Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(освітній рівень)

на тему: "Дослідження безперервної автентифікації користувачів на основі динаміки натискання клавіш"

Виконав: студент (ка) VI курсу, групи СБмз-61

Спеціальності:
125 «Кібербезпека та захист інформації»
(шифр і назва напряму підготовки, спеціальності)
Літвінчук Софія Сергіївна
підпис (прізвище та ініціали)

Керівник Загородна Н.В.
підпис (прізвище та ініціали)

Нормоконтроль Стадник М. А.
підпис (прізвище та ініціали)

Завідувач кафедри Загородна Н.В.
підпис (прізвище та ініціали)

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«__» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека та захист інформації
(шифр і назва спеціальності)

Студенту Літвінчук Софії Сергіївні
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження безперервної автентифікації користувачів на основі динаміки натискання клавіш

Керівник роботи Загородна Наталія Володимирівна, к.т.н., доцент, завідувач кафедри КБ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 24 » 11 2025 року № 4/7-1025

2. Термін подання студентом завершеної роботи 12.12.2025

3. Вихідні дані до роботи наукові джерела з тематики поведінкової біометрії та безперервної автентифікації користувача за клавіатурним почерком; теоретичні основи архітектур машинного навчання; датасет Clarkson II; технічні засоби та середовище розробки.

4. Зміст роботи (перелік питань, які потрібно розробити)
Розділ 1 Аналіз проблеми та огляд літератури; 1.1 Проблема автентифікації користувачів в інформаційних системах; 1.1.1 Традиційні методи автентифікації;1.1.2 Біометричні методи автентифікації користувачів;1.2 Безперервна автентифікація на основі динаміки натискання клавіш;1.3 Сучасні підходи та наукові дослідження;Висновки до розділу 1;Розділ 2 Побудова моделі автентифікації користувача за клавіатурним почерком;2.1 Ознаки клавіатурного ритму;2.2 Постановка задачі класифікації; 2.3 Архітектура моделі ;2.3.1 Вхідний шар; 2.3.2 Шари вбудовування ознак;2.3.3 Конкатенація ознак;2.3.4 Рекурентний шар Bi-LSTM;2.3.5 Механізм уваги;2.3.6 Вихідний шар;2.4 Алгоритм безперервної автентифікації користувача; 2.5 Показники оцінювання ефективності;Висновки до розділу 2; Розділ 3 Практична реалізація та аналіз результатів;3.1 Програмне середовище та інструменти реалізації; 3.1.1 Характеристики апаратного забезпечення;3.1.2 Програмне забезпечення та середовище розробки;3.1.3 Структура проєкту;3.2 Датасет та його попередня обробка;3.3 Методика формування експериментальних вибірок;3.4 Програмна реалізація моделі;3.5 Дослідження впливу довжини послідовності натискань клавіш;3.6 Дослідження впливу параметрів формування вибірки;3.7 Дослідження впливу порогового значення;3.8 Дослідження зміни точності моделі з часом; 3.9 Порівняльний аналіз;Висновки до розділу 3;Розділ 4 Охорона праці та безпека в надзвичайних ситуаціях; 4.1 Охорона праці;4.2 Безпека життєдіяльності; Висновки до розділу 4; Висновки; Список використаних джерел; Додакти.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1.Тема, 2.Мета, об`єкт та предмет, 3.Актуальність, 4.Ознаки клавіатурного вводу, 5. Архітектура досліджувальної моделі, 6. Алгоритм безперевної автентифікації користувача, 7. Датасет, попередня обробка даних та формування вибірок. 8.Дослідження впливу довжини послідовності натискання клавіш, 9.Дослідження впливу параметрів формування вибірки, 10. Дослідження впливу порогового значення прийняття рішення, 11.Дослідження зміни точності моделі з часом, 12.Порівняльний аналіз, 13. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Осухівська Г.М., к.т.н., доцент		
Безпека в надзвичайних ситуаціях	Теслюк В. М., проректор з адміністративно-господарської роботи та будівництва		

7. Дата видачі завдання 19.09.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	19.09 – 22.09	Виконано
2.	Підбір та опрацювання наукових джерел	25.09 – 30.09	Виконано
3.	Аналіз проблем та обмежень існуючих методів безперервної автентифікації за клавіатурним почерком	01.10 – 05.10	Виконано
4.	Розроблення архітектури моделі та визначення її ключових компонентів	08.10 – 15.10	Виконано
5.	Проведення досліджень факторів, що впливають на точність результатів роботи моделі	18.10 – 27.10	Виконано
6.	Проведення порівняльного аналізу роботи досліджувальної моделі з альтернативними підходами	28.10 – 09.11	Виконано
7.	Оформлення розділу 1 «Аналіз проблеми та огляд літератури»	10.11 – 14.11	Виконано
8.	Оформлення розділу 2 «Побудова моделі автентифікації користувача за клавіатурним почерком»	15.11 – 22.11	Виконано
9.	Оформлення розділу 3 «Практична реалізація та аналіз результатів»	23.11 – 29.11	Виконано
10.	Оформлення кваліфікаційної роботи	30.11 – 04.12	Виконано
12.	Нормоконтроль	08.12 – 10.12	Виконано
13.	Перевірка на плагіат	12.12 – 14.12	Виконано
14.	Попередній захист кваліфікаційної роботи	20.12 – 15.12	Виконано
15.	Захист кваліфікаційної роботи	24.12.2025	

Студент _____ Літвінчук С.С.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Загородна Н.В.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Дослідження безперервної автентифікації користувачів на основі динаміки натискання клавіш // ОР «Магістр» // Літвінчук Софія Сергіївна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБмз-61 // Тернопіль, 2025 // С. 82, рис. – 18, табл. – 4, кресл. – __ , додат. – 4.

Ключові слова: безперервна автентифікація, клавіатурна динаміка, поведінкова біометрія, Bi-LSTM, глибоке навчання, інформаційна безпека.

У кваліфікаційній роботі виконано дослідження методу безперервної автентифікації користувачів за динамікою натискання клавіш. Проведено аналіз сучасних підходів до поведінкової біометрії та визначено особливості автентифікації в умовах коротких послідовностей клавіатурних подій у неконтрольованих умовах. Програмно реалізовано модель автентифікації на основі архітектури Bi-LSTM з механізмом уваги та протестовано її роботу на даних датасету Clarkson University Keystroke Dataset II.

Проведено експериментальні дослідження впливу ключових параметрів моделі на точність автентифікації, зокрема довжини послідовності, кроку «ковзного вікна», способу розподілу вибірки та порогового значення. Сформовано тренувальні та тестові набори даних із поділом користувачів на легітимних, внутрішніх та зовнішніх порушників. Проведено порівняльний аналіз архітектури Bi-LSTM з механізмом уваги з альтернативними нейронними моделями.

ABSTRACT

Research of continuous user authentication based on keystroke dynamics // Thesis of educational level "Master"// Sofiia Litvinchuk // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, group СБмз-61 // Ternopil, 2025 // p. 82, figs. 18, tbls. 4, drws. _ , apps. 4.

Keywords: continuous authentication, keystroke dynamics, behavioral biometrics, Bi-LSTM, deep learning, information security.

This thesis investigates a method of continuous user authentication based on keystroke dynamics. A review of modern approaches to behavioral biometrics has been conducted, and the specific challenges of authentication under short and uncontrolled keystroke sequences have been identified. An authentication model based on a Bi-LSTM architecture with an attention mechanism was implemented and evaluated using the Clarkson University Keystroke Dataset II.

A series of experimental studies was carried out to examine the impact of key model parameters on authentication accuracy, including sequence length, sliding window step, dataset partitioning strategy, and decision threshold. Training and test sets were constructed with a separation of users into genuine, internal, and external impostors. A comparative analysis of the Bi-LSTM with attention against alternative neural architectures was performed, demonstrating its advantages in scenarios involving short behavioral sequences.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ACC — Accuracy, загальна точність класифікації.

Bi-LSTM — Bidirectional Long Short-Term Memory, двонапрямлена рекурентна нейронна мережа з довгою короткочасною пам’яттю.

CNN — Convolutional Neural Network, згорткова нейронна мережа.

EER — Equal Error Rate, точка рівності помилок першого та другого роду

FAR — False Acceptance Rate, імовірність помилкового прийняття неавторизованого користувача.

FRR — False Rejection Rate, імовірність помилкової відмови авторизованому користувачу.

GRU — Gated Recurrent Unit, рекурентна нейронна мережа з керованими вентилями.

eFAR — external False Acceptance Rate, імовірність помилкового прийняття зовнішнього порушника.

iFAR — internal False Acceptance Rate, імовірність помилкового прийняття внутрішнього порушника.

k-NN — k-Nearest Neighbors, метод k найближчих сусідів.

KCA — Keystroke-based Continuous Authentication, безперервна автентифікація на основі динаміки натискання клавіш.

LSTM — Long Short-Term Memory, рекурентна нейронна мережа з довгою короткочасною пам’яттю.

NB — Naive Bayes, наївний баєсівський класифікатор.

RNN — Recurrent Neural Network, рекурентна нейронна мережа.

SVM — Support Vector Machine, метод опорних векторів.

ВСТУП

У сучасних інформаційних системах питання автентифікації користувачів залишається одним із ключових аспектів кібербезпеки. Традиційні методи, засновані на паролях, PIN-кодах або токенах, мають суттєві недоліки – вони вразливі до крадіжки, підбору чи методів соціальної інженерії. За даними Microsoft [18], понад 99 % компрометацій облікових записів відбуваються через відсутність додаткових засобів перевірки особи. Одноразова автентифікація не гарантує, що саме авторизований користувач продовжує роботу протягом усієї сесії, що створює ризики несанкціонованого доступу до конфіденційних даних, особливо в умовах віддаленої роботи або спільного користування пристроями.

Вирішенням цієї проблеми є впровадження безперервної автентифікації, що забезпечує постійне підтвердження особи користувача під час роботи системи та компенсує недоліки традиційних методів. Серед поведінкових біометричних підходів особливий інтерес становить аналіз динаміки натискання клавіш, який є неінвазивним, простим у реалізації, не потребує додаткового обладнання та може застосовуватися після первинної автентифікації.

За останні десятиліття спостерігається активний розвиток методів безперервної автентифікації користувачів на основі динаміки натискання клавіш (Keystroke-based Continuous Authentication, KCA). У роботі [19] вперше продемонстровано можливість розпізнавання користувачів за часовими характеристиками клавіатурного вводу, а подальші дослідження, зокрема [8], підтвердили стабільність індивідуальних патернів набору тексту та їх придатність для тривалого моніторингу. Надалі увага дослідників була зосереджена на підвищенні точності, стійкості й адаптивності моделей KCA із застосуванням методів машинного та глибокого навчання, що дало змогу ефективно аналізувати послідовності різної довжини в реальних умовах [2, 15].

Попри досягнутий прогрес, низка питань у сфері безперервної автентифікації користувачів залишається відкритою. Зокрема, точність більшості моделей значно знижується в неконтрольованих умовах, коли дані клавіатурного вводу збираються без попереднього інструктування користувача

та без фіксованого тексту, у різних середовищах і за змінних поведінкових факторів. Додатковою складністю є обмежена кількість натискань клавіш у більшості реальних сценаріїв взаємодії, що ускладнює формування стабільних поведінкових шаблонів. Відомо, що стилі набору тексту різних користувачів можуть бути схожими, що підвищує ризик помилкової ідентифікації. Отже, актуальним завданням сучасних досліджень є створення моделей KCA, здатних забезпечувати високу точність автентифікації на основі коротких послідовностей клавіатурного вводу в неконтрольованих умовах.

Метою кваліфікаційної роботи є дослідження ефективності методу безперервної автентифікації користувачів на основі динаміки натискання клавіш, адаптованого до коротких послідовностей клавіатурного вводу в неконтрольованих умовах.

Об’єкт дослідження – процес безперервної автентифікації користувачів в інформаційних систем на основі клавіатурного почерку.

Предмет дослідження – методи збору, оброблення та статистичного аналізу часових характеристик натискання клавіш для підтвердження особи користувача за короткими послідовностями вводу.

Основні завдання кваліфікаційної роботи:

1. Провести ґрунтовний аналіз сучасного стану проблеми автентифікації користувачів в інформаційних системах

2. Дослідити існуючі підходи до безперервної автентифікації користувачів на основі динаміки натискання клавіш: методи збору даних, типи часових ознак, способи формування послідовностей, алгоритми машинного та глибокого навчання, які використовуються для побудови моделей автентифікації.

3. Провести детальний аналіз архітектури обраної моделі автентифікації за клавіатурним почерком.

4. Програмно реалізувати модель безперервної автентифікації на основі рекурентної архітектури Bi-LSTM у поєднанні з механізмом уваги.

5. Провести серію експериментальних досліджень впливу різних факторів та показників на точність класифікації моделі.

6. Порівняти ефективність архітектури, що досліджується у цій роботі з альтернативними моделями.

7. Сформулювати висновки та рекомендації щодо використання запропонованої моделі безперервної автентифікації у практичних інформаційних системах та окреслити перспективи подальших досліджень.

Наукова новизна роботи полягає у поглибленому дослідженні впливу ключових параметрів моделі на точність безперервної автентифікації користувачів за короткими послідовностями клавіатурних подій у неконтрольованих умовах із застосуванням архітектури Bi-LSTM з механізмом уваги. Уперше на даних датасету Clarkson II проведено комплексний експериментальний аналіз впливу довжини послідовності, кроку «ковзного» вікна та способу формування навчальних і тестових вибірок на підсумкову точність моделі. Додатково досліджено довгострокову стабільність моделі, а саме – зміну точності автентифікації в залежності від часової відстані між тренувальними та тестовими даними

Практичне значення роботи полягає в можливості використання отриманих результатів під час проєктування та впровадження систем безперервної автентифікації у програмних комплексах, що потребують підвищеного рівня безпеки.

Перспективним напрямом удосконалення безперервної автентифікації за клавіатурним почерком є використання методів глибокого навчання для автоматичного вилучення та корекції біометричних ознак. Як показано в роботі [11], автоенкодерні моделі здатні ефективно згладжувати аномалії.

Основні результати дослідження були апробовані на XIII науково-технічній конференції «Інформаційні моделі, системи та технології» (м.Тернопіль, Україна). Відповідна наукова публікація наведена у Додатку А.

Я висловлюю щиру подяку професору Daqing Hou (Clarkson University, США) за надання доступу до набору даних Clarkson University Keystroke Dataset II. Отримані дані були використані виключно в наукових цілях у рамках виконання даної кваліфікаційної роботи.

РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМИ ТА ОГЛЯД ЛІТЕРАТУРИ

1.1 Проблема автентифікації користувачів в інформаційних системах

Автентифікація користувача – це процес перевірки та підтвердження його ідентичності перед наданням доступу до інформаційних ресурсів. Вона є ключовим компонентом системи контролю доступу, спрямованої на забезпечення конфіденційності, цілісності та доступності даних. Як зазначає Microsoft [17], автентифікація є основою сучасної кібербезпеки, оскільки саме на цьому етапі визначається, хто має право взаємодіяти з інформаційними системами. Відтак вибір методу автентифікації безпосередньо впливає на рівень захисту, зручність використання та ризики компрометації облікових даних.

Згідно з узагальненням, поданим у дослідженні [2], методи автентифікації користувачів класифікуються на дві ключові категорії – традиційні та біометричні (рис. 1.1).

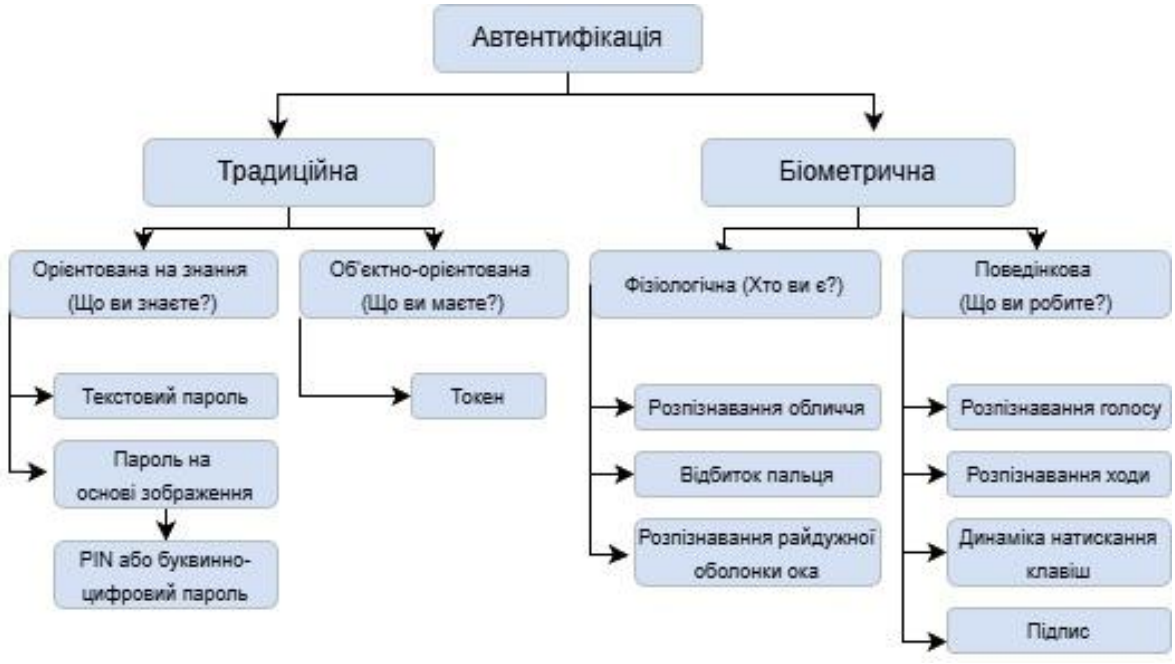

Рисунок 1.1 – Класифікація методів автентифікації користувачів [2]

Представлена класифікація демонструє еволюцію методів автентифікації від простих механізмів перевірки паролів до інтелектуальних систем аналізу поведінки користувача. Зростання кількості кібератак та соціотехнічних загроз

стимулює перехід до більш надійних та адаптивних рішень. У цьому контексті біометричні методи дедалі частіше розглядаються як ключовий напрям підвищення рівня кіберзахисту.

1.1.1 Традиційні методи автентифікації

Традиційні методи автентифікації є найбільш поширеними й історично первинними механізмами перевірки особи в інформаційних системах. Вони становлять основу значної частини сучасних рішень у сфері кібербезпеки, оскільки тривалий час залишалися єдиним доступним способом контролю доступу.

Загалом традиційні методи автентифікації поділяються на дві основні групи: орієнтовані на знання та об’єктно-орієнтовані. Методи першої групи ґрунтуються на інформації, відомій лише користувачу (паролі, PIN-коди, секретні запитання), тоді як методи другої групи передбачають наявність у користувача фізичного носія, зокрема токена, смарт-карти, USB-ключа або генератора одноразових кодів.

Попри появу сучасних біометричних та поведінкових підходів, традиційні методи зберігають низку переваг, що пояснює їхнє тривале домінування. По-перше, вони вирізняються простотою впровадження. Адже більшість систем безпеки вже містять готові механізми роботи з паролями або токенами, що дозволяє інтегрувати їх без додаткових витрат на інфраструктуру. По-друге, такі методи є економічно доступними, оскільки не потребують спеціального обладнання чи використання складних технологій. Крім того, традиційні підходи добре знайомі користувачам, що забезпечує високий рівень прийнятності й мінімальні вимоги до навчання персоналу. Безпекові політики на основі паролів підтримуються практично всіма сучасними операційними системами й сервісами, що робить такі механізми універсальними та сумісними з широким спектром інформаційних платформ.

