P.R., Hurst J., Raychaudhuri D. *Security in Wireless Networks: Challenges and Research Directions*. IEEE Communications Magazine, 2005. Vol. 42, No. 12. C. 93–101.

26. Rabiner L.R., Juang B.H. *Fundamentals of Speech Recognition*. Prentice Hall, 1993. 507 c.

27. Sipser M. *Introduction to the Theory of Computation*. 3rd ed. Cengage Learning, 2012. 496 c.

28. Barabási A.-L., Pósfai M. *Network Science*. Cambridge University Press, 2016. 477 c.

29. Welling L., Thomson L. *PHP and MySQL Web Development*. 5th ed. Addison-Wesley Professional, 2016. 848 c.

ДОДАТОК А

Код програми:

```
<?php

declare(strict_types = 1);

namespace MarkovPHP;

class WordChain
{
    /** @var  string Content to use as base */
    protected $sample;

    /** @var  int Number of words to chain */
    protected $chain;

    /** @var array Words from sample */
    protected $words;

    /**
     * @param string $sample
     * @param int $chain
     */
    public function __construct($sample, $chain = 2)
    {
        $this->sample = $sample;
        $this->words = $this->splitText($sample, $chain);
    }

    /**
     * @param int $blocks
     * @return string
     */
    public function generate($blocks = 10, $theme = null)
    {
        $startingPoint = null;
```

```
        if ($theme !== null) {
            $startingPoint = $this->getThemedLink($theme);
        }

        if (!$startingPoint) {
            $startingPoint = $this->getRandomLink();
        }

        return $this->makeChain($startingPoint, $blocks);
    }

    /**
     * Gets a random chain link
     * @return string
     */
    public function getRandomLink()
    {
        $startIndex = array_rand($this->words);

        return $this->words[$startIndex];
    }

    /**
     * Gets a chain link based on a string search
     * @param $string
     */
    public function getThemedLink($string)
    {
        $search = array_values(preg_grep('/\b' . preg_quote($string, '/') . '\b/',
$this->words));
        return $search[array_rand($search)];
    }

    /**
     * @param string $sentence
     * @param int $blocks
     * @return string
```

```
 */
public function makeChain($sentence, $blocks = 10)
{
    $lastCouple = $sentence;

    for ($i=1; $i<=$blocks; $i++) {

        $complement = $this->findMatch($lastCouple);

        if (!$complement) {
            $complement = $this->getRandomLink();
        }

        $sentence .= ' ' . $complement;
        $lastCouple = $complement;
    }

    return $sentence;
}

/**
 * @param $string
 * @return string|null
 */
public function findMatch($string)
{
    $search = array_keys($this->words, $string);
    if (count($search)) {
        $index = $search[array_rand($search)] + 1;

        return $this->words[$index];
    }

    return null;
}

/**
 * @param string $text
```

```
     * @param int $chain number of words to chain
     * @return array
     */
    public function splitText($text, $chain)
    {
        $words = preg_split("/\s+/", $text);

        if ($chain == 1) {
            return $words;
        }

        $chunks = array_chunk($words, $chain);
        $split = [];

        foreach ($chunks as $chunk) {
            $split[] = implode(' ', $chunk);
        }

        return $split;
    }
}

namespace MarkovChain;

use MarkovChain\Tokenizer\TokenizerInterface;

class MarkovChain
{

      @var array

    protected $result = [];


     @var TokenizerInterface|null

    protected $tokenizer = null;
```

```
    MarkovChain constructor.

    @param TokenizerInterface $tokenizer

   public function __construct(TokenizerInterface $tokenizer)
   {
      $this->tokenizer = $tokenizer;
   }


    @param array $input

   public function learn(array $input)
   {
      $matrix = [];

      foreach ($input as $value)
      {
         $array = $this->tokenizer->tokenize($value);

         for ($pos = 1, $length = count($array); $pos < $length; $pos++,
$counter++)
         {
            $counter = &$matrix[ $array[$pos - 1] ][ $array[$pos] ];
         }
      }

      foreach ($matrix as $prev => $array)
      {
         $total = array_sum($array);

         foreach ($array as $next => $count)