Незважаючи на широку поширеність, традиційні методи автентифікації мають низку суттєвих обмежень, що істотно знижують рівень захищеності

інформаційних систем. Зокрема, облікові дані користувачів можуть бути скомпрометовані внаслідок фішингових атак, підбору паролів, застосування методів соціальної інженерії або шкідливого програмного забезпечення, у результаті чого секрет, на якому ґрунтується автентифікація, втрачає свою захисну функцію. Значний вплив має і людський фактор, оскільки користувачі часто створюють слабкі паролі, повторно використовують однакові облікові дані або зберігають їх у незахищеному вигляді. Додаткові ризики пов’язані з використанням апаратних носіїв, таких як смарт-карти, USB-ключі чи токени, які можуть бути втрачені, пошкоджені або викрадені, що ускладнює доступ до системи та створює потенційну загрозу безпеці. Крім того, традиційні механізми автентифікації здійснюють перевірку особи лише на етапі входу в систему та не забезпечують подальшого контролю користувача протягом сесії, що не дозволяє своєчасно виявити захоплення активної сесії або заміну користувача.

Традиційні методи автентифікації, попри свою доступність і поширеність, вже не відповідають сучасним вимогам безпеки. Уразливість до компрометації, людський фактор, ризики фізичних носіїв і неможливість безперервної перевірки особи обмежують їхню ефективність. Це створює потребу у впровадженні стійкіших та адаптивних рішень – зокрема біометричних і поведінкових методів, які забезпечують надійний, безперервний контроль автентичності користувача.

1.1.2 Біометричні методи автентифікації користувачів

У відповідь на обмеження традиційних підходів усе більшої популярності набувають біометричні методи автентифікації, які пропонують принципово вищий рівень безпеки.

Біометрична автентифікація ґрунтується на розпізнаванні користувача за його унікальними фізіологічними або поведінковими характеристиками. На відміну від традиційних методів, які спираються на знання чи володіння певним об’єктом, біометричні системи забезпечують прямий зв’язок між ідентичністю користувача та його біологічними або поведінковими ознаками, що значно підвищує надійність перевірки особи.

Згідно з узагальненням, поданим в роботі [7], усі біометричні методи поділяються на дві основні групи – фізіологічні та поведінкові (рис. 1.2).

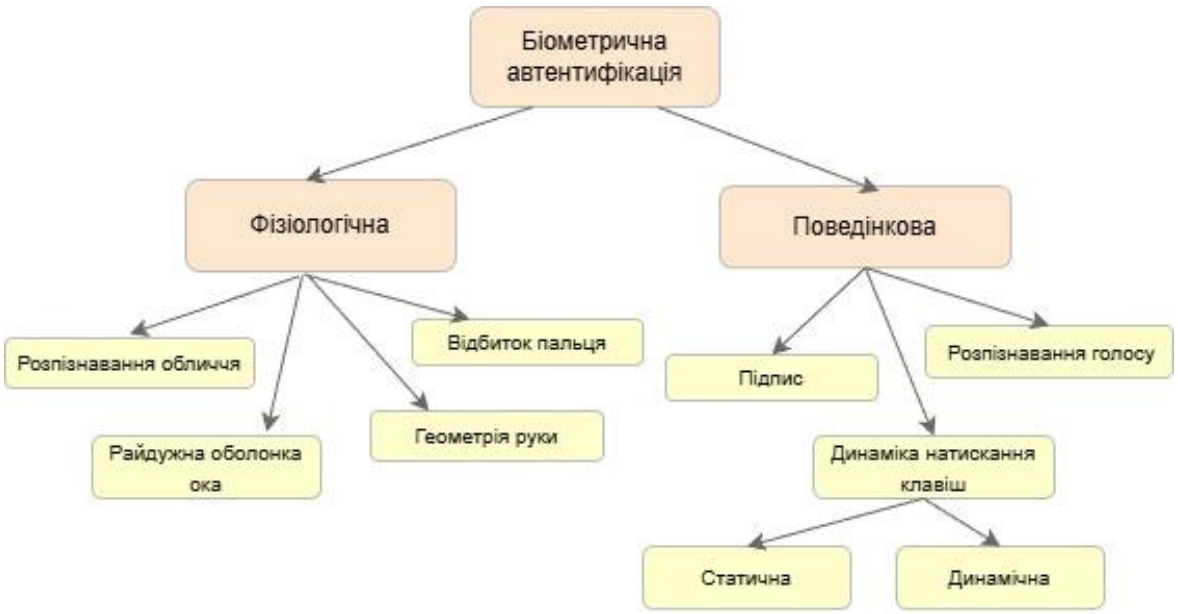

Рисунок 1.2 – Класифікація біометричних методів автентифікації [7]

Фізіологічні методи – методи, що базуються на стабільних біологічних характеристиках людини, які залишаються відносно незмінними протягом життя. До найпоширеніших фізіологічних біометричних методів належать ідентифікація за відбитками пальців, геометрією долоні, розпізнаванням обличчя, райдужки та сітківки ока, а також ДНК-ідентифікація. Зазначені методи характеризуються високим рівнем надійності та стійкістю до фальсифікацій, однак мають певні обмеження, пов’язані з вартістю впровадження, необхідністю використання спеціалізованих сенсорів і питаннями захисту персональних даних.

Поведінкові методи – це методи, що ґрунтуються на аналізі динаміки дій користувача. при цьому біометричні шаблони формуються на основі його індивідуальних моторних і когнітивних особливостей. До основних поведінкових біометричних методів належать автентифікація за динамікою натискання клавіш, розпізнавання голосу, аналіз підпису, а також ідентифікація за ходою. Перевагою поведінкових методів є можливість їх реалізації з використанням стандартних пристроїв без потреби в додатковому обладнанні,

що робить такі підходи придатними для безперервної автентифікації користувачів у процесі роботи з інформаційними системами.

Біометричні методи автентифікації мають низку суттєвих переваг порівняно з традиційними підходами. Використання унікальних фізіологічних та поведінкових характеристик забезпечує високу точність і надійність перевірки особи, що значно зменшує ймовірність помилкового допуску або відмови у доступі. Біометричні дані не можуть бути забуті або випадково передані, а їх підробка є надзвичайно складною, особливо у випадку фізіологічних ознак. Процес автентифікації за допомогою біометрії є зручним для користувача та не потребує запам’ятовування складних секретів чи використання апаратних носіїв, що знижує вплив людського фактора. Крім того, біометричні системи, зокрема поведінкові, дають змогу здійснювати безперервний контроль доступу в режимі реального часу, забезпечуючи своєчасне виявлення аномалій, перехоплення сесії або заміну користувача.

У дослідженні [1] підкреслюється, що біометричні технології забезпечують найкращий баланс між зручністю та захистом, перевершуючи традиційні методи, такі як паролі чи токени. Вони вже стали основою сучасних рішень автентифікації у мобільних пристроях, банківських і онлайн-системах. Водночас такі технології потребують захисту чутливих біометричних шаблонів і надійних механізмів протидії спуфінгу.

З огляду на це зростає інтерес до поведінкових біометричних методів, які не потребують спеціального обладнання та здатні забезпечувати безперервну автентифікацію під час роботи користувача. Одним із найперспективніших напрямів є автентифікація на основі динаміки натискання клавіш Keystroke-based Continuous Authentication, KCA), що розглядається далі.

1.2 Безперервна автентифікація на основі динаміки натискання клавіш

Безперервна автентифікація – це процес постійного підтвердження особи користувача протягом всієї взаємодії з інформаційною системою [23]. На відміну від традиційної (одноразової) автентифікації, яка здійснюється лише під час

входу в систему, безперервна реалізує регулярні перевірки в реальному часі, що дозволяє оперативно виявляти несанкціонований доступ і підвищує загальний рівень безпеки системи.

Одним із найперспективніших підходів до її реалізації є автентифікація на основі динаміки натискання клавіш – поведінковий метод, який аналізує унікальні характеристики набору тексту користувачем: ритм, швидкість, часові інтервали між натисканням і відпусканням клавіш, паузи тощо. Оскільки ці параметри формуються на основі моторних і когнітивних особливостей людини, вони є індивідуальними і важко підроблюваними.

У межах цього підходу розрізняють два основні типи систем:

1. Статичні – виконують перевірку користувача під час введення фіксованого тексту (наприклад, пароля);

2. Безперервні – здійснюють постійний моніторинг набору тексту користувачем в процесі роботи, незалежно від змісту.

Основна різниця між цими підходами, як зазначено у [6], полягає у фазі автентифікації: статичний метод застосовується лише один раз – під час входу в систему, тоді як безперервний забезпечує постійний контроль ідентичності протягом усього сеансу, що підвищує рівень захисту .

Загальна архітектура біометричної системи автентифікації користувачів на основі клавіатурного почерку відповідає типовій структурі біометричних систем (рис. 1.3).

Рисунок 1.3 - Структура біометричної системи автентифікації [2]

Дана архітектура передбачає послідовне виконання кількох функціональних етапів. Спочатку первинні дані користувача надходять від сенсора, після чого у модулі вилучення ознак формується вектор характеристик, що описує індивідуальні властивості взаємодії з пристроєм. Далі отримані ознаки передаються до модуля порівняння, який звертається до бази даних системи та визначає ступінь відповідності між поточними та еталонними шаблонами користувача. На основі результатів порівняння модуль прийняття рішення формує кінцевий висновок щодо автентичності користувача, забезпечуючи допуск або відмову в доступі.

Переваги безперервної автентифікації на основі динаміки натискання клавіш включають неінвазивність, відсутність потреби у спеціальному обладнанні, зручність у використанні та можливість реалізації на будь-якому пристрої з клавіатурою. Серед недоліків варто зазначити чутливість до змін у стані користувача (втома, стрес), залежність від кількості зібраних даних та зниження точності при коротких або неструктурованих текстах.

Попри це, KCA розглядається як один із найефективніших напрямів поведінкової біометрії, придатний для створення інтелектуальних систем безперервного контролю доступу.

Залежно від частоти та способу перевірки користувача, системи KCA поділяються на два основні типи [12]:

1. Епізодичні – здійснюють перевірку користувача після фіксованої кількості дій або за певний часовий інтервал.

2. Справжні безперервні – виконують ідентифікацію після кожного натискання клавіші, забезпечуючи безперервний контроль у режимі реального часу.

Окрім цього, системи безперервної автентифікації класифікують за обсягом «вікна введення», тобто кількістю клавіш, протягом яких збираються дані для аналізу. Виділяють підходи з короткими та довгими послідовностями введення. Використання коротких послідовностей передбачає аналіз невеликих фрагментів набору, що забезпечує швидке прийняття рішень, однак супроводжується зниженням точності автентифікації. Натомість підходи з довгими

послідовностями базуються на аналізі більшого обсягу клавіатурних подій, що підвищує точність результатів, проте призводить до зростання затримки обробки та обчислювальних витрат.

Для забезпечення ефективної роботи системи безперервної автентифікації на основі динаміки натискання клавіш використовуються різні часові характеристики клавіатурного вводу, які відображають індивідуальну моторну поведінку користувача. Як підкреслюють у працях [16, 20], правильне виділення та структуризація ознак є критично важливими для побудови надійних моделей машинного навчання й безпосередньо впливають на точність класифікації. У роботі [6] зазначають, що у процесі аналізу враховуються часові інтервали між подіями натискання та відпускання клавіш, що дозволяє сформувати унікальний профіль користувача.

Залежно від способу формування, виділяють два основні типи ознак клавіатурної динаміки: ознаки окремих клавіш та ознаки пар клавіш.

Ознаки окремих клавіш описують тривалість натискання однієї клавіші та визначаються як різниця між моментами її натискання і відпускання. Ознаки пар клавіш характеризують часові інтервали між подіями двох сусідніх клавіш і широко використовуються в задачах безперервної автентифікації. Зокрема, у праці [26] до цієї групи відносять такі типи часових інтервалів: Press–Press (PP), Release–Release (RR), Press–Release (PR) та Release–Press (RP), які відображають затримки між відповідними подіями натискання та відпускання клавіш.

Крім того, в окремих дослідженнях, зокрема у [6], розглядається загальний час набору слова або фрази, який може виступати інтегральною характеристикою темпу друку користувача.

Ці параметри формують вектор ознак, який подається на вхід моделі класифікації або статистичному алгоритму для прийняття рішення про автентичність користувача.

Для класифікації користувача за сформованим вектором ознак у системах безперервної автентифікації на основі динаміки натискання клавіш застосовуються різні підходи машинного навчання. Залежно від складності даних та вимог до швидкодії системи, їх можна поділити на три основні групи:

1. Традиційні статистичні моделі.

2. Методи ансамблевого навчання.

3. Глибокі нейронні мережі.

До традиційних статистичних моделей належать метод k найближчих сусідів (k-Nearest Neighbors, k-NN), метод опорних векторів (Support Vector Machines, SVM), наївний баєсівський класифікатор (Naïve Bayes, NB) та дерева рішень. Ці підходи забезпечують достатню точність за умови невеликих обсягів даних, відзначаються простотою реалізації та низькими обчислювальними витратами.

Для подолання обмежень окремих базових моделей широкого застосування набули методи ансамблевого навчання, зокрема Random Forest, AdaBoost та Gradient Boosting, які поєднують результати кількох класифікаторів з метою підвищення точності та стійкості до шуму в даних.

Подальший розвиток підходів до автентифікації пов’язаний із застосуванням глибоких нейронних мереж, зокрема згорткових нейронних мереж (CNN), рекурентних нейронних мереж (RNN), а також архітектур LSTM і GRU. Такі моделі є особливо ефективними для аналізу часових послідовностей і динамічних змін у поведінці користувача.

У низці сучасних досліджень також використовуються гібридні моделі, що поєднують нейронні мережі з класичними алгоритмами класифікації, забезпечуючи баланс між точністю та швидкодією.

Тестування проводиться переважно на відкритих наборах даних, серед яких найвідомішими є: CMU Keystroke Dataset, GREYC Dataset, Balabit Dataset, Clarkson Dataset.

Таким чином, сучасні методи безперервної автентифікації на основі динаміки натискання клавіш поєднують поведінкові біометричні характеристики користувача з потужними алгоритмами машинного навчання. Подальші дослідження у цій сфері спрямовані на вдосконалення алгоритмів обробки даних та побудову більш надійних моделей автентифікації.

1.3 Сучасні підходи та наукові дослідження

Упродовж останнього десятиліття безперервна автентифікація на основі динаміки натискання клавіш привертає значну увагу наукової спільноти, що зумовлено потребою у надійних та неінвазивних засобах підтвердження особи. Дослідження у цій сфері зосереджені на вдосконаленні методів збору та аналізу клавіатурних даних, оптимізації вибору ознак, а також застосуванні сучасних алгоритмів машинного навчання для підвищення точності та стійкості системи до поведінкових змін користувача.

Перші спроби систематизації знань у цій галузі представлено у фундаментальній оглядовій роботі [24]. Автори узагальнили низку досліджень, присвячених як статичним, так і безперервним методам автентифікації, класифікувавши їх за типом ознак, довжиною тексту, методами обробки та алгоритмами розпізнавання. Було показано, що класичні статистичні підходи (метод k-найближчих сусідів, метод опорних векторів, наївний баєсівський класифікатор) демонструють точність у межах 85–95 % на контрольованих наборах даних. Водночас у роботі підкреслено ключові проблеми: зниження точності при зміні емоційного стану користувача та потребу в адаптивних моделях, здатних враховувати поведінкові коливання.

Подальший розвиток методів безперервної автентифікації пов’язаний із впровадженням глибинного навчання. Як зазначено у роботах [21, 28], глибинні та класичні ML-методи демонструють високу ефективність у задачах виявлення аномалій, що робить їх перспективними й для систем безперервної автентифікації. У роботі [15] запропоновано комбіновану архітектуру на основі згорткової нейронної мережі та рекурентної нейронної мережі для аналізу вільного тексту. CNN-шари виконують попереднє вилучення ознак, тоді як RNN моделює часову послідовність натискань клавіш, відображаючи індивідуальний ритм користувача. Така комбінація дозволила досягти високої точності (EER ≈ 0.03) і стала одним із перших успішних прикладів застосування глибинних архітектур у KCA-системах.

Зі зростанням інтересу до багатомодальних методів з’явилися дослідження, у яких динаміка натискання клавіш розглядається у поєднанні з іншими поведінковими характеристиками. Так, у праці [26] запропоновано підхід, що інтегрує клавіатурні поведінкові ознаки та ознаки, що отримані під час взаємодії з мишкою. Автори використали метод ансамблевого навчання з об’єднанням незалежних моделей, досягнувши підвищення точності на 8–12 % порівняно з однофакторними рішеннями.

Проблема роботи систем у неконтрольованих умовах була детально досліджена у статті [27]. У ній розроблено метод TKCA, що використовує короткі послідовності (10–15 натискань клавіш) для швидкої ідентифікації користувача. На відміну від більшості моделей, що потребують довгих фрагментів тексту, запропонований метод забезпечує точність близько 92 % при мінімальній затримці прийняття рішення. Алгоритм дозволив досягти балансу між швидкодією та стійкістю до поведінкових змін, що робить цей підхід особливо перспективним для реального застосування.

Актуальність алгоритмів, здатних працювати з мінімальною затримкою, підтверджується й у іншій сфері кібербезпеки: у дослідженні [25] продемонстровано ефективне виявлення Slowloris-атак у реальному часі.

Ще одним напрямом розвитку є спроби зменшити потребу у великих навчальних вибірках. У новітній роботі [14] дослідники застосували сіамську нейронну мережу, яка навчається визначати схожість між парами зразків, а не класифікувати користувачів напряму. Такий підхід дозволяє ефективно працювати навіть при невеликих обсягах даних, забезпечуючи EER менше 2 %.

Загалом, аналіз сучасних досліджень свідчить про поступовий перехід від класичних статистичних методів до інтелектуальних гібридних систем, які поєднують поведінкові, часові та контекстні ознаки. Це все вказує на тенденцію до створення більш адаптивних і надійних систем безперервної автентифікації, здатних працювати в реальних умовах із високою точністю та мінімальним втручанням у роботу користувача.

Висновки до розділу 1

У першому розділі розглянуто проблему автентифікації користувачів та сучасні підходи до захисту інформаційних систем. Показано, що традиційні методи, засновані на паролях і фізичних носіях, дедалі частіше не відповідають вимогам безпеки через вразливість до компрометації та залежність від людського фактора. Особливу увагу приділено біометричним технологіям, які вирізняються здатністю забезпечувати значно вищий рівень надійності завдяки унікальності характеристик користувача та складності їх підробки. У сфері поведінкової біометрії окреме місце займає метод автентифікації за динамікою натискання клавіш, який дає змогу перевіряти особу безперервно під час її роботи з системою.

Аналіз наукових досліджень показав, що безперервна автентифікація на основі динаміки натискання клавіш дозволяє здійснювати постійний моніторинг користувача в реальному часі та забезпечує високу точність завдяки використанню часових характеристик набору тексту й сучасних алгоритмів машинного навчання.

Отримані результати підтверджують актуальність та доцільність використання методів безперервної автентифікації на основі динаміки натискання клавіш як перспективного напрямку поведінкової біометрії. Узагальнений аналіз теоретичних засад і сучасних підходів формує необхідне підґрунтя для подальшого дослідження ефективності роботи вибраної моделі безперервної автентифікації користувачів за клавіатурним почерком.

РОЗДІЛ 2 ПОБУДОВА МОДЕЛІ АВТЕНТИФІКАЦІЇ КОРИСТУВАЧА ЗА КЛАВІАТУРНИМ ПОЧЕРКОМ

2.1 Ознаки клавіатурного ритму

Ознаки клавіатурного ритму відображають динамічні характеристики процесу набору тексту, що є унікальними для кожного користувача. На відміну від статичних біометричних параметрів, такі ознаки описують часові залежності між подіями клавіатурного вводу та відображають індивідуальний стиль друку, темп, ритм та моторну координацію користувача.

У межах даного дослідження було використано два основні часові параметри, що є найбільш інформативними для аналізу клавіатурного ритму та широко застосовуються у поведінковій біометрії:

1. Hold Time (HT) – тривалість утримання клавіші, що визначається як різниця між моментами натискання та відпускання тієї самої клавіші;

2. Down–Down Flight Time (DF) – інтервал часу між натисканнями двох послідовних клавіш.

Для наочності обидва параметри зображено на рис. 2.1, де наведено часову послідовність подій клавіатурного вводу під час натискання та відпускання кількох клавіш.

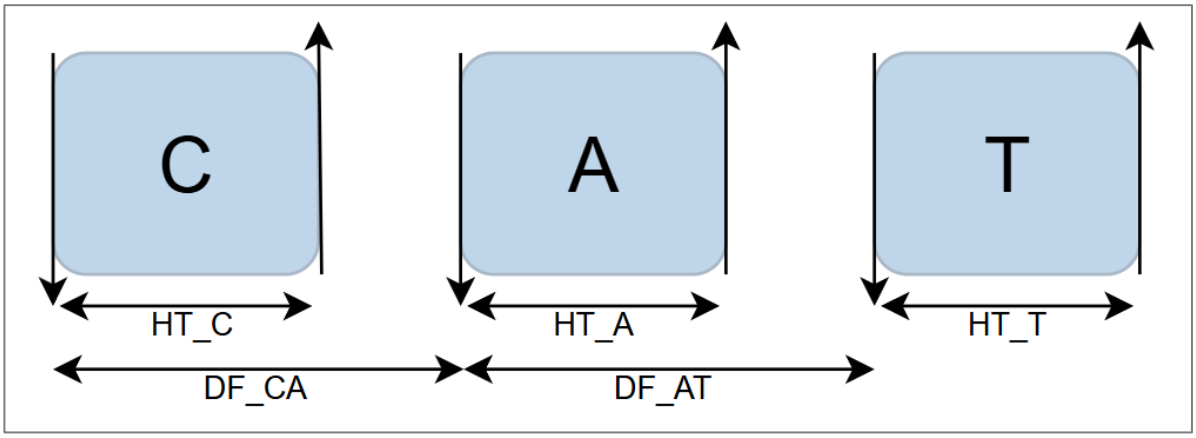

Рисунок 2.1 - Схематичне зображення часових характеристик клавіатурного вводу (Hold Time і Flight Time)

2.2 Постановка задачі класифікації

Завдання автентифікації користувача за клавіатурним почерком у даному дослідженні формулюється як бінарна задача класифікації, у межах якої необхідно визначити, чи належить задана послідовність клавіатурних подій конкретному користувачу. Для кожного користувача будується індивідуальна модель типу one-vs-all, яка розрізняє два класи:

1. Genuine (0) – послідовність сформована відповідним користувачем;
2. Impostor (1) – послідовність належить іншому користувачу.

Таким чином, для кожного користувача виконується окрема процедура навчання, що дозволяє моделі адаптуватися до індивідуальних особливостей його клавіатурного ритму.

Оскільки стиль друку проявляється у часових взаємозв'язках між окремими натисканнями, об'єктом класифікації виступає послідовність клавіатурних подій фіксованої довжини, а не окремі події. Кожна така послідовність має вигляд:

$$S = \{< key_i, ht_i, df_i >\}_{i=1}^{w} \quad (2.1)$$

де key_i – код клавіші,

ht_i– тривалість утримання,

df_i – інтервал між натисканнями,

w – довжина послідовності.

Використання послідовностей дозволяє моделі враховувати ритмічні та динамічні зв'язки між сусідніми подіями, що є критично важливим для розпізнавання індивідуальних особливостей друку. Крім того, такий підхід забезпечує стійкість моделі до варіативності окремих натискань і дозволяє виявляти характерні закономірності клавіатурного почерку.

Для підвищення інформативності вибірки послідовності формуються методом ковзного вікна, що забезпечує максимальне використання доступних даних та зберігає хронологічні зв'язки між подіями.

Модель реалізує відображення:

$$F : S \rightarrow \{0,1\} \quad (2.2)$$

де 0 відповідає автентичній послідовності (genuine),
1 – імпосторській (impostor).

Вихід моделі інтерпретується як показник, що відображає ступінь відповідності або, навпаки, відхилення поточної поведінки введення тексту від еталонного клавіатурного почерку користувача. Чим ближче значення виходу до еталонного профілю, тим більш ймовірно, що певна послідовність належить справжньому користувачу; значні ж відхилення сигналізують про потенційну спробу доступу сторонньої особи. На основі цього значення у подальшому здійснюється прийняття рішення щодо автентичності користувача.

2.3 Архітектура моделі

Архітектура досліджувальної моделі безперервної автентифікації ґрунтується на моделі глибокого навчання Bi-LSTM з механізмом уваги. На рисунку 2.2 подано узагальнену схему моделі, яка відображає її основні структурні блоки та логіку взаємодії між ними – від представлення вхідних послідовностей до формування вихідного рішення щодо автентичності користувача.

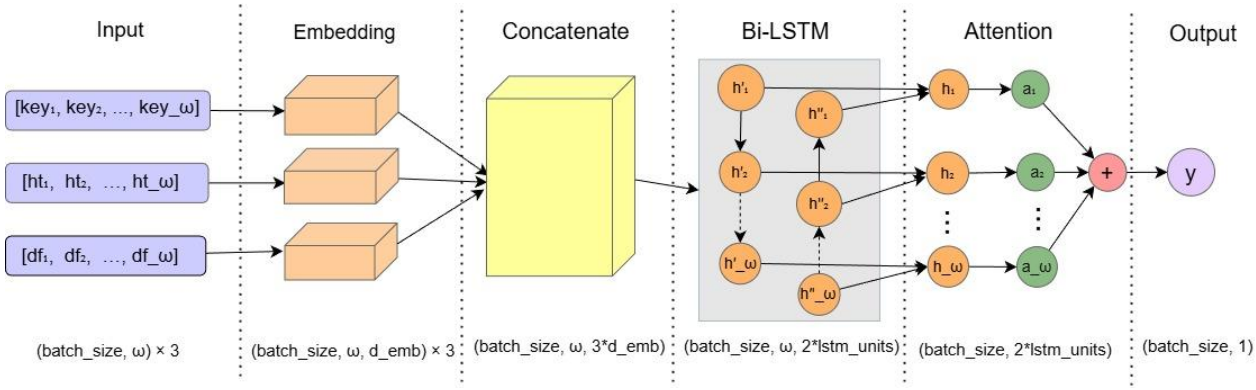

Рисунок 2.2 - Архітектура моделі безперервної автентифікації на основі динаміки натискання клавіш [27]

У подальших підрозділах кожен етап цієї архітектури буде розглянуто детально, з поясненням його функціонального призначення та ролі в процесі автентифікації.

2.3.1 Вхідний шар

Для побудови моделі безперервної автентифікації використовуються часові характеристики клавіатурного вводу, попередньо сформовані у вигляді послідовностей фіксованої довжини. Такий формат даних забезпечує можливість аналізу динаміки друку в межах невеликих фрагментів тексту.

Вхід моделі складається з трьох паралельних каналів, кожен з яких відображає окремий аспект клавіатурного ритму:

1. Послідовність кодів натиснених клавіш (key_ids).
2. Тривалість утримання кожної клавіші (Hold Time, ht_ids).
3. Інтервали між натисканнями клавіш (Down–Down Flight Time, df_ids).

Кожна з цих послідовностей подається у вигляді вектора довжини ω, що відповідає кількості подій клавіатурного вводу в одному «вікні» аналізу. Таким чином, вхід моделі представляє собою три вектори розмірності (1×w), що подаються на відповідні канали нейронної мережі.

Під час навчання дані подаються не по одному, а у вигляді партій (batch), що формують тензори, багатовимірний масив числових даних, розмірності (batch_size×w), де batch_size – це кількість послідовностей, що обробляються моделлю одночасно. Використання батчів підвищує ефективність оптимізації, дозволяє стабілізувати оновлення ваг та прискорює навчання нейронної мережі на графічному або центральному процесорі.

2.3.2 Шари вбудовування ознак

Після формування вхідних послідовностей наступним етапом обробки даних є перетворення дискретних індексованих значень на щільні векторні представлення за допомогою шарів вбудовування (Embedding). Метою цього

етапу є відображення кожного елемента послідовності у багатовимірний простір ознак, у якому модель здатна ефективніше виявляти приховані закономірності та взаємозв’язки між подіями клавіатурного вводу.

Для кожного з трьох каналів – кодів клавіш, тривалостей утримання та інтервалів між натисканнями – використовується окремий шар вбудовування. Це дає змогу моделі незалежно формувати простори ознак для кожного типу інформації, не змішуючи їх на ранніх етапах обробки. Вбудовування мають фіксовану розмірність (d_emb), що забезпечує достатню місткість для представлення поведінкових характеристик без надмірного збільшення обчислювальної складності.

У результаті проходження через шари Embedding кожен канал вхідних даних трансформується у тензор розмірності (batch_size×w×d_emb).

Такі представлення містять уже не окремі значення ознак, а їхні багатовимірні векторні подання, що відображають відносні подібності та відмінності між елементами часової послідовності.

Застосування окремих шарів вбудовування для різних груп ознак дає можливість моделі більш точно інтегрувати структурну і часову інформацію про клавіатурний ритм на наступних етапах нейронної мережі.

2.3.3 Конкатенація ознак

Після перетворення кожного з трьох каналів вхідних даних у щільні векторні представлення за допомогою шарів вбудовування виникає потреба інтегрувати ці представлення в єдину структуру. Оскільки кожен канал відображає різний аспект клавіатурного ритму їх об’єднання дає змогу моделі одночасно враховувати взаємозв’язки між усіма складовими поведінкової послідовності. Для цього на наступному етапі застосовується операція конкатенації (Concatenate), яка виконується по останній осі тензора. У результаті три окремі векторні подання розмірності (batch_size×ω×demb) поєднуються у спільний тензор (batch_size×ω×3demb).

Отримане представлення містить повну інформацію про кожен такт клавіатурного вводу – від натиснутої клавіші до характеристик її утримання та переходу до наступної. Така інтеграція дозволяє моделі розглядати події клавіатурного ритму не ізольовано, а як узгоджену систему ознак.

2.3.4 Рекурентний шар Bi-LSTM

Традиційні архітектури глибинного навчання демонструють низку методологічних обмежень під час аналізу часових послідовностей. Як показано у праці [12], рекурентні нейронні мережі (RNN) характеризуються нестабільністю під час оптимізації, що ускладнює формування стійких довгострокових залежностей у даних. Моделі MLP та CNN не мають механізмів, здатних відображати часову структуру сигналу, і оперують переважно локальними ознаками, що обмежує їхнє застосування в задачах послідовної обробки. Архітектура LSTM частково усуває ці недоліки, однак її односпрямований характер призводить до втрати інформації, що міститься у подальших елементах послідовності.

У зв’язку з цим доцільним є використання двонаправленої архітектури LSTM (Bi-LSTM), яка забезпечує одночасну обробку послідовності у прямому та зворотному напрямках, формуючи повніший контекст. Ефективність такого підходу продемонстровано у дослідженні [9], де Bi-LSTM досягла значно вищої точності порівняно з RNN та звичайними LSTM у задачах класифікації часових сигналів. Для моделювання клавіатурної динаміки це є принципово важливим, оскільки ритм друку залежить як від попередніх, так і від наступних дій користувача.

Структура Bi-LSTM (рис. 2.3) демонструє механізм паралельної обробки часової послідовності у прямому та зворотному напрямках. Нижня гілка моделі відповідає за послідовне опрацювання вхідних даних у їх природному хронологічному порядку, тоді як верхня гілка виконує аналіз послідовності у протилежному напрямку. У кожному кроці часу t ці дві гілки формують окремі приховані стани, які відображають різні часові аспекти сигналу: один

характеризує залежності, пов’язані з попередніми подіями, тоді як інший – інформацію, що зумовлена майбутніми елементами послідовності. Поєднання прихованих станів обох напрямків на вихідному рівні створює цілісне контекстуальне представлення кожного елемента послідовності.

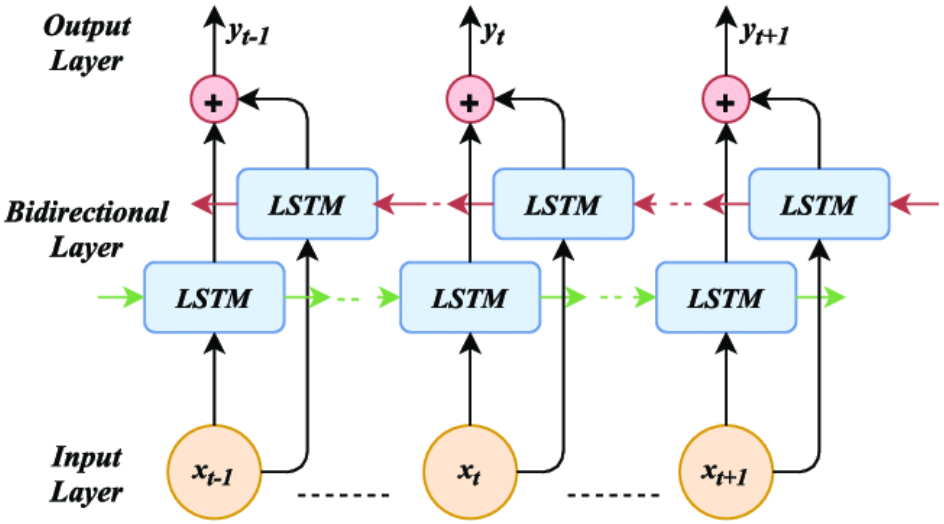

Рисунок 2.3 - Загальна схема BiLSTM [10]

Така двонаправлена архітектура здатна відтворювати складні закономірності, властиві індивідуальному стилю друку, що підвищує точність моделювання у задачах безперервної автентифікації.

Для детального розуміння механізмів обробки часових послідовностей у BiLSTM варто розглянути внутрішню будову базового елемента – нейрона. Його структурну схему наведено на рисунку 2.4.

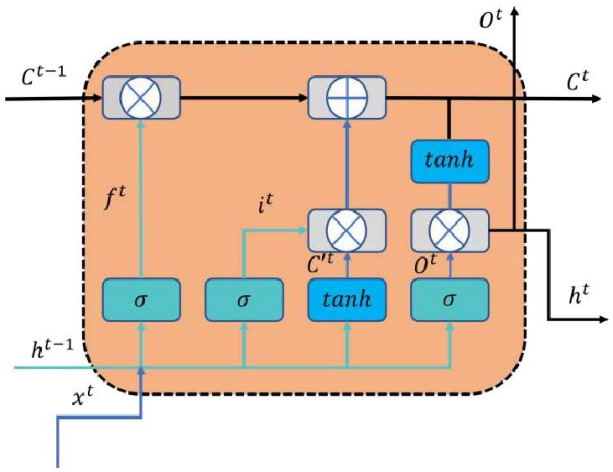

Рисунок 2.4 - Структурна схема нейрона Bi-LSTM [22]

У структурі нейронних мереж типу Bi-LSTM ключову роль відіграє внутрішній механізм перетворення стану, який забезпечує узгоджене оновлення пам’яті та регулювання потоків інформації між тактами послідовності. Цей механізм базується на спеціальній організації рекурентної комірки, що містить декілька керуючих елементів – так званих «воріт»: ворота входу, ворота забуття та ворота виходу. Їхня взаємодія забезпечує можливість ефективного моделювання процесів, у яких значущі залежності можуть охоплювати як короткі, так і тривалі часові інтервали.

Ключовим елементом роботи нейрона Bi-LSTM є внутрішній механізм перетворення стану пам’яті, який забезпечує узгоджене передавання інформації між сусідніми тактами часової послідовності. Його завдання полягає в тому, щоб на основі попереднього стану комірки та нових вхідних даних сформувати актуальний стан, зберігаючи важливі компоненти минулої інформації та інтегруючи релевантні елементи поточного сигналу. Саме цей механізм визначає, які фрагменти акумульованої пам’яті мають бути збережені, а які – оновлені, що є критично важливим для моделювання довготривалих залежностей у послідовності.

Функціональна залежність, яка описує процес переходу від попереднього стану C^{t-1} до нового стану C^{t}, подається у вигляді загального рівняння:

$$C^{t} = f^{t} \times C^{t-1} + i^{t} \times C'^{t} \tag{2.3}$$

де f^{t}–вихідний коефіцієнтом воріт забуття,

i^{t}– вихідний коефіцієнт воріт входу,

C'^{t}– значення кандидатного оновлення стану комірки [22].

Коефіцієнт f^{t}, який входить до формули 2.3 та визначає ступінь збереження попереднього стану пам’яті, задається воротами забуття. Його значення обчислюється на основі попереднього вихідного вектора та поточного вхідного сигналу згідно з функціональною залежністю:

$$f^{t} = \sigma(w_f \cdot [h^{t-1}, x^{t}] + b_f]) \tag{2.4}$$

де σ – сигмоїдна функція,

w_f – матриця ваг для воріт забуття,

h^{t-1} – попередній вихід нейрона,

x^t – поточне вхідне значенням,

b_f – вектор зміщення для воріт забуття [22].

Сигмоїдна функція задається такою математичною формулою:

$$\sigma(z) = \frac{1}{1 - e^{-z}} \tag{2.5}$$

Саме її використання обмежує значення виходу f^t у діапазоні від 0 до 1, що дає змогу інтерпретувати його як коефіцієнт важливості відповідних елементів попереднього стану пам’яті. Значення, що близькі до 1, означають необхідність повного збереження інформації, що міститься у відповідних компонентах C^{t-1}. Натомість значення, близькі до 0, сигналізують про те, що ці компоненти не мають суттєвого впливу на подальший контекст і можуть не враховуватися. Таким чином, ворота забуття виконують роль селектора, який відокремлює релевантні елементи накопиченої пам’яті від шумових або тимчасових патернів, властивих динаміці натискання клавіш.

Паралельно з механізмом вибіркового збереження попередньої інформації формується і надходження нових даних до внутрішнього стану комірки. Цей процес контролюється воротами входу, які визначають, яка частина поточного вхідного сигналу повинна бути інтегрована до оновленого стану пам’яті. Ворота входу генерують коефіцієнт i^t, що характеризує інтенсивність включення нової інформації, і обчислюються за формулою:

$$i^t = \sigma(w_i \cdot [h^{t-1}, x^t] + b_i) \tag{2.6}$$

де $\sigma(\cdot)$ – сигмоїдна функція,

w_i – матриця ваг для воріт входу,

h^{t-1} – попередній вихід нейрона,

x^t – поточне вхідне значенням,

b_i– вектор зміщення для воріт входу [22].

Значення i^t, подібно до коефіцієнта забуття, нормується у діапазоні від 0 до 1, однак виконує іншу функцію: воно визначає, наскільки суттєвий внесок має зробити новий фрагмент послідовності у формування оновленого стану пам'яті. Якщо значення i^tнаближається до 1, модель вважає поточний вхід значущим і включає його до комірки пам'яті; якщо ж воно наближається до 0 – переважно ігнорує цей фрагмент.

Після визначення того, яку частину нової інформації слід допустити до механізму оновлення, формується так зване потенційне оновлення стану комірки, що позначається як C'^t. Воно відображає змістовний внесок поточного вхідного сигналу та попереднього прихованого стану, який може бути інтегрований до пам'яті нейрона залежно від роботи воріт входу.

Математично ця величина визначається рівнянням:

$$C'^t = \tanh(w_c \cdot [h^{t-1}, x^t] + b_c) \quad (2.7)$$

де $\tanh(\cdot)$ – гіперболічний тангенс,

w_c – матриця ваг,

h^{t-1} – попередній вихід нейрона,

x^t – поточне вхідне значенням,

b_c – вектор зміщення для обчислення кандидатного оновлення стану комірки [22].

Після формування попереднього оновлення стану комірки необхідно визначити, яка частина цієї оновленої інформації буде передана назовні – до наступних нейронів BiLSTM. За це відповідають ворота виходу, що генерують коефіцієнт o^t. Їх завдання полягає у дозуванні тієї частини внутрішнього стану, яка має бути відображена у вихідному сигналі нейрона на такті t.