         {
            $this->result[$prev][$next] = ($count / $total);
         }
      }
```

```
    }


     @param string $subject
     @return array|null

    public function classify(string $subject)
    {
        if (!isset($this->result[$subject])) {
            return null;
        }

        arsort($this->result[$subject]);

        return $this->result[$subject];
    }
}

declare(strict_types = 1);

namespace MarkovChain\Tokenizer;


  Class CharTokenizer

  @author Patrick Schur <patrick_schur@outlook.de>
  @package MarkovChain\Tokenizer

class CharTokenizer implements TokenizerInterface
{
    public function tokenize(string $string): array
    {
        return preg_split('//u', preg_replace('/[^\pL]+/u', '', $string), -1,
PREG_SPLIT_NO_EMPTY);
    }
}

namespace MarkovChain\Tokenizer;
```

```
  Interface TokenizerInterface

  @author Patrick Schur <patrick_schur@outlook.de>
  @package MarkovChain\Tokenizer

interface TokenizerInterface
{
    public function tokenize(string $string): array;
}

namespace MarkovChain\Tokenizer;


  Class WordTokenizer

  @author Patrick Schur <patrick_schur@outlook.de>
  @package MarkovChain\Tokenizer

class WordTokenizer implements TokenizerInterface
{
    public function tokenize(string $string): array
    {
        return preg_split('/[^\pL]+/u', $string, -1, PREG_SPLIT_NO_EMPTY);
    }
}

class MixedSourceChain extends WordChain
{
    /** @var WordChain  */
    protected $source1Words;

    /** @var string */
    protected $source2;

    /** @var int */
    protected $sizeEach;
```

```
    public function __construct($source1, $source2, $totalWords = 20)
    {
        $this->sizeEach = round($totalWords/2);
        $source1 = $this->prepareContent($source1);
        $source2 = $this->prepareContent($source2);

        $this->source1Words = new WordChain($source1, $this->sizeEach);
        $this->source2 = $source2;
    }

    /**
     * @return string
     */
    public function generate()
    {
        $first = $this->source1Words->getRandomLink();
        $second = $this->splitText($this->source2, 1);

        $split = $this->splitText($first, 1);
        $connector = $split[count($split)-1];

        $search = preg_grep('/\b' . preg_quote($connector, '/') . '\b/', $second);

        // if nothing was found, use the whole sample and get a rand
complement
        if (!count($search)) {
            $search = $second;
        }

        $pos = array_rand($search);
        $complement = array_slice($second, $pos + 1, $this->sizeEach);

        $first[0] = strtoupper($first[0]);
        $complement[count($complement)-1] = $this-
>removeAllPunctuation($complement[count($complement)-1]);

        return $first . ' ' . implode(' ', $complement);
```

```
    }

    public function prepareContent($content)
    {
        $content = strtolower($content);
        $content = str_replace(['.', '-', ';', ':', "", "“"], "", $content);

        return $content;
    }

    public function removeAllPunctuation($content)
    {
        return str_replace([',', '.', '-', ';', ':', "", "!", "?", "\""], "", $content);
    }
}

namespace MarkovChain\Tests;

use MarkovChain\MarkovChain;
use MarkovChain\Tokenizer\CharTokenizer;
use MarkovChain\Tokenizer\WordTokenizer;
use PHPUnit\Framework\TestCase;

  Class MarkovChainTest

  @author Patrick Schur <patrick_schur@outlook.de>
  @package MarkovChain\Tests

class MarkovChainTest extends TestCase
{
    public function testWordTokenizer()
    {
        $c = new MarkovChain(new WordTokenizer());

        $c->learn([
            'a b c d e f g',
            'g e c f a b d',
            'g f e b g a d',
```

```
        ]);
        $this->assertEquals(['b' => 0.6666666666666667, 'd' =>
0.3333333333333333], $c->classify('a'));
    }
    public function testCharTokenizer()
    {
        $c = new MarkovChain(new CharTokenizer());

        $this->assertEquals($c->classify('a')['a'], $c->classify('b')['b']);
    }
    public function testIsNull()
    {
        $c = new MarkovChain(new WordTokenizer());

        $c->learn(['']);

        $this->assertNull($c->classify('a'));
    }
}
```