Значення коефіцієнта o^t обчислюється за формулою:

$$o^t = \sigma(w_o \cdot [h^{t-1}, x^t] + b_o) \quad (2.8)$$

де $\sigma(\cdot)$ – сигмоїдна функція,

w_o – матриця ваг для воріт виходу,

h^{t-1} – попередній вихід нейрона,

x^t – поточне вхідне значенням,

b_o– вектор зміщення для воріт виходу [22].

Коефіцієнт o^t визначає, наскільки оновлений стан комірки впливатиме на поточний вихід. Значення, близькі до одиниці, означають, що більша частина внутрішньої інформації буде відображена назовні, тоді як значення, близькі до нуля, обмежують цей вплив.

Таким чином, ворота виходу виконують роль фільтра, що вибірково пропускає лише релевантні компоненти стану пам’яті, необхідні для подальшого аналізу часової послідовності.

Отримавши коефіцієнт o^t , нейрон може сформувати свій остаточний вихідний вектор h^t, який і передається до наступних шарів або часових кроків моделі. Його значення відображає поєднання актуального стану пам’яті та регулятивного впливу воріт виходу.

Формула обчислення виходу має вигляд:

$$h^t = o^t \cdot \tanh(C^t) \tag{2.9}$$

де $\tanh(\cdot)$ – гіперболічний тангенс,

o^t – вихідний коефіцієнт воріт забуття,

C^t - новий стан комірки [22].

Таким чином, вихідний вектор становить завершальний етап внутрішньої роботи LSTM-комірки: він узагальнює попередні обчислення та формує представлення поточного такту, яке зберігає як локальні характеристики, так і довгостроковий контекст часової послідовності. Це робить механізм BiLSTM ефективним для моделювання поведінкових патернів, зокрема динаміки натискання клавіш, де важливо одночасно враховувати як миттєві реакції, так і сталі індивідуальні особливості моторики.

2.3.5 Механізм уваги

Після формування двонаправлених прихованих станів у шарі BiLSTM виникає потреба визначити, які саме елементи часової послідовності роблять найбільший внесок у підсумкове рішення моделі. У задачах поведінкової біометрії різні частини ритму друку можуть мати різну інформативність: одні такти відображають стійкі моторні звички користувача, тоді як інші містять затримки або незначні коливання, пов'язані з випадковими факторами. Саме тому після шару Bi-STM застосовується механізм уваги, який забезпечує зважене агрегування інформації.

Він виконує роль адаптивного фільтра, що аналізує кожний прихований стан Bi-LSTM та присвоює йому певний ваговий коефіцієнт важливості. На відміну від рівномірного усереднення, цей механізм дозволяє моделі самостійно визначити, які частини часової послідовності є ключовими для класифікації користувача. Такий підхід є особливо ефективним для клавіатурної динаміки, де певні фрагменти друку – наприклад, переходи між конкретними клавішами, – мають більшу діагностичну цінність.

Спочатку кожний прихований стан BiLSTM перетворюється у проміжний вектор u_{ti}, що відображає його внутрішні характеристики:

$$u_{ti} = \tanh(wh_{ti} + b) \qquad (2.10)$$

де $\tanh(\cdot)$ – гіперболічний тангенс,

w – матриця ваг механізму уваги,

h_{ti} – вихід BiLSTM на і-му такті часу,

b – вектор зміщення.

Таке перетворення дозволяє виділити ті компоненти прихованого стану, які можуть бути значущими для подальшої класифікації.

Далі визначається ваговий коефіцієнт a_{ti}, що відображає важливість і-го такту послідовності:

$$a_{ti} = \frac{\exp(u_{ti}^{T} u_w)}{\sum_{j=1}^{n} \exp(u_{tj}^{T} u_w)} \quad (2.11)$$

де $\exp(\cdot)$ – експоненціальна функція,

$u_{ti}^{T} u_w$ – скалярний добуток між проміжним вектором і-го елемента та контекстним вектором уваги u_w,

n – кількість елементів (тактових кроків) у послідовності.

У наведеній формулі (2.11) знаменник виконує роль нормувального фактора. Він забезпечує перетворення всіх вагових коефіцієнтів у валідний розподіл, значення якого належать проміжку [0;1], а їхня сума дорівнює одиниці. Така нормалізація відповідає класичній Softmax-функції й гарантує коректне порівняння важливості елементів незалежно від їх абсолютних значень. У результаті нормалізації коефіцієнти a_{ti} можна трактувати як показники того, наскільки важливим є відповідний такт послідовності для подальшого формування контексту.

Фінальним етапом механізму уваги є обчислення підсумкового контекстного вектора s_t, що агрегує інформацію з усієї послідовності:

$$s_t = \sum_i a_{ti} h_{ti} \quad (2.12)$$

де a_{ti} –ваговий коефіцієнт,

h_{ti} – прихований стан.

Елементи послідовності, що отримали вищий ваговий коефіцієнт, матимуть більший вплив на підсумкове представлення.

Таким чином, механізм уваги дає змогу моделі адаптивно виділяти інформативні такти та придушувати несуттєві або шумові фрагменти часової послідовності.

2.3.6 Вихідний шар

Після того як механізм уваги сформував узагальнене контекстуальне представлення послідовності, модель переходить до завершального етапу – отримання прогнозу щодо автентичності. Тут відбувається інтерпретація високорівневих ознак, які були витягнуті попередніми шарами, та їх перетворення у підсумкове рішення.

Отриманий від механізму уваги вектор подається на повнозв'язний шар, який виконує роль класифікатора. Завдання цього шару полягає у тому, щоб зіставити поєднану інформацію про ритм друку зі знаннями, отриманими під час навчання моделі, та повернути скалярне значення, що інтерпретується як ймовірність того, що спостережувана послідовність належить легітимному користувачеві.

Для цього використовується сигмоїдна функція активації, яка відображає вихід у діапазон від нуля до одиниці. Значення, близькі до 0, свідчать про високу ймовірність того, що послідовність є чужою, тоді як значення біля 1 відповідають автентичному користувачу. Такий формат виходу є зручним для подальшого порогового прийняття рішень у контексті бінарної класифікації.

Таким чином, вихідний шар узагальнює попередні обчислення та формує фінальний прогноз, який використовується як основа для процедури безперервної автентифікації. Він забезпечує перехід від внутрішніх представлень часової послідовності до конкретного рішення щодо належності певної клавіатурної динаміки тому чи іншому користувачу.

2.4 Алгоритм безперервної автентифікації користувача

Безперервна автентифікація виконує постійний моніторинг поведінки користувача за динамікою натискання клавіш, аналізуючи останні введені події та періодично приймаючи рішення щодо справжності користувача. Загальна структура алгоритму роботи системи безперервної автентифікації користувача за клавіатурною динамікою подана на рисунку 2.5.

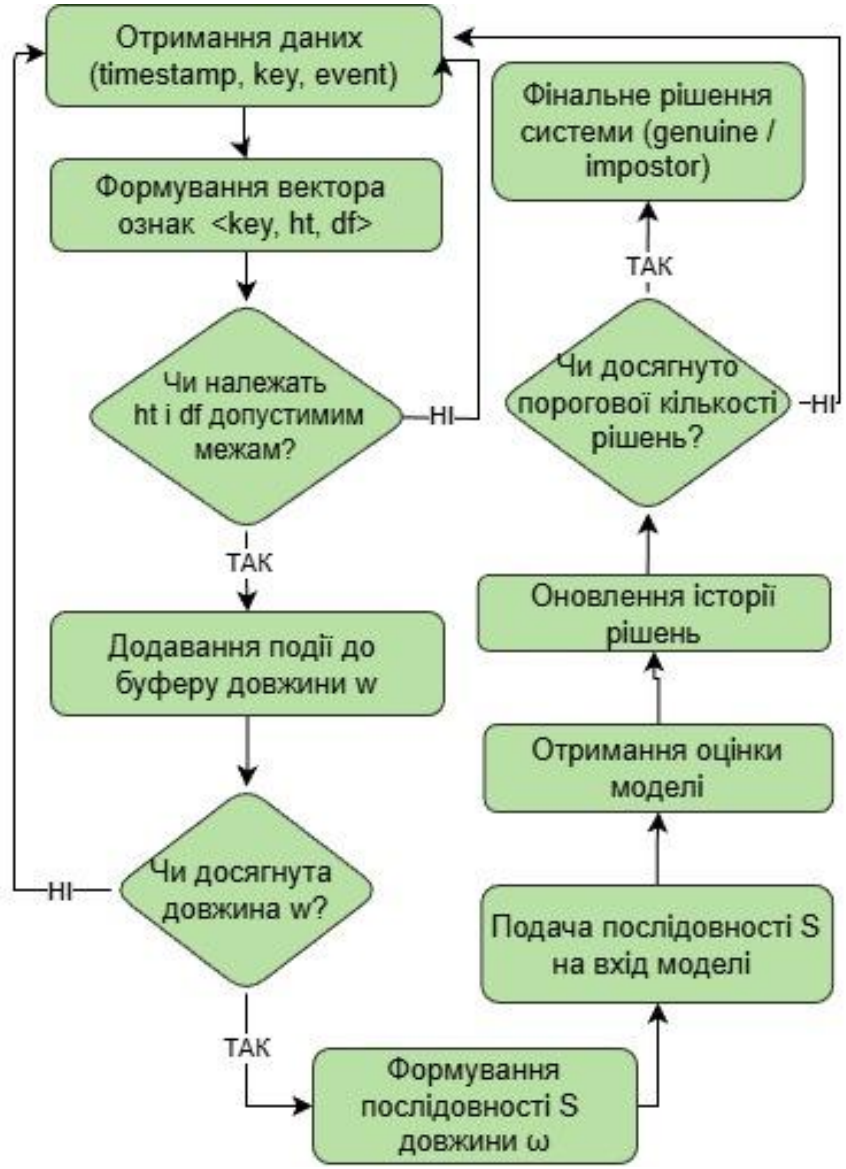

Рисунок 2.5 – Блок-схема алгоритму безперервної автентифікації користувача за клавіатурною динамікою [27]

Перший етап передбачає отримання сирих клавіатурних подій, які складаються з часової мітки, коду клавіші та типу події (натискання або відпускання). На основі послідовності цих подій система обчислює вектор ознак, що містить інформацію про клавішу, час її утримання та інтервал між її настиканням та натисканням наступної клавішін. Після цього виконується перевірка на відповідність значень обчислених часових показників допустимим межам. Події з аномальними значеннями відкидаються, щоб уникнути викривлень через системні паузи, апаратні затримки або шум.

Валідні події додаються до буфера фіксованої довжини ω, який виконує роль ковзного вікна останніх натискань користувача. Значення ω визначає,

скільки останніх клавіатурних дій формують одну послідовність для аналізу. Коли буфер накопичує потрібну кількість елементів, формується послідовність S, яка передається як вхід до моделі, в основі якої Bi-LSTM. Модель повертає оцінку – ймовірність того, що поведінка відповідає легітимному користувачу.

Отриманий результат зберігається в історії рішень, після чого система перевіряє, чи накопичено їх достатньо для винесення остаточного вердикту. Порогова кількість оцінок задається значенням 2n+1, яке використовується як правило більшості. Така кількість забезпечує стійкість до випадкових коливань результатів моделі та підвищує надійність процесу автентифікації. Якщо накопичена кількість рішень менша за порогову, система продовжує збір подій і формування нових послідовностей, працюючи у фоновому режимі та не перериваючи дії користувача. У разі досягнення порогового значення система ухвалює фінальне рішення – «справжній» або «чужий», після чого може ініціювати відповідні заходи безпеки.

Такий підхід забезпечує більшу безперервність і стійкість автентифікації, оскільки рішення формується не за одним фрагментом, а з урахуванням кількох останніх оцінок. Це дає змогу згладжувати випадкові коливання у ритмі друку та підвищує надійність визначення справжності користувача навіть за умов природних змін у його стилі введення.

2.5 Показники оцінювання ефективності

Оцінювання ефективності системи автентифікації здійснюється за допомогою низки кількісних метрик, що характеризують точність прийняття рішень.

У праці [3] виокремлено основні показники, які найчастіше застосовуються для оцінювання ефективності біометричних систем, зокрема систем безперервної автентифікації.

Однією з базових метрик є загальна точність системи Accuracy (ACC), що відображає частку правильних рішень серед усіх проведених перевірок:

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}} \quad (2.13)$$

де TP (true positives) – кількість правильних прийнять,
TN (true negatives) – правильних відхилень,
FP (false positives) – хибних прийнять,
FN (false negatives) – хибних відхилень.

Також використовується показник False Acceptance Rate (FAR), який характеризує ймовірність помилкового прийняття неавторизованого користувача і визначається формулою:

$$\mathrm{FAR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \quad (2.14)$$

Водночас для оцінювання протилежного типу помилок застосовується показник False Rejection Rate (FRR), що відображає ймовірність помилкового відхилення авторизованого користувача та обчислюється таким чином:

$$\mathrm{FRR} = \frac{\mathrm{FN}}{\mathrm{FB} + \mathrm{TP}} \quad (2.15)$$

Важливою узагальненою характеристикою є також показник Equal Error Rate (EER). Це точка рівноваги між показниками FAR і FRR. Чим нижчим є значення цього показника, тим вищою є точність і стабільність системи автентифікації.

Окрім базових показників, у межах даної роботи для більш детального аналізу результатів автентифікації використано розширені метрики точності, які дозволяють розмежувати різні типи помилок та оцінити поведінку системи в умовах як внутрішніх, так і зовнішніх атак. Зокрема, показни iFAR (internal False Acceptance Rate) характеризує частку хибних прийнять серед спроб, здійснених зареєстрованими користувачами, які намагаються видати себе за інших. Водночас показник eFAR (external False Acceptance Rate) відображає частку

хибних прийнять серед спроб, здійснених зовнішніми, тобто незареєстрованими, користувачами.

Використання наведених показників забезпечує об’єктивне порівняння ефективності досліджуваного методу автентифікації та дає змогу оцінити його поведінку в умовах внутрішніх і зовнішніх атак.

Висновки до розділу 2

У цьому розділі було подано та детально розглянуто методику побудови моделі автентифікації користувача за клавіатурною динамікою, яка досліджується у цій роботі. Описано ключові часові ознаки клавіатурного ритму (HT, DF), що формують основу поведінкових характеристик користувача. Задачу автентифікації подано як бінарну класифікацію у форматі one-vs-all, у межах якої модель розрізняє автентичні та неавтентичні послідовності.

Представлено архітектуру моделі, що складається з шарів вбудовування ознак, конкатенації трьох поведінкових каналів, рекурентного шару Bi-LSTM та механізму уваги, який дозволяє виділяти найбільш інформативні елементи послідовності. Вихідний шар формує ймовірнісну оцінку належності певного клавіатурного введення конкретному користувачу.

Окрему увагу приділено алгоритму безперервної автентифікації, який працює з ковзним вікном довжини ω та формує рішення на основі правила більшості. Наприкінці розділу наведено основні метрики оцінювання – ACC, FAR, FRR, EER, а також розширені показники iFAR та eFAR, які дають змогу аналізувати стійкість системи до внутрішніх і зовнішніх атак.

Викладені положення створюють цілісну основу для дослідження ефективності роботи описаної моделі.

РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

3.1 Програмне середовище та інструменти реалізації

3.1.1 Характеристики апаратного забезпечення

Усі етапи проєкту виконувалися на персональному ноутбуці, що функціонує на операційній системі Windows 10. Апаратна конфігурація включає процесор AMD Ryzen 5 3500U з тактовою частотою 2.10 GHz та 8 GB оперативної пам’яті. Система має 64-бітну архітектуру, що забезпечує сумісність із сучасними інструментами для аналізу даних та побудови моделей глибинного навчання.

Зазначені характеристики визначають доступний рівень обчислювальних ресурсів, зокрема продуктивність при виконанні операцій обробки даних та навчання нейронних мереж. Оскільки реалізована модель працює на центральному процесорі без використання апаратного прискорення на основі графічних процесорів (GPU), час тренування є більшим порівняно з системами, оснащеними спеціалізованими обчислювальними прискорювачами. Однак наявних ресурсів достатньо для коректного виконання усіх етапів експерименту, включаючи підготовку даних, навчання моделі та оцінювання її ефективності.

3.1.2 Програмне забезпечення та середовище розробки

Програмна реалізація здійснювалася у середовищі Visual Studio Code, яке було обрано завдяки своїй гнучкості, підтримці численних розширень, інтегрованому терміналу та зручним засобам налагодження коду.

Основною мовою програмування є Python 3.10.4, що забезпечує стабільність, сумісність із сучасними бібліотеками машинного навчання та достатню продуктивність для обробки великих обсягів даних. Python також є стандартом де-факто для побудови моделей машинного навчання завдяки широкій екосистемі інструментів для аналізу даних, статистики та експериментальних досліджень.

У процесі реалізації та експериментального дослідження моделі було використано низку бібліотек і фреймворків. Для роботи з числовими даними та багатовимірними масивами застосовувалася бібліотека NumPy, тоді як Pandas використовувалася для зчитування, попередньої обробки та структурування вхідних даних. Побудова, навчання й оцінювання нейронної мережі здійснювалися за допомогою фреймворку TensorFlow з високорівневим API Keras, що дозволило реалізувати архітектуру з embedding-шарами, двонапрямленою LSTM та механізмом уваги. Для розрахунку метрик оцінювання та аналізу продуктивності моделі використовувалася бібліотека scikit-learn. Додатково модулі ast та os застосовувалися відповідно для безпечного розбору даних, збережених у CSV-файлах у вигляді рядків, і для роботи з файловою системою та організації структури даних.

Вибрані бібліотеки забезпечили достатню швидкодію для виконання навчання моделей у рамках експериментальної частини дослідження.

3.1.3 Структура проєкту

Програмна реалізація дослідження організована у вигляді модульного програмного комплексу, структура якого забезпечує послідовність та відтворюваність усіх етапів обробки клавіатурних даних, формування векторів ознак і навчання моделей автентифікації. Структура проєкту зображено на рисинку 3.1.

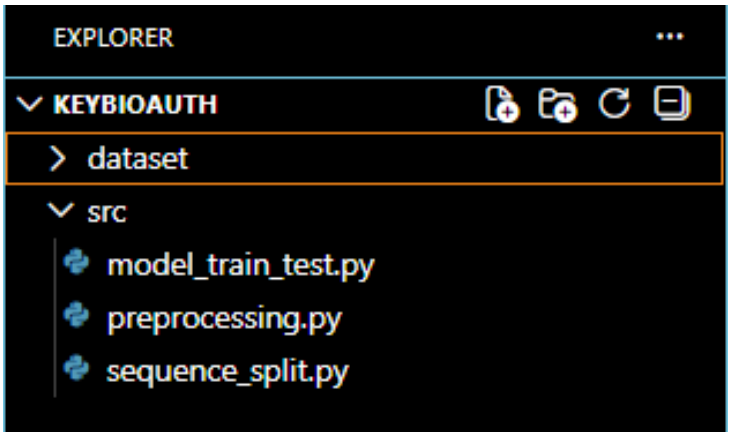

Рисунок 3.1 - Структура проєкту

Програмна реалізація системи автентифікації має модульну структуру та складається з кількох основних компонентів. Вихідні дані зберігаються в каталозі dataset/, який містить сирі дані датасету Clarkson University Keystroke Dataset II у вигляді окремих текстових файлів для кожного користувача з послідовностями клавіатурних подій. Попередня обробка даних здійснюється модулем preprocessing.py (Додаток Б), що відповідає за зчитування, перевірку коректності, узгодження часових характеристик і приведення подій клавіатурного вводу до стандартизованого формату. Результатом цього етапу є інтегрований файл data.csv, який містить очищені та уніфіковані дані для всіх користувачів. Формування послідовностей фіксованої довжини та підготовка тренувальних і тестових вибірок виконується модулем sequence_split.py (Додаток В), який автоматично генерує послідовності клавіатурних подій на основі заданого розміру вікна. Завершальним етапом є модуль model_train_test.py (Додаток Г), у якому реалізовано побудову, навчання та оцінювання моделі автентифікації типу one-vs-all для розрізнення автентичних і імпосторських послідовностей.

3.2 Датасет та його попередня обробка

Для реалізації експериментальної частини дослідження використано Clarkson University Keystroke Dataset II (Clarkson II) – набір клавіатурних даних, сформований дослідницькою групою під керівництвом професора Daqing Hou (Clarkson University, США). Датасет містить записи клавіатурних подій, отримані в умовах вільного набору тексту і відображає природну поведінку користувачів під час взаємодії з клавіатурою.

Доступ до датасету є обмеженим і надається виключно з науковою та некомерційною метою. У межах даної роботи дозвіл було отримано після офіційного запиту. Наданий архів містив 100 файлів, кожен з яких відповідає окремому користувачу. Формат сирих даних є однаковим для всіх файлів і включає три стовпці (рис. 3.2):

1. timestamp – часові мітки подій, задані у тіках Windows FILETIME (1 тік = 100 нс).

2. event_type – тип події: 0 – натискання клавіші; 1 – відпускання клавіші.

3. key_name – символьне позначення клавіші.

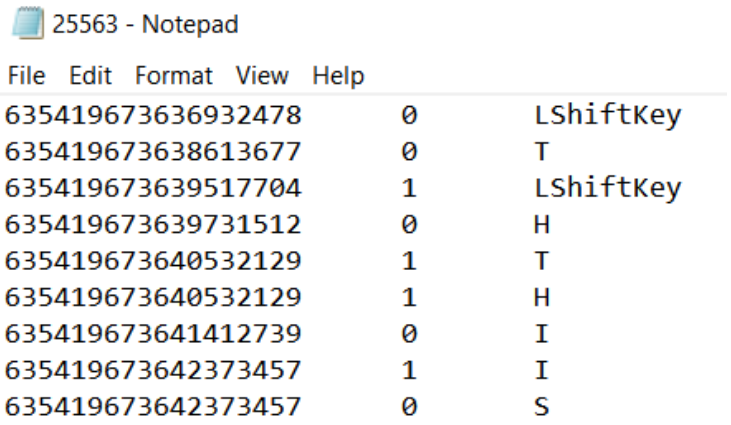

Рисунок 3.2 - Приклад сирих даних клавіатурного вводу з датасету Clarkson II

Оскільки вихідні дані містять шум, дублікати та нефіксовану структуру часових подій, перед формуванням векторів ознак було виконано поетапну попередню обробку, реалізовану програмним модулем preprocessing.py.

На початковому етапі зчитано всі 100 файлів датасету Clarkson University Keystroke Dataset II із загальним обсягом 25 261 511 записів клавіатурних подій. У результаті очищення від дубльованих подій було видалено 1 549 246 записів, після чого залишилося 23 663 334 унікальні події.

Для забезпечення статистичної репрезентативності подальшого аналізу застосовано відбір користувачів із мінімальним порогом 10 000 подій, унаслідок чого було відібрано 84 користувачі. Далі виконано корекцію часових міток, представлених у форматі Windows FILETIME, шляхом усічення молодших бітів з метою усунення шуму та забезпечення фізично коректного обчислення часових інтервалів.

На наступному етапі з очищених подій сформовано короткочасні динамічні характеристики клавіатурного вводу, зокрема тривалість утримання клавіші Hold Time (HT) та інтервал між послідовними натисканнями Down–Down Flight Time (DF). На основі цих параметрів кожну клавіатурну подію було перетворено

на стандартизовану дію, що містить ідентифікатор користувача, часову мітку, назву клавіші та значення HT і DF. Після цього етапу отримано близько 8,15 млн валідних дій.

Для усунення аномальних значень застосовано порогові обмеження HT ≤ 240 мс та DF ≤ 600 мс, що дозволило вилучити 1 691 373 записів і залишити 6 457 881 коректних рядків. Подальша дискретизація часових параметрів з кроком приблизно 6 мс і кодування ознак у вигляді цілочисельних індексів забезпечили сумісність даних із вхідними шарами нейронної мережі.

Після фінальної фільтрації користувачів за мінімальним обсягом 5000 подій остаточний набір даних містив записи для 78 користувачів загальним обсягом 6 430 879 рядків. Дані було впорядковано та збережено у фінальний файл (рис. 3.3), який використовується на наступних етапах формування послідовностей і навчання моделі.

	A	B	C	D	E	F	G
1	user_id,timestamp,key_name,ht,df,key_id,ht_id,df_id						
2	118630,635587541175183200,G,478000.0,1187968.0,41,7,19						
3	118630,635587541178771264,O,900032.0,2400128.0,74,15,40						
4	118630,635587541181171312,G,1100032.0,1000064.0,41,18,16						
5	118630,635587541182171344,L,800016.0,1399936.0,48,13,23						
6	118630,635587541183571360,E,800016.0,600064.0,25,13,10						
7	118630,635587541184171376,OemPeriod,900016.0,1600000.0,82,15,26						
8	118630,635587541185771408,C,1100032.0,1356032.0,6,18,22						
9	118630,635587541187127440,O,780016.0,1716096.0,74,13,28						
10	118630,635587542280823872,RShiftKey,1776048.0,1404160.0,96,29,23						

Рисунок 3.3 - Фрагмент фінального набору даних після попередньої обробки та дискретизації часових характеристик

3.3 Методика формування експериментальних вибірок

На основі очищених та впорядкованих у хронологічному порядку даних для кожного користувача формується множина послідовностей за допомогою ковзного вікна довжини ω, яке послідовно зміщується на крок s. Параметри варіюються в межах експериментів, що дає змогу дослідити вплив глибини контексту та ступеня перекриття між сусідніми фрагментами на точність моделі.

Після формування повного набору послідовностей для кожного користувача здійснюється розподіл даних на тренувальний та тестувальний набори.

Передусім визначається цільовий користувач, для якого будується модель автентифікації. Він розглядається як легітимний (genuine) учасник системи. Усі інші користувачі випадковим чином поділяються на дві групи: внутрішніх порушників та зовнішніх порушників. До групи внутрішніх порушників відносяться ті, чиї дані частково використовуються під час тренування та подальшого тестування. Зовнішні порушники, навпаки, повністю виключаються з етапу навчання і залучаються лише на етапі оцінювання моделі.

Важливо, що для формування навчальних та тестувальних вибірок беруться всі доступні послідовності користувачів відповідної групи. Ці послідовності об’єднуються у два окремі пули – внутрішній та зовнішній.

Після визначення груп користувачів здійснюється розподіл послідовностей легітимного користувача між тренувальним і тестувальним наборами. Для цього застосовується коефіцієнт розподілу, який задає частку даних, що потрапляє до тренувального набору. У межах експериментів досліджуються різні варіанти такого коефіцієнта.

Тренувальний набір складається з частки послідовностей легітимного користувача та такої ж кількості послідовностей із внутрішнього пулу Це забезпечує збалансованість класів у співвідношенні 1:1.

Тестувальний набір формується на основі решти послідовностей легітимного користувача, а також внутрішніх і зовнішніх порушників. Обсяг даних для кожної групи вирівнюється до спільного мінімального значення, що забезпечує пропорційність 1:1:1 між легітимними, внутрішніми та зовнішніми порушниками. Такий підхід гарантує контрольовані умови тестування та надає можливість окремо оцінювати точність виявлення внутрішніх і зовнішніх атак.

Структуру сформованих тренувальних і тестувальних вибірок подано в таблиці 3.1.

Кожен сформований тренувальний або тестувальний набір для окремого користувача зберігається у вигляді окремого CSV-файла, який містить усі необхідні поля для подальшого навчання та оцінювання моделі автентифікації.

Таблиця 3.1 - Структура тренувальних і тестових вибірок

Тип вибірки	Категорія даних	Опис групи користувачів	Значення поля source	Мітка y
Тренувальна (1:1)	Легітимний користувач	Послідовності користувача, для якого створюється модель	genuine	0
	Внутрішні порушники	Послідовності інших користувачів, залучених до етапу навчання	internal	1
Тестувальна (1:1:1)	Легітимний користувач	Послідовності цього ж користувача, не використані в навчанні	genuine	0
	Внутрішні порушники	Інші послідовності тих самих користувачів, були присутні в навчанні	internal	1
	Зовнішні порушники	Послідовності користувачів, які не брали участі в навчанні і є новими для моделі	external	1

Структура файлів є однаковою як для тренувальних, так і для тестувальних даних. Зокрема, кожен рядок відповідає одній сформованій послідовності ковзного вікна та містить такі поля:

1. key_ids – послідовність кодів натиснутих клавіш довжини w, перетворена у фіксований масив числових значень.

2. ht_ids – відповідна послідовність значень тривалостей натискання клавіш.

3. df_ids – відповідна послідовність затримок між натисканнями клавіш

4. y – цільова мітка класу (0 – легітимний користувач, 1 – порушник).

5. source – ідентифікатор походження послідовності (genuine – послідовність належить цільовому користувачу, internal – послідовність належить порушнику).

6. external – послідовність сформована користувачем, що не був присутній у тренувальних даних.

Описана методика формування експериментальних вибірок забезпечує відтворюваність експериментів, коректну побудову тренувальних та тестувальних наборів і створює умови для об’єктивного оцінювання поведінкової моделі автентифікації.

3.4 Програмна реалізація моделі

Програмна реалізація моделі автентифікації здійснена на основі фреймворку TensorFlow/Keras, що забезпечує зручні засоби для побудови багатоканальних архітектур обробки часових послідовностей. Функцію побудови нейронної мережі наведено на рисунку 3.4.

```
def build_model(vocab_key, vocab_ht, vocab_df,
seq_len=SEQUENCE_LENGTH):
    inp_keys = Input(shape=(seq_len,), name="keys")
    inp_hts  = Input(shape=(seq_len,), name="hts")
    inp_dfs  = Input(shape=(seq_len,), name="dfs")
    emb_keys = Embedding(vocab_key, EMBEDDING_DIM)(inp_keys)
    emb_hts  = Embedding(vocab_ht,  EMBEDDING_DIM)(inp_hts)
    emb_dfs  = Embedding(vocab_df,  EMBEDDING_DIM)(inp_dfs)
    merged = Concatenate(axis=-1)([emb_keys, emb_hts, emb_dfs])
    x = Bidirectional(LSTM(LSTM_UNITS, dropout=0.5,
return_sequences=True))(merged)
    x = Attention()([x, x])
    x = tf.keras.layers.GlobalAveragePoolinglD()(x)
    x = Dropout(0.4)(x)
    out = Dense(1, activation="sigmoid")(x)
    model = Model(inputs=[inp_keys, inp_hts, inp_dfs], outputs=out)
    model.compile(optimizer=Adam(LEARNING_RATE),loss="binary_crossen
tropy",metrics=["accuracy"])
    return model
```

Рисунок 3.4 – Функція побудови моделі Bi-LSTM+Attention

Модель працює з трьома паралельними вхідними послідовностями фіксованої довжини: кодами клавіш, тривалістю їх утримання та інтервалами між натисканнями. Ці дані подаються у вигляді індексованих векторів, що

забезпечує компактне представлення часових характеристик клавіатурного ритму.

Усі три канали проходять через окремі шари вбудовування, після чого об’єднуються в єдине представлення та подаються на двонапрямлений LSTM-шар. Рекурентна обробка дає змогу моделі враховувати контекст послідовності як у прямому, так і в зворотному напрямках. Над прихованими станами застосовується механізм уваги, який виділяє найбільш інформативні частини поведінкової послідовності.

На виході модель формує скалярну оцінку – ймовірність того, що дана послідовність належить легітимному користувачу.

Оскільки якість навчання значною мірою залежить від правильно підібраних гіперпараметрів, у межах дослідження було сформовано набір ключових параметрів, що визначають поведінку моделі під час обробки послідовностей клавіатурних подій. Узагальнений перелік цих параметрів та їх значень наведено в таблиці 3.2. Кожен із них суттєво впливає на здатність моделі виявляти закономірності динаміки набору тексту, а також на стабільність і швидкість процесу навчання.

Таблиця 3.2 - Основні гіперпараметри моделі та їх значення

Гіперпараметр	Значення	Опис
Розмір векторів вбудовування	128	Розмірність простору ознак для key_ids, ht_ids і df_ids.
Кількість LSTM-одиниць	256	Розмір прихованого стану у двонапрямленому рекурентному шарі.
Dropout у LSTM-шарі	0.5	Регуляризація для зменшення перенавчання
Dropout після механізму уваги	0.4	Друга стадія регуляризації перед класифікацією
Оптимізатор	Adam	Алгоритм оптимізації ваг під час навчання
Функція втрат	binary cross-entropy	Функція втрат, що використовується для бінарної класифікації

Продовження таблиці 3.2

Гіперпараметр	Значення	Опис
Розмір batch	32	Кількість послідовностей, що обробляються за одну ітерацію навчання.
Кількість епох	15	Максимальна кількість циклів навчання моделі.
Механізм EarlyStopping	patience = 3	Припиняє навчання за відсутності покращення на валідації.

Значення гіперпараметрів було обрано, базуючись на опрацьованих наукових працях, присвячених методам KCA, а також на результатах власних експериментів, що дозволило визначити найбільш ефективну конфігурацію моделі.

3.5 Дослідження впливу довжини послідовності натискань клавіш

Метою цього експерименту є визначення того, як зміна довжини послідовності натискань клавіш (w) впливає на здатністьмоделі автентифікації та рівень помилкових рішень.

Довжина послідовності визначає обсяг поведінкової інформації, що потрапляє до одного вхідного фрагмента, тому її вибір є критично важливим для якості автентифікації.

Під час дослідження було протестовано чотири варіанти довжин: 15, 30, 45 та 60. Отримані результати подано в таблиці 3.3, а узагальнену динаміку зміни основних показників наведено на рисунку 3.5.

Таблиця 3.3 - Результати моделі при різних довжинах послідовності клавіатурних подій

w	Acc_genuine, %	Acc_total, %	FRR, %	iFAR, %	eFAR, %	FAR, %	EER, %
15	96,54	73,19	3,46	0,29	76,68	38,49	29,03

Продовження таблиці 3.3

w	Acc_genuine, %	Acc_total, %	FRR, %	iFAR, %	eFAR, %	FAR, %	EER, %
30	88,43	92,69	12,62	0,29	12,99	6,53	7,83
45	99,74	71,62	0,26	0,52	84,37	42,44	31,16
60	99,87	70,89	0,12	0,18	87,22	43,61	29,30

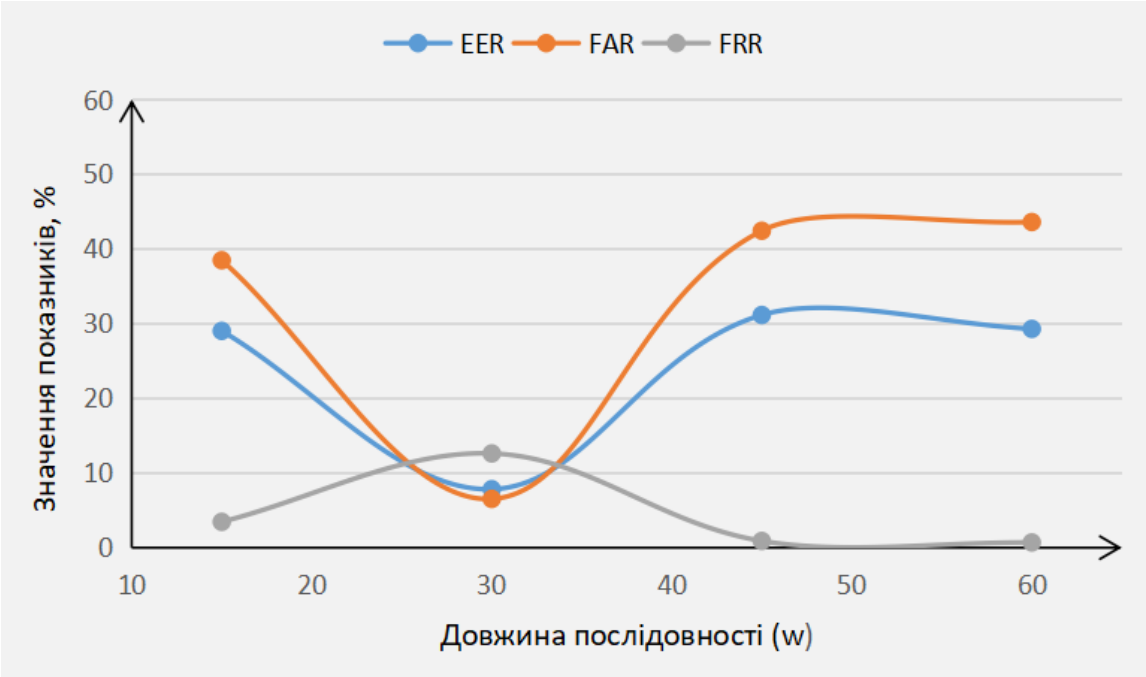

Рисунок 3.5 – Вплив довжини послідовності на показники точності моделі

Короткі послідовності (w = 15) демонструють дуже високу точність для легітимних користувачів (Acc_genuine ≈ 96,5 %), однак мають низьку загальною точністю та значні помилки прийняття нелегітимних користувачів (FAR понад 38,5 %). Такий дисбаланс пояснюється тим, що короткі послідовності містять надто мало інформації про динаміку друку. Унаслідок цього модель формує спрощене уявлення про «свого» користувача й помилково відносить більшість подібних фрагментів до автентичних. Оскільки схожі короткі патерни трапляються й у інших користувачів, модель не здатна чітко їх розмежувати, що призводить до різкого зростання хибних прийнять. Тому висока точність для автентичних послідовностей у цьому випадку не є показником якості, а свідчить про слабку здатність моделі відокремлювати справжні послідовності від чужих.

Зі збільшенням довжини послідовності до 30 елементів спостерігається суттєве покращення всіх ключових показників якості. Загальна точність моделі зростає до 92,69 %, а ймовірність помилкового прийняття чужих послідовностей зменшується більш ніж у шість разів порівняно з варіантом, де послідовність містила лише 15 подій. Також істотно знижується показник EER, досягаючи значення 7,83 %, що є найкращим серед усіх протестованих випадків. Довжина у 30 елементів надає моделі достатній контекст клавіатурної поведінки, дозволяючи їй ефективно розмежовувати справжні та чужі послідовності без накопичення зайвого шуму чи надмірної повторюваності інформації.

Для довших послідовностей , що містять 45 та 60 подій, результати виявилися суттєво гіршими. При довжині 45 загальна точність знижується до 88,95 %, а частка помилкових прийнять чужих послідовностей зростає понад 42 %. Значення EER у цьому випадку перевищує 14 %, що майже вдвічі більше, ніж для оптимального варіанта.

Аналогічна тенденція простежується і для 60 подій: точність падає до 88,29 %, а EER досягає 15,14 %. Погіршення результатів для довших послідовностей пов’язане з тим, що клавіатурний почерк користувача не є стабільним на великих часових інтервалах. У такі послідовності потрапляє більше випадкових пауз, змін темпу та дрібних коливань моторики, які не несуть корисної інформації для автентифікації. Це підвищує шум у даних і ускладнює роботу моделі, оскільки вона змушена обробляти багато другорядних фрагментів замість зосереджуватися на стійких закономірностях. У результаті знижується здатність моделі узагальнювати поведінку користувачів, зростає кількість хибних прийнять, а показник EER погіршується.

Таким чином, найефективнішою виявилася довжина послідовності у 30 подій, оскільки коротші фрагменти не забезпечують достатнього контексту, а довші створюють надлишковий шум і погіршують точність моделі.

3.6 Дослідження впливу параметрів формування вибірки

У цьому підрозділі аналізується вплив двох ключових параметрів формування вибірки – коефіцієнта розподілу даних між тренувальною та тестувальною частинами і кроку «ковзного вікна» – на якість моделі автентифікації.

У межах експерименту було розглянуто два варіанти розподілу даних: 70 на 30 та 50 на 50. Крок «ковзного вікна» також змінювався у трьох значеннях – 1, 15 та 30, що відповідають низькому, середньому та високому рівню перекриття між сусідніми послідовностями. Саме ці значення були обрані на підставі результатів попереднього дослідження впливу довжини послідовності, де оптимальним виявилося значення 30.

Для кожної комбінації параметрів було сформовано тренувальні та тестувальні набори згідно з методикою, описаною у підрозділі 3.3. Після навчання моделі для кожного користувача обчислювалися показники точності.. Узагальнені результати наведено у таблиці 3.4.

Таблиця 3.4 - Результати моделі при різних розподілах вибірки та кроках «ковзного вікна»

Розподіл	Крок	Acc_genuine, %	Acc_total, %	FRR, %	iFAR, %	eFAR, %	FAR, %	EER, %
70 на 30	1	88,43	92,69	12,62	0,29	12,99	6,53	7,83
	15	80,11	82,37	26,43	4,71	21,76	13,23	15,27
	30	73,15	81,79	26,84	5,44	22,32	13,84	17,49
50 на 50	1	76,48	89,41	23,51	0,21	8,11	4,12	9,76
	15	77,65	83,19	25,64	4,5	20,27	12,38	17,26
	30	77,67	83,47	22,32	5,9	21,46	13,68	16,97

Графічна інтерпретація результатів наведена на рисунках 3.6 та 3.7.

Результати показують, що найкраща якість спостерігається при кроці 1, незалежно від варіанта розподілу. У цьому режимі формується найбільша

кількість перекривних послідовностей, що забезпечує модель великим та різноманітним набором прикладів.

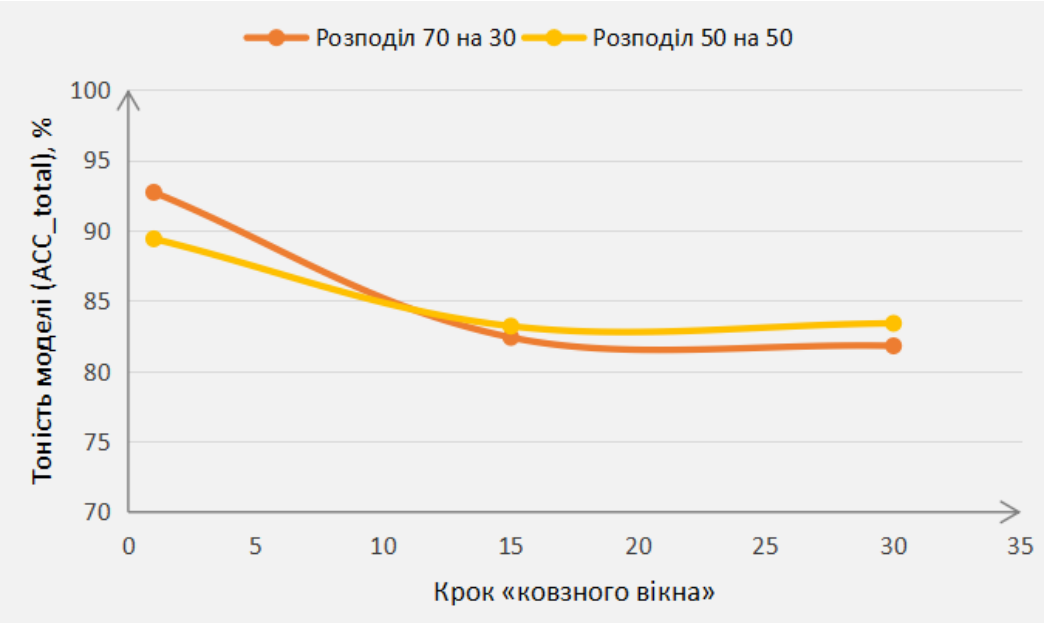

Рисунок 3.6 – Вплив кроку «ковзного вікна» на точність для різних варіантів розподілу вибірки

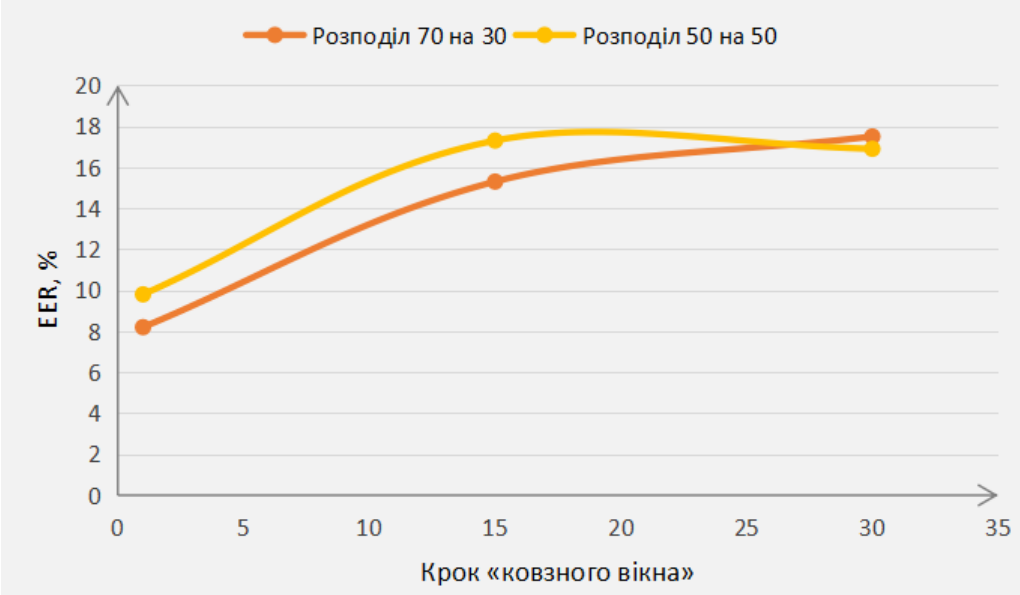

Рисунок 3.7 – Вплив кроку «ковзного вікна» на EER для різних варіантів розподілу вибірки

Для розподілу 70 на 30 та кроку 1 досягається найвища загальна точність – понад 92 %, низьке значення FRR та мінімальні показники хибних прийнять як внутрішніх, так і зовнішніх користувачів. У варіанті 50 на 50 спостерігається

дещо нижча точність (близько 89 %), що логічно пов’язано з меншим обсягом тренувальних даних, однак і тут крок 1 забезпечує найкращий баланс між помилками першого і другого роду.

При переході до кроку 15 якість моделі помітно погіршується. Це спостерігається в обох варіантах розподілу: загальна точність падає до рівня приблизно 82–83 %, а показники FAR і EER суттєво зростають.

Найгірші результати отримано для кроку 30 – незалежно від розподілу вибірки. За цього значення точність наближається до 81–83 %, натомість рівень хибних прийнять (як iFAR, так і eFAR) зростає до понад 21–22 %, що створює суттєві ризики для безпеки. Великі інтервали між послідовностями призводять до втрати значущих проміжних фрагментів поведінки користувача, тому модель спирається на менш повні та менш репрезентативні приклади. У поєднанні з меншою кількістю тренувальних даних (особливо при розподілі 50 на 50) це призводить до значного погіршення EER, який сягає 16–17 %.

Отож, для моделі автентифікації вирішальне значення мають достатній обсяг тренувальних даних та високий рівень перекриття між сусідніми послідовностями. Саме така комбінація забезпечує найповніше охоплення поведінкових патернів і мінімізує кількість помилок. Відповідно, оптимальним варіантом у проведених експериментах став розподіл 70 на 30 у поєднанні з кроком «ковзного вікна» 1.

3.7 Дослідження впливу порогового значення

У моделі автентифікації рішення про належність послідовності легітимному користувачеві приймається на основі порогового значення: якщо вихідна ймовірність перевищує цей поріг, подія класифікується як автентична, інакше – як чужа. Зміна порогового значення безпосередньо впливає на баланс між двома типами помилок: якщо підвищити поріг, модель рідше пропускатиме чужі послідовності, але частіше відхилятиме справжні; якщо знизити поріг – навпаки, легітимні користувачі будуть рідше отримувати відмову, однак зростає ризик неправильної авторизації сторонніх осіб. Тому важливо оцінити, як модель

поводиться у широкому діапазоні порогових значень, оскільки інший поріг може виявитися більш оптимальним для конкретного користувача або сценарію безпеки.

У цьому експерименті для декількох користувачів були побудовані ROC-криві, що ілюструють зміну показників FAR та FRR під час поступової зміни порогового значення в інтервалі від 0 до 1. На рисунку 3.8 подано приклади таких кривих. Кожна з них відображає індивідуальні особливості поведінки моделі для конкретного користувача та дозволяє оцінити, наскільки точно модель розмежовує справжні та чужі послідовності в умовах варіації порога.

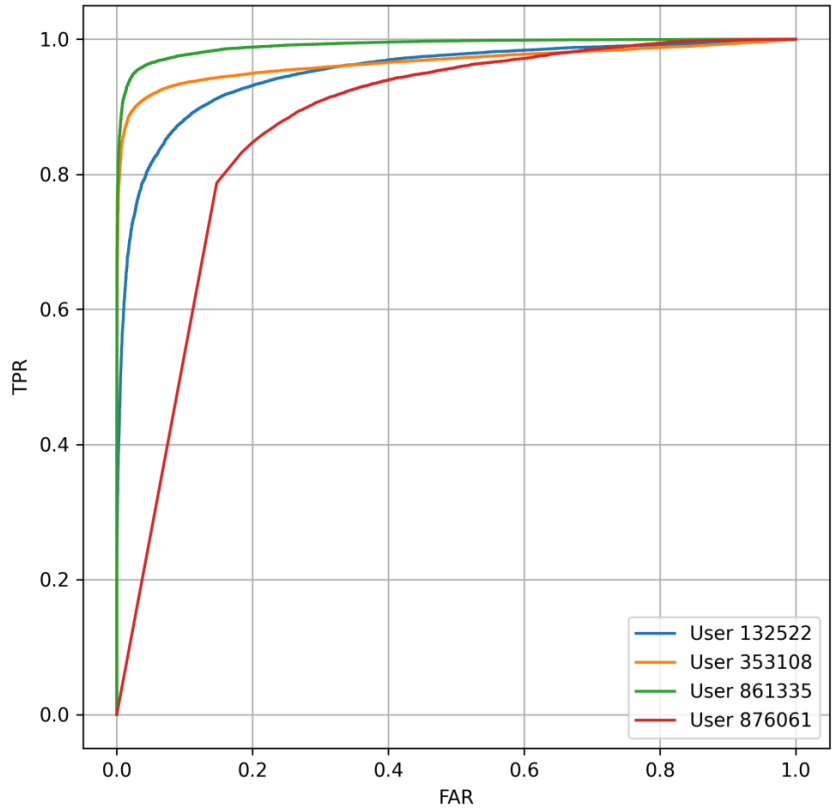

Рисунок 3.8 - ROC-криві моделі автентифікації для різних користувачів

Аналіз ROC-кривих показує, що ефективність моделі помітно відрізняється між користувачами, що відображає індивідуальні особливості їхньої клавіатурної поведінки. Для більшості користувачів криві розташовані у верхній лівій частині графіка, демонструючи високе значення TPR за низького FAR і, відповідно, хорошу здатність моделі відокремлювати справжні послідовності від чужих. Користувачі 353108 та 861335 мають найбільш «круті» криві, що свідчить

про стабільно високу точність у широкому діапазоні порогів. Натомість для користувача 876061 крива суттєво нижча, що вказує на вищу кількість хибних прийнять або відмов. Це свідчить, що оптимальний поріг може бути різним для різних користувачів, а персональне налаштування порога здатне істотно зменшити помилки моделі. Загалом ROC-аналіз підтверджує, що модель працює стабільно, але її можна додатково покращити завдяки індивідуальній адаптації порогового значення.

3.8 Дослідження зміни точності моделі з часом

Метою цього експерименту було оцінити, наскільки якість моделі зберігається за умов збільшення часової відстані між тренувальними і тестовими даними, адже клавіатурний почерк користувача може змінюватися під впливом зовнішніх умов, фізичного стану, накопичення досвіду.

Для аналізу було відібрано користувачів, що мали клавіатурних події, зібрані протягом 1-2 років. Тестова вибірка для кожного з них була поділена на ранню, середню та пізню сесії, що відповідають даним, зафіксованим у різні часові відрізки після початкового навчання. Результати тестування моделі на кожній із цих вибірок зображено на рисунку 3.9.

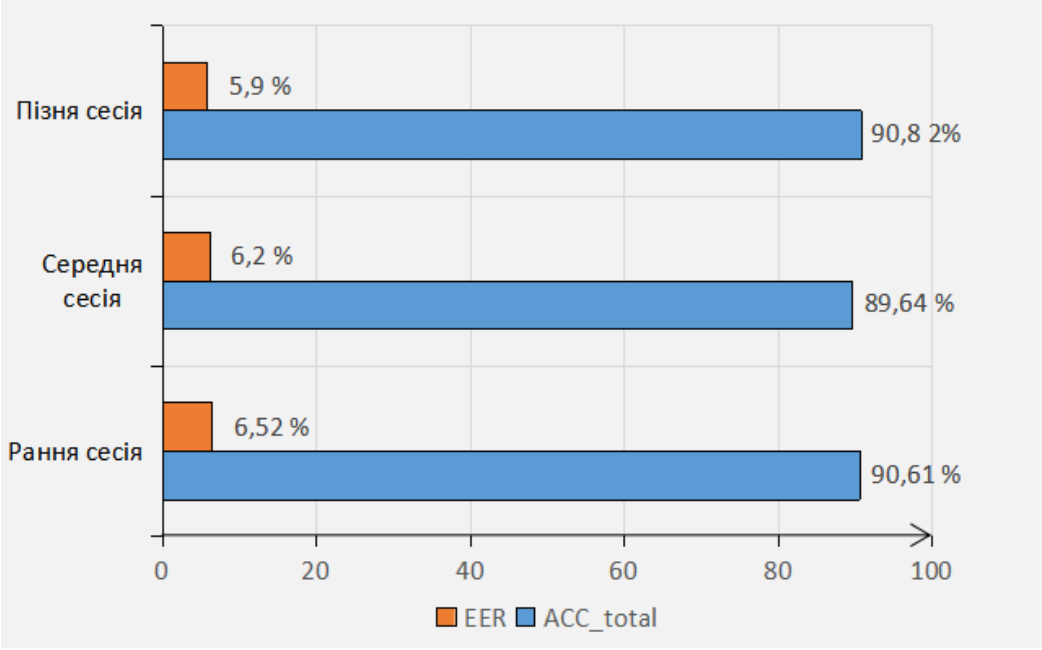

Рисунок 3.9 - Точність моделі на різних часових сесіях

Отримані результати свідчать про загальну стабільність моделі в часі. Значення ACC_total для всіх трьох часових інтервалів залишаються на високому рівні – від 89,6 % до 90,8 %, а показник EER демонструє лише незначні коливання. Це вказує на те, що модель зберігає здатність коректно розмежовувати легітимні та нелегітимні послідовності навіть тоді, коли тестові дані були зібрані через значний час після тренування. Легке зниження або підвищення точності може бути пов’язане з природною варіативністю ритму друку, однак загальна тенденція свідчить про відсутність суттєвої деградації моделі.

3.9 Порівняльний аналіз

Для перевірки ефективності обраної для дослідження архітектури було проведено порівняльний аналіз з іншими нейронними моделями, що традиційно застосовуються в задачах обробки часових послідовностей. У межах експерименту було розглянуто чотири архітектури: CNN, LSTM, Bi-LSTM та Bi-LSTM з механізмом уваги (досліджувальна модель). При цьому принципи формування вибірки, довжина послідовності, крок ковзного вікна та умови навчання залишалися незмінними. Змінювалася лише функція побудови моделі, що дало змогу оцінити внесок саме архітектури в загальну якість автентифікації.

На рисунку 3.10 наведено порівняння моделей за двома ключовими метриками – загальною точністю та показником EER.

Результати показують, що різні архітектури нейронних мереж по-різному справляються з задачею автентифікації за клавіатурною динамікою. Найнижчу якість продемонструвала модель CNN: її точність становила приблизно 85 %, а EER був понад 15 %. Це пояснюється тим, що згорткові мережі добре працюють із просторовими ознаками, але не здатні достатньо точно враховувати часові залежності між натисканнями клавіш.

Архітектура LSTM показала вищі результати – близько 89 % точності та EER на рівні 10 %. Модель краще розпізнає послідовності, оскільки вміє запам’ятовувати попередні стани та відтворювати характерний ритм друку.

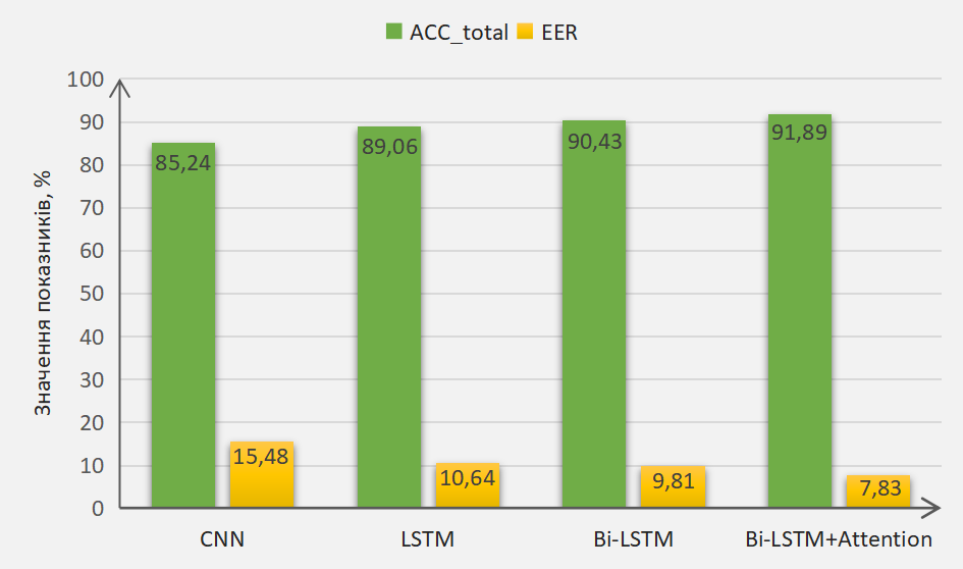

Рисунок 3.10 - Вплив різних нейронних архітектур на точність автентифікації

Ще кращі показники отримано для Bi-LSTM. Точність перевищила 90 %, а EER знизився до близько 9,8 %. Саме завдяки аналізу послідовності в обох напрямках ця архітектура формує повніший контекст і точніше відокремлює справжні послідовності від чужих.

Найвищу якість забезпечила модель Bi-LSTM з механізмом уваги. Вона досягла точності 91,89 % та найнижчого серед усіх моделей значення EER – 7,83 %. У цій моделя, механізм уваги відіграє важливу роль: він оцінює важливість кожного елемента поведінкової послідовності та надає моделі можливість зосереджуватися саме на тих фрагментах, які найбільше характеризують індивідуальний ритм друку. Таким чином, модель не лише опрацьовує часові зв’язки, а й активно «відфільтровує» менш інформативні або шумові ділянки послідовностей. Саме це дає змогу досягти найнижчого рівня критичних помилок автентифікації та забезпечує найвищу точність серед порівнюваних архітектур.

Таким чином, проведене порівняння підтверджує, що архітектура Bi-LSTM з механізмом уваги є найбільш ефективною для задачі автентифікації за клавіатурною динамікою, забезпечуючи оптимальний баланс між точністю та здатністю коректно розмежовувати автентичні й неавтентичні послідовності.

Висновки до розділу 3

У цьому розділі проведено комплексну експериментальну оцінку моделі автентифікації за клавіатурною динамікою. Попередня обробка даних, що включала очищення, дискретизацію та відбір користувачів, забезпечила формування коректних часових характеристик, які стали надійною основою для подальшого моделювання.

Результати експериментів показали, що довжина послідовності є одним із ключових параметрів моделі: оптимальним значенням виявилася довжина 30 подій, за якої досягнуто 92,69 % загальної точності та EER на рівні 7,83 %. Коротші послідовності не забезпечували достатнього контексту, тоді як довші призводили до зростання шуму та хибних прийнять.

Важливим чинником виявилися й параметри формування вибірки. Найкращі результати досягалися за використання розподілу 70:30 та кроку «ковзного вікна» 1, що забезпечило понад 92 % точності та мінімальні значення FRR і FAR. Збільшення кроку до 15 або 30 знижувало точність до 81–83 % та підвищувало EER понад 16 %.

Аналіз ROC-кривих показав залежність якості роботи моделі від індивідуальних особливостей користувачів: для більшості з них криві розташовані у верхній лівій частині графіка, що свідчить про високу ймовірність правильного прийняття за низького рівня помилок. Оцінка точності в часовій перспективі підтвердила стабільність моделі — для різних сесій точність залишалася в межах 89,6–90,8 %, а EER змінювався від 6,52 % до 5,9 %.

Порівняльний аналіз архітектур засвідчив, що найбільш ефективною є модель Bi-LSTM з механізмом уваги, яка забезпечує найвищу точність і найнижчий показник EER серед розглянутих підходів завдяки врахуванню повного часового контексту та виділенню інформативних елементів послідовності.

РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

Оскільки дана кваліфікаційна робота присвячена дослідженню безперервної автентифікації користувачів на основі динаміки натискання клавіш, реалізація та використання відповідної системи передбачає тривалу роботу з комп'ютерною технікою, зокрема персональними комп'ютерами, клавіатурою та іншими периферійними пристроями. У зв'язку з цим обов'язковим є дотримання вимог охорони праці, техніки безпеки та санітарно-гігієнічних норм, що регламентують умови роботи з екранними пристроями та електрообладнанням.

Організація безпечних умов праці для користувачів і фахівців, які працюють із програмно-апаратними засобами обробки даних, регламентується чинним законодавством України у сфері охорони праці. Відповідальність за створення безпечних умов праці та дотримання нормативно-правових актів покладається на керівника організації відповідно до Закону України «Про охорону праці».

Особливу увагу необхідно приділяти вимогам НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», які визначають загальні умови організації та експлуатації комп'ютерної техніки.

Приміщення, у яких розміщуються робочі місця користувачів персональних комп'ютерів, повинні відповідати вимогам пожежної безпеки та бути обладнані відповідними засобами пожежної сигналізації і пожежогасіння. Відповідно до вимог державних будівельних норм ДБН В.2.5-56:2014 та нормативних актів НАПБ А.01.001-2014, такі приміщення мають бути оснащені автоматичною пожежною сигналізацією та переносними вогнегасниками, а проходи до засобів пожежогасіння повинні залишатися вільними. Кількість і тип вогнегасників визначаються згідно з вимогами ДСТУ 4297:2004 з урахуванням площі приміщення та характеру встановленого обладнання.

Електроживлення комп’ютерної техніки та периферійних пристроїв повинно здійснюватися від окремої трипровідної мережі з обов’язковим використанням нульового захисного провідника для заземлення електроприймачів. Згідно з вимогами НПАОП 40.1-1.01-97, забороняється використовувати нульовий робочий провідник як захисний, а також підключати комп’ютери до двопровідної електромережі або через несправні перехідні пристрої. У приміщеннях, де одночасно експлуатується значна кількість комп’ютерів, має бути передбачений аварійний вимикач для повного знеструмлення приміщення, за винятком освітлення.

Важливим аспектом охорони праці під час роботи з комп’ютером є дотримання ергономічних і санітарно-гігієнічних вимог. Робоче місце повинно бути організоване відповідно до положень ДСТУ 8604:2015, які регламентують розташування елементів робочого місця для роботи в положенні сидячи. Розміщення монітора та клавіатури має забезпечувати зручну робочу позу користувача та мінімізувати зорове й м’язове навантаження під час тривалої роботи. Освітлення робочої зони повинно забезпечувати достатній рівень природного та штучного світла відповідно до вимог НПАОП 0.00-7.15-18, без виникнення відблисків на екрані.

З урахуванням специфіки даної кваліфікаційної роботи, особливого значення набуває дотримання режимів праці та відпочинку. Періодичні перерви, правильне положення рук і зап’ясть під час набору тексту, а також ергономічне розміщення пристроїв введення сприяють зменшенню втоми та запобігають виникненню професійних захворювань.

Таким чином, дотримання вимог охорони праці під час експлуатації комп’ютерної техніки є необхідною умовою безпечного та ефективного використання розробленої системи безперервної автентифікації за клавіатурною динамікою.

4.2 Фактори, що впливають на функціональний стан користувачів комп'ютерів

Функціональний стан користувачів персональних комп’ютерів формується під впливом комплексу факторів виробничого середовища, організації праці та індивідуальних психофізіологічних особливостей людини. У межах даної кваліфікаційної роботи ці фактори мають особливе значення, оскільки досліджувана система безперервної автентифікації ґрунтується на аналізі поведінкових характеристик користувача, які можуть змінюватися залежно від умов роботи та рівня працездатності.

Одним із основних чинників, що впливають на функціональний стан, є тривале зорове навантаження, пов’язане з постійною роботою за екраном монітора. Недотримання вимог до освітлення робочого місця, наявність відблисків або недостатній контраст зображення призводять до швидкого розвитку зорової втоми. Це може супроводжуватися зниженням концентрації уваги, погіршенням сприйняття інформації та загальним зменшенням ефективності роботи користувача.

Важливим фактором є також статичне м’язове навантаження, яке виникає внаслідок тривалого перебування в сидячому положенні та виконання одноманітних рухів руками під час роботи з клавіатурою. Неправильне розташування робочих елементів, зокрема клавіатури та монітора, сприяє перенапруженню м’язів спини, шиї та верхніх кінцівок. У результаті зростає втомлюваність, знижується точність рухів і погіршується загальний фізичний стан користувача.

Психоемоційні фактори відіграють не менш важливу роль у формуванні функціонального стану. Інтенсивна розумова діяльність, необхідність обробки великих обсягів інформації та монотонність роботи можуть спричиняти емоційне напруження і стрес. Такі умови призводять до зниження уваги, уповільнення реакцій і нестабільності поведінкових проявів, що безпосередньо відображається на характері взаємодії користувача з комп’ютером.

Суттєвий вплив на працездатність мають мікрокліматичні умови робочого приміщення, зокрема температура повітря, рівень вологості та якість вентиляції. Відхилення від оптимальних значень можуть викликати сонливість або, навпаки, підвищену дратівливість, що негативно позначається на функціональному стані користувача та його здатності тривалий час підтримувати стабільний рівень працездатності.

Окрему увагу слід приділити впливу шуму та інших фонових подразників. Постійний шум від обладнання або зовнішніх джерел знижує концентрацію уваги та сприяє швидшому розвитку втоми. У поєднанні з розумовим навантаженням це створює додаткове навантаження на нервову систему людини.

Важливим елементом забезпечення безпеки життєдіяльності є дотримання раціонального режиму праці та відпочинку. Безперервна робота за комп’ютером протягом тривалого часу призводить до накопичення втоми та зниження функціональних можливостей організму. Регламентовані перерви, зміна виду діяльності та короткочасний відпочинок сприяють відновленню працездатності та зменшенню негативного впливу шкідливих факторів.

З огляду на специфіку досліджуваної системи безперервної автентифікації, функціональний стан користувача має додаткове значення, оскільки саме він визначає стабільність поведінкових характеристик клавіатурного вводу. Зміни фізичного або психоемоційного стану можуть призводити до варіацій часових параметрів натискання клавіш, що необхідно враховувати під час проєктування та експлуатації таких систем у реальних умовах.

Отже, забезпечення безпеки життєдіяльності користувачів комп’ютерів передбачає комплексний підхід, який включає оптимальну організацію робочого місця, дотримання санітарно-гігієнічних норм, урахування психофізіологічних факторів та раціональний режим праці. Це дозволяє не лише зберігати здоров’я користувачів, а й забезпечувати стабільну та ефективну роботу інформаційних систем, зокрема систем безперервної автентифікації за клавіатурною динамікою.

Висновки до розділу 4

У розділі розглянуто основні аспекти охорони праці під час роботи з комп’ютерною технікою в контексті розробки та використання систем безперервної автентифікації за клавіатурною динамікою. Проаналізовано нормативні вимоги щодо організації безпечних умов праці, пожежної та електробезпеки, а також ергономічні й санітарно-гігієнічні вимоги до робочого місця користувача персонального комп’ютера. Показано, що дотримання зазначених вимог є необхідною умовою безпечної та ефективної експлуатації комп’ютерної техніки.

У межах розділу також досліджено фактори, що впливають на функціональний стан користувачів комп’ютерів, зокрема зорове, м’язове та психоемоційне навантаження, умови мікроклімату й режим праці та відпочинку. Обґрунтовано, що врахування психофізіологічних особливостей користувачів сприяє збереженню їхнього здоров’я та забезпечує стабільність поведінкових характеристик клавіатурного вводу, що є важливим для надійної роботи систем безперервної автентифікації в реальних умовах експлуатації.

ВИСНОВКИ

У кваліфікаційній роботі проведено комплексне дослідження методу безперервної автентифікації користувачів інформаційних систем на основі динаміки натискання клавіш, орієнтованого на аналіз коротких послідовностей клавіатурного вводу в неконтрольованих умовах. Досягнута мета роботи полягає у підтвердженні можливості забезпечення високої точності автентифікації за умов обмеженого обсягу вхідних даних та відсутності фіксованого тексту.

У ході виконання роботи було здійснено ґрунтовний аналіз сучасного стану проблеми автентифікації користувачів в інформаційних системах, а також досліджено існуючі підходи до біометричної та, зокрема, поведінкової автентифікації. Показано, що традиційні механізми підтвердження особи, засновані на паролях або апаратних носіях, не забезпечують належного рівня захисту в умовах сучасних кіберзагроз. У результаті аналізу встановлено, що методи динаміки натискання клавіш є одним із найбільш перспективних напрямів для побудови систем безперервної автентифікації завдяки їх неінвазивності, відсутності потреби в додатковому обладнанні та можливості фонової роботи.

У роботі досліджено основні принципи поведінкової біометрії та проаналізовано часові характеристики клавіатурного вводу, які використовуються для формування поведінкового профілю користувача. Для подальшого моделювання обґрунтовано вибір трьох базових ознак: назви клавіші, тривалості її утримання та інтервалу між послідовними натисканнями, які є фундаментальними для опису індивідуального стилю набору тексту.

На основі проведеного аналізу була розроблена та програмно реалізована модель безперервної автентифікації, що базується на рекурентній архітектурі Bi-LSTM у поєднанні з механізмом уваги. Також реалізовано повний алгоритм функціонування системи, який охоплює всі етапи обробки даних — від зчитування сирих клавіатурних подій і формування ознак до прийняття фінального рішення про автентичність користувача.

Експериментальні дослідження ефективності роботи моделі, проведені з використанням датасету Clarkson University Keystroke Dataset II, дозволили оцінити вплив ключових параметрів на якість автентифікації. Встановлено, що оптимальною є довжина послідовності у 30 подій, за якої модель досягла 92,69 % загальної точності та значення EER на рівні 7,83 %. Менші послідовності не забезпечували достатнього контексту, тоді як більші призводили до зростання шуму та збільшення кількості хибних прийнять.

Також показано, що параметри формування вибірки суттєво впливають на результати класифікації. Найкращу ефективність система демонструє за використання кроку «ковзного вікна» 1 та поділу даних у співвідношенні 70:30, що забезпечує оптимальний баланс між різноманітністю тренувальних прикладів і здатністю моделі до узагальнення. Збільшення кроку призводить до зниження точності та погіршення показників FAR, FRR і EER.

Аналіз ROC-кривих підтвердив високу роздільну здатність запропонованої моделі та її здатність стабільно відокремлювати класи genuine та impostor за різних порогових значень. Додаткове дослідження точності в часовій перспективі показало, що модель зберігає стійкість навіть за значних часових інтервалів між тренувальними та тестовими даними, що свідчить про довгострокову стабільність поведінкового профілю користувачів.

Порівняльний аналіз з альтернативними архітектурами, зокрема CNN, LSTM та класичною Bi-LSTM, показав, що використання бінапрямленої рекурентної мережі з механізмом уваги забезпечує приріст точності на 3–5 % та зменшення значення EER. Це підтверджує доцільність поєднання рекурентних архітектур із механізмами уваги для задач безперервної автентифікації за короткими послідовностями клавіатурного вводу.

Отримані результати дозволяють сформулювати практичні рекомендації щодо використання запропонованої моделі в інформаційних системах, що потребують підвищеного рівня безпеки, зокрема в умовах віддаленої роботи або спільного користування пристроями.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Allafi, R., & Darem, A. (2025). Usability and security in online authentication systems. International Journal of Advanced and Applied Sciences, 12(6), 1–12. https://doi.org/10.21833/ijaas.2025.06.001.

2. Amin, R., Gaber, T., Eltaweel, G., & Hassanien, A. E. (2014). Biometric and traditional mobile authentication techniques: Overviews and open issues. In Advanced biometric technologies (pp. 215–238). Springer. https://doi.org/10.1007/978-3-662-43616-5_16.

3. Baig, A. F., & Eskeland, S. (2021). Security, privacy, and usability in continuous authentication: A survey. Sensors, 21(17), Article 5967. https://doi.org/10.3390/s21175967.

4. Bhattacharyya, D., Ranjan, R., Alisherov, F., & Choi, M. (2009). Biometric authentication: A review. International Journal of u- and e-Service, Science and Technology, 2(3), 13–28.

5. Clarkson University. (2009). Clarkson University keystroke dataset II (Clarkson II) [Data set]. Department of Electrical and Computer Engineering, Clarkson University.

6. ElMenshawy, D., Mokhtar, H., & Hegazy, O. (2014). A keystroke dynamics based approach for continuous authentication. In Communications in computer and information science (Vol. 424, pp. 415–424). Springer. https://doi.org/10.1007/978-3-319-06932-6_40.

7. Fouad, K., Hassan, B., & Hassan, M. (2016). User authentication based on dynamic keystroke recognition. International Journal of Ambient Computing and Intelligence, 7(3), 1–32. https://doi.org/10.4018/IJACI.2016070101.

8. Furnell, S., & Haskell-Dowland, P. (2004). A long-term trial of keystroke profiling using digraph, trigraph and keyword latencies. In Security and protection in information processing systems (pp. 275–289). Springer. https://doi.org/10.1007/1-4020-8143-X_18.

9. Graves, A., Fernández, S., & Schmidhuber, J. (2005). Bidirectional LSTM networks for improved phoneme classification and recognition. In Artificial neural

networks: Formal models and their applications – ICANN 2005 (Lecture Notes in Computer Science, Vol. 3697, pp. 799–804). Springer. https://doi.org/10.1007/11550907_126.

10. Ihianle, I., Nwajana, A., Ebenuwa, S., Otuka, R., Owa, K., & Orisatoki, M. (2020). A deep learning approach for human activities recognition from multimodal sensing devices. IEEE Access, 8, 179028–179038. https://doi.org/10.1109/ACCESS.2020.3027979.

11. Karpinski, M., Khoma, V., Dudvkevych, V., Khoma, Y., & Sabodashko, D. (2018). Autoencoder neural networks for outlier correction in ECG-based biometric identification. In 2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS) (pp. 210–215). IEEE. https://doi.org/10.1109/IDAACS-SWS.2018.8525836.

12. Kiyani, A., Lasebae, A., Ali, K., Ur Rehman, M., & Haq, B. (2020). Continuous user authentication featuring keystroke dynamics based on robust recurrent confidence model and ensemble learning approach. IEEE Access, 8, 136975–136989. https://doi.org/10.1109/ACCESS.2020.3019467.

13. Lindemann, B., Müller, T., Vietz, H., Jazdi, N., & Weyrich, M. (2020). A survey on long short-term memory networks for time series prediction. arXiv Preprint. https://doi.org/10.13140/RG.2.2.36761.65129.

14. Lis, K., Niewiadomska-Szynkiewicz, E., & Dziewulska, K. (2023). Siamese neural network for keystroke dynamics-based authentication on partial passwords. Sensors, 23(15), Article 6685. https://doi.org/10.3390/s23156685.

15. Lu, X., Zhang, S., Hui, P., & Lio, P. (2020). Continuous authentication by free-text keystroke based on CNN and RNN. Computers & Security, 96, Article 101861. https://doi.org/10.1016/j.cose.2020.101861.

16. Lypa, B., Horyn, I., Zagorodna, N., Tymoshchuk, D., & Lechachenko, T. (2024). Comparison of feature extraction tools for network traffic data. CEUR Workshop Proceedings, 3896, 1–11.

17. Microsoft. (2025). What is authentication? https://www.microsoft.com/en-us/security/business/security-101/what-is-authentication. (дата звернення: 28.09.2025)

18. Microsoft. (2025). Security at your organization. https://learn.microsoft.com/en-us/partner-center/security/security-at-your-organization. (дата звернення: 28.09.2025)

19. Monrose, F., & Rubin, A. D. (1997). Authentication via keystroke dynamics. In Proceedings of the 4th ACM Conference on Computer and Communications Security (pp. 48–56). ACM. https://doi.org/10.1145/266420.266434.

20. Shmatko, O., Balakireva, S., Vlasov, A., et al. (2020). Development of methodological foundations for designing a classifier of threats to cyberphysical systems. Eastern-European Journal of Enterprise Technologies, 3(9), 6–19.

21. Skarga-Bandurova, I., Biloborodova, T., Skarha-Bandurov, I., Boltov, Y., & Derkach, M. (2021). A multilayer LSTM auto-encoder for fetal ECG anomaly detection. Studies in Health Technology and Informatics, 285, 147–152.

22. Sun, W., Chen, Y., Du, Q., Huang, Z., Rehman, Z., & Huang, L. (2024). Physics-guided deep learning-based constitutive modeling for the gravelly soil–structure interface. Scientific Reports. https://doi.org/10.21203/rs.3.rs-4596626/v1

23. TechTarget. (2025). Continuous authentication. https://www.techtarget.com/searchsecurity/definition/continuous-authentication.

24. Teh, P. S., Teoh, A. B., & Yue, S. (2013). A survey of keystroke dynamics biometrics. The Scientific World Journal, 2013, Article 408280. https://doi.org/10.1155/2013/408280.

25. Tymoshchuk, D., & Yatskiv, V. (2024). Slowloris DDoS detection and prevention in real time. In Collection of scientific papers “ΛΌГОΣ” (pp. 171–176).

26. Wang, X., Shi, Y., Zheng, K., Zhang, Y., Hong, W., & Cao, S. (2022). User authentication method based on keystroke dynamics and mouse dynamics with scene-irrelevant features in hybrid scenes. Sensors, 22(17), Article 6627. https://doi.org/10.3390/s22176627.

27. Yang, L., Li, C., You, R., et al. (2021). TKCA: A timely keystroke-based continuous user authentication with short keystroke sequence in uncontrolled settings. Cybersecurity, 4, Article 13. https://doi.org/10.1186/s42400-021-00075-9.

28. Zagorodna, N., Stadnyk, M., Lypa, B., Gavrylov, M., & Kozak, R. (2022). Network attack detection using machine learning methods. Challenges to National Defence in Contemporary Geopolitical Situation, 55–61.

29. Matiuk D., Skarga-Bandurova I., Derkach M. (2025) EMG pattern recognition for thumb muscle states using wearable sensing and adaptive neural network. Scientific Journal of TNTU (Tern.), vol 119, no 3, pp. 5–11.

30. Muzh, V., & Lechachenko, T. (2024). Computer technologies as an object and source of forensic knowledge: challenges and prospects of development. Вісник Тернопільського національного технічного університету, 115(3), 17-22.

31. Lupenko, S., Orobchuk, O., & Xu, M. (2019, September). Logical-structural models of verbal, formal and machine-interpreted knowledge representation in Integrative scientific medicine. In Conference on Computer Science and Information Technologies (pp. 139-153).

32. Derkach, M., Matiuk, D., Skarga-Bandurova, I., Biloborodova, T., & Zagorodna, N. (2024, October). A Robust Brain-Computer Interface for Reliable Cognitive State Classification and Device Control. In 2024 14th International Conference on Dependable Systems, Services and Technologies (DESSERT) (pp. 1-9). IEEE.

33. Skorenkyy, Y., Kozak, R., Zagorodna, N., Kramar, O., & Baran, I. (2021, March). Use of augmented reality-enabled prototyping of cyber-physical systems for improving cyber-security education. In Journal of Physics: Conference Series (Vol. 1840, No. 1, p. 012026). IOP Publishing.

34. Деркач М. В., Мишко О. Є. Використання алгоритму шифрування AES-256-CBC для зберігання даних автентифікації автономного помічника. Наукові вісті Далівського університету. 2023. №24.

Додаток А Публікація

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

М А Т Е Р І А Л И

XIII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

17-18 грудня 2025 року

ТЕРНОПІЛЬ
2025

Продовження додатку А

УДК 004.056.53
С. Літвінчук; Н. Загородна, к.т.н., доц.
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ДОСЛІДЖЕННЯ БЕЗПЕРЕРВНОЇ АВТЕНТИФІКАЦІЇ КОРИСТУВАЧІВ НА ОСНОВІ ДИНАМІКИ НАТИСКАННЯ КЛАВІШ

UDC 004.056.53
S. Litvinchuk; N. Zagorodna, Ph.D., Assoc. Prof.

RESEARCH OF CONTINUOUS USER AUTHENTICATION BASED ON KEYSTROKE DYNAMICS

У сучасних інформаційних системах традиційні процедури автентифікації здійснюються лише на етапі початкового доступу й не передбачають подальшої перевірки особи під час сеансу роботи, що створює можливість несанкціонованого використання активної сесії. За таких умов одним із найбільш обґрунтованих підходів до реалізації безперервної автентифікації є аналіз динаміки натискання клавіш.

У роботі розглянуто модель автентифікації на основін архітектури Bi-LSTM з механізмом уваги (рис. 1), що забезпечує врахування часових закономірностей і виділення найбільш інформативних фрагментів послідовності.

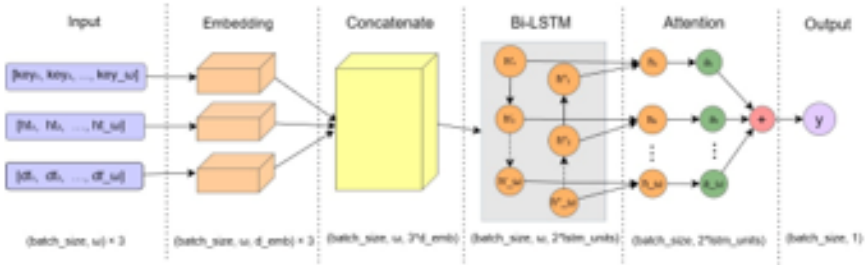

Рисунок 1. Архутектура досліджувальної моделі [1]

У якості вхідних даних використано датасет Clarkson University Keystroke Dataset II [2]. У межах дослідження обрано три базові параметри клавіатурних подій: тип клавіші, тривалість її утримання та інтервал між натисканнями. На базі цих характеристик формувалися послідовності для подальшого аналізу.

Було проведено серію експериментів щодо впливу довжини послідовності, кроку «ковзного вікна» та способу формування начальних та тестувальних вибірок на точність класифікації. Отримані результати показали, що найкращі показники досягаються для довжини послідовності 30 клавіатурних подій з розподілом даних 70:30 та кроком «ковзного вікна» 1, де модель продемонструвала загальну точність 92,69 % та EER 7,83 %, перевершивши базові архітектури CNN, LSTM та звичайну Bi-LSTM. Також доведено, що модель зберігає стійкість при тестуванні на віддалених у часі сесіях (1 - 2 роки).

Література

1. Yang, L., Li, C., You, R. et al. TKCA: A timely keystroke-based continuous user authentication with short keystroke sequence in uncontrolled settings. Cybersecurity 4, 13 (2021). https://doi.org/10.1186/s42400-021-00075-9.
2. Clarkson University. Clarkson University Keystroke Dataset II (Clarkson II). Provided under the Joint Multi-modal Biometric Dataset Release Agreement. Potsdam, NY: Clarkson University, Department of Electrical and Computer Engineering, 2009.

Додаток Б Лістинг файлу preprocessing.py

```
import os
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from collections import defaultdict
dataset_path = r"C:/Users/User/MasterProject/KeyBioAuth/dataset"
output_path =
r"C:/Users/User/MasterProject/KeyBioAuth/processed_data.csv"
HT_LIMIT = 2400000
DF_LIMIT = 6000000
RESOLUTION = 60000
MIN_EVENTS = 10000
MIN_FINAL_ROWS = 5000
TICKS_PER_MS = 10000
all_data = []
print(f"Зчитування файлів із папки: {dataset_path}")
for user_file in os.listdir(dataset_path):
    user_path = os.path.join(dataset_path, user_file)
    if not os.path.isfile(user_path):
        continue
    try:
        df = pd.read_csv(user_path,
sep="\t",header=None,names=["timestamp", "event_type",
"key_name"],on_bad_lines="skip")
        df["user_id"] = user_file
        all_data.append(df)
    except Exception as e:
        print(f"Помилка у файлі {user_file}: {e}")
data = pd.concat(all_data, ignore_index=True)
print(f"Усього рядків після зчитування: {len(data):,}")
before_len = len(data)
data = data.drop_duplicates(subset=["user_id", "timestamp",
"event_type", "key_name"])
print(f"Видалено {before_len - len(data):,} дублікатів (на сирих
timestamp)")
print(f"Рядків після видалення дублікатів: {len(data):,}")
user_counts = data["user_id"].value_counts()
valid_users = user_counts[user_counts >= MIN_EVENTS].index
data = data[data["user_id"].isin(valid_users)]
print(f"Користувачів з ≥{MIN_EVENTS} подіями: {len(valid_users)}")
print(f"Рядків після фільтрації: {len(data):,}")
data["timestamp"] = data["timestamp"].astype("int64")
data["timestamp"] = data["timestamp"].apply(lambda x: (x >> 4) <<
4)
all_actions = []
for user_id, group in data.groupby("user_id"):
    group = group.sort_values("timestamp").reset_index(drop=True)
    group["ht"] = np.nan
    stacks = defaultdict(list)
    for i, row in group.iterrows():
        key = row["key_name"]
        et = row["event_type"]
        ts = row["timestamp"]
```

Продовження додатку Б

```
        if et == 0:  # key down
            stacks[key].append(i)
        elif et == 1:  # key up
            if stacks[key]:
                j = stacks[key].pop()
                press_ts = group.at[j, "timestamp"]
                ht_ticks = ts - press_ts
                if ht_ticks > 0:
                    group.at[j, "ht"] = ht_ticks
    press_events = group[(group["event_type"] == 0) &
(group["ht"] > 0)].copy()
    if len(press_events) == 0:
        continue
    press_events =
press_events.sort_values("timestamp").reset_index(drop=True)
    press_events["next_key"] = press_events["key_name"].shift(-1)
    press_events["next_time"] = press_events["timestamp"].shift(-
1)
    press_events["df"] = press_events["next_time"] -
press_events["timestamp"]
    actions = press_events[
        (press_events["df"] > 0) &
        (~press_events["next_time"].isna()) &
        (press_events["key_name"] !=
press_events["next_key"])].copy()
    if len(actions) == 0:
        continue
    final = actions[["user_id", "timestamp", "key_name", "ht",
"df"]]
    all_actions.append(final)
final_data = pd.concat(all_actions, ignore_index=True)
print(f"Усього рядків після обчислення HT/DF:
{len(final_data):,}")
before_len = len(final_data)
final_data = final_data[(final_data["ht"] <= HT_LIMIT) &
(final_data["df"] <= DF_LIMIT)]
print(f"\nВидалено {before_len - len(final_data):,} аномальних
рядків (HT/DF за межами порогів)")
print(f"Залишилось після фільтрації аномалій:
{len(final_data):,}")
encoder = LabelEncoder()
final_data["key_id"] =
encoder.fit_transform(final_data["key_name"])
final_data["ht_id"] = (final_data["ht"] // RESOLUTION).astype(int)
final_data["df_id"] = (final_data["df"] // RESOLUTION).astype(int)
counts_per_user = final_data["user_id"].value_counts()
valid_final_users = counts_per_user[counts_per_user >=
MIN_FINAL_ROWS].index
filtered_data =
final_data[final_data["user_id"].isin(valid_final_users)].copy()
removed_users = set(final_data["user_id"].unique()) -
set(valid_final_users)
```

Продовження додатку Б

```
print(f"\nВидалено {len(removed_users)} користувачів з
<{MIN_FINAL_ROWS} фінальних рядків")
print(f"Залишилось користувачів: {len(valid_final_users)}")
print(f"Рядків після остаточної фільтрації:
{len(filtered_data):,}")
filtered_data = filtered_data.sort_values(["user_id",
"timestamp"]).reset_index(drop=True)
ht_ms_after = filtered_data["ht"] / TICKS_PER_MS
df_ms_after = filtered_data["df"] / TICKS_PER_MS
print("\nСтатистика HT/DF ")
print(f"HT медіана: {ht_ms_after.median():.1f} мс, середнє:
{ht_ms_after.mean():.1f} мс")
print(f"DF медіана: {df_ms_after.median():.1f} мс, середнє:
{df_ms_after.mean():.1f} мс")
filtered_data.to_csv(output_path, index=False)
print(f"\nФінальні дані збережено у {output_path}")
print(f"Усього користувачів у фінальних даних:
{filtered_data['user_id'].nunique()}")
print("\nПриклад фінальних даних:")
print(filtered_data.head(10))
```

Додаток В Лістинг файлу sequence_split.py

```
import os
import numpy as np
import pandas as pd
DATA_CSV_PATH =
r"C:/Users/User/MasterProject/KeyBioAuth/processed_data.csv"
BASE_DIR  =  r"C:/Users/User/MasterProject/KeyBioAuth/train_test"
SEED = 42
np.random.seed(SEED)
rng = np.random.default_rng(SEED)
SEQUENCE_LENGTH = 30
STEP = 1
def build_sequences_for_users(data, user_ids, step):
    sequences = []
    for user_id in user_ids:
        user_data = data[data["user_id"] ==
user_id].reset_index(drop=True)
        if len(user_data) < SEQUENCE_LENGTH:
            print(f"Користувач {user_id}: замало даних
({len(user_data)}) – пропуск.")
            continue
        for i in range(0, len(user_data) - SEQUENCE_LENGTH + 1,
step):
            seq = user_data.iloc[i:i + SEQUENCE_LENGTH]
            sequences.append({"user_id": user_id,
                "key_ids": seq["key_id"].tolist(),
                "ht_ids":  seq["ht_id"].tolist(),
                "df_ids":  seq["df_id"].tolist(),})
    return sequences
def save_per_user(dataset_dict, base_dir):
    os.makedirs(base_dir, exist_ok=True)
    for user_id, d in dataset_dict.items():
        df = pd.DataFrame({
            "key_ids": [list(x) for x in d["X_keys"]],
            "ht_ids":  [list(x) for x in d["X_hts"]],
            "df_ids":  [list(x) for x in d["X_dfs"]],
            "y":       d["y"],"source":  d["source"]})
        df.to_csv(os.path.join(base_dir, f"{user_id}.csv"),
index=False)
data = pd.read_csv(DATA_CSV_PATH)
data = data.sort_values(["user_id",
"timestamp"]).reset_index(drop=True)
user_ids = data["user_id"].unique()
sequences = build_sequences_for_users(data, user_ids, step=STEP)
X_keys = np.array([s["key_ids"] for s in sequences])
X_hts  = np.array([s["ht_ids"]  for s in sequences])
X_dfs  = np.array([s["df_ids"]  for s in sequences])
y_user = np.array([s["user_id"] for s in sequences])
train_data = {}
test_data  = {}
for user in user_ids:
    print(f"\nКористувач {user}")
    pos_idx = np.where(y_user == user)[0]
    if len(pos_idx) < 20:
```

Продовження додатку В

```
        print("Замало genuine послідовностей. Пропуск.")
        continue
    pos_idx = np.sort(pos_idx)
    split = int(len(pos_idx) * 0.7)
    pos_train = pos_idx[:split]
    pos_test  = pos_idx[split:]
    other_users = [u for u in user_ids if u != user]
    rng.shuffle(other_users)
    half = len(other_users) // 2
    internal_users = other_users[:half]
    external_users = other_users[half:]
    neg_internal_idx = np.where(np.isin(y_user,
internal_users))[0]
    neg_external_idx = np.where(np.isin(y_user,
external_users))[0]
    if len(neg_internal_idx) == 0 or len(neg_external_idx) == 0:
        print("Недостатньо impostor. Пропуск.")continue
    rng.shuffle(neg_internal_idx)
    rng.shuffle(neg_external_idx)
    train_size = min(len(pos_train), len(neg_internal_idx))
    pos_train = rng.choice(pos_train, train_size, replace=False)
    neg_train = neg_internal_idx[:train_size]
    remaining_internal = neg_internal_idx[train_size:]
    test_size = min(len(pos_test), len(remaining_internal),
len(neg_external_idx))
    if test_size == 0:
        print("Недостатньо тестових impostor. Пропуск.")continue
    pos_test = rng.choice(pos_test, test_size, replace=False)
    neg_test_internal = remaining_internal[:test_size]
    neg_test_external = neg_external_idx[:test_size]
    train_data[user] = {
        "X_keys": np.vstack([X_keys[pos_train],
X_keys[neg_train]]),"X_hts":  np.vstack([X_hts[pos_train],  X_hts[
neg_train]]),
        "X_dfs":  np.vstack([X_dfs[pos_train],  X_dfs[neg_train]])
,"y": np.concatenate([np.zeros(train_size), np.ones(train_size)]),
        "source": ["genuine"] * train_size + ["internal"] *
train_size}
    test_data[user] = {"X_keys": np.vstack([X_keys[pos_test],
X_keys[neg_test_internal], X_keys[neg_test_external]]),
        "X_hts":  np.vstack([X_hts[pos_test],  X_hts[neg_test_inte
rnal], X_hts[neg_test_external]]),
        "X_dfs":  np.vstack([X_dfs[pos_test],  X_dfs[neg_test_inte
rnal], X_dfs[neg_test_external]]),
        "y":
np.concatenate([np.zeros(test_size),np.ones(test_size),np.ones(tes
t_size)]),"source": (["genuine"] * test_size +["internal"] *
test_size +["external"] * test_size)}
out_dir = os.path.join(BASE_DIR, f"w{SEQUENCE_LENGTH}_step1")
train_dir = os.path.join(out_dir, "train")
test_dir  = os.path.join(out_dir, "test")
save_per_user(train_data, train_dir)
save_per_user(test_data, test_dir)
```

Додаток Г Лістинг файлу model_train_test.py

```
import os
import numpy as np
import pandas as pd
import ast
import tensorflow as tf
from tensorflow.keras.models import Model
from tensorflow.keras.layers import (Input, Embedding, LSTM,
Dense, Concatenate, Dropout, Bidirectional, Attention)
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.metrics import accuracy_score, roc_curve
BASE_DIR = r"C:/Users/User/MasterProject/KeyBioAuth/train_test"
DATASET_DIR = os.path.join(BASE_DIR, "w30_step1")
TRAIN_DIR = os.path.join(DATASET_DIR, "train")
TEST_DIR  = os.path.join(DATASET_DIR, "test")
SEQUENCE_LENGTH = 30
EMBEDDING_DIM = 128
LSTM_UNITS = 256
EPOCHS = 15
BATCH_SIZE = 32
LEARNING_RATE = 0.001
TARGET_USERS = None
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
tf.config.threading.set_intra_op_parallelism_threads(4)
tf.config.threading.set_inter_op_parallelism_threads(4)
print("TensorFlow:", tf.__version__)
def build_model(vocab_key, vocab_ht, vocab_df,
seq_len=SEQUENCE_LENGTH):
    inp_keys = Input(shape=(seq_len,), name="keys")
    inp_hts  = Input(shape=(seq_len,), name="hts")
    inp_dfs  = Input(shape=(seq_len,), name="dfs")
    emb_keys = Embedding(vocab_key, EMBEDDING_DIM)(inp_keys)
    emb_hts  = Embedding(vocab_ht,  EMBEDDING_DIM)(inp_hts)
    emb_dfs  = Embedding(vocab_df,  EMBEDDING_DIM)(inp_dfs)
    merged = Concatenate(axis=-1)([emb_keys, emb_hts, emb_dfs])
    x = Bidirectional(LSTM(LSTM_UNITS, dropout=0.5,
return_sequences=True))(merged)
    x = Attention()([x, x])
    x = tf.keras.layers.GlobalAveragePooling1D()(x)
    x = Dropout(0.4)(x)
    out = Dense(1, activation="sigmoid")(x)
    model = Model(inputs=[inp_keys, inp_hts, inp_dfs],
outputs=out)
    model.compile(optimizer=Adam(LEARNING_RATE),loss="binary_cross
entropy",metrics=["accuracy"])
    return model
def safe_parse_list(s):
    if pd.isna(s): return []
    try: return list(ast.literal_eval(s))
    except: return [int(x) for x in str(s).split()]
def to_fixed_array(list_of_lists, seq_len=SEQUENCE_LENGTH):
    return np.array([
```

Продовження додатку Г

```
lst[:seq_len] + [0]*(seq_len - len(lst)) if len(lst) < seq_len
else lst[:seq_len]
        for lst in list_of_lists
    ], dtype=np.int32)
def load_user_file(path):
    df = pd.read_csv(path)
    Xk = to_fixed_array([safe_parse_list(x) for x in
df["key_ids"]])
    Xh = to_fixed_array([safe_parse_list(x) for x in
df["ht_ids"]])
    Xd = to_fixed_array([safe_parse_list(x) for x in
df["df_ids"]])
    y  = df["y"].astype(int).values
    src = df["source"].astype(str).values
    return Xk, Xh, Xd, y, src
train_files = [f for f in os.listdir(TRAIN_DIR) if
f.endswith(".csv")]
users = [os.path.splitext(f)[0] for f in train_files]
print("\nUsers to process:", users, "\n")
for USER_ID in users:
    print(f"TRAINING & TESTING USER {USER_ID}")
    train_path = os.path.join(TRAIN_DIR, f"{USER_ID}.csv")
    test_path  = os.path.join(TEST_DIR,  f"{USER_ID}.csv")
    Xk_train, Xh_train, Xd_train, y_train, src_train =
load_user_file(train_path)
    Xk_test,  Xh_test,  Xd_test,  y_test,  src_test  =
load_user_file(test_path)
    vocab_key = int(max(np.max(Xk_train), np.max(Xk_test))) + 1
    vocab_ht  = int(max(np.max(Xh_train), np.max(Xh_test))) + 1
    vocab_df  = int(max(np.max(Xd_train), np.max(Xd_test))) + 1
    model = build_model(vocab_key, vocab_ht, vocab_df)
    early_stop = EarlyStopping(monitor="val_loss", patience=3,
restore_best_weights=True)
    print("\nTraining model")
    history = model.fit([Xk_train, Xh_train, Xd_train],
y_train,epochs=EPOCHS,batch_size=BATCH_SIZE,validation_split=0.1,c
allbacks=[early_stop],verbose=1)
    print("Training completed\n")
    mask_g = src_test == "genuine"
    mask_i = src_test == "internal"
    mask_e = src_test == "external"
    def predict_block(Xk, Xh, Xd):
        if len(Xk) == 0:
            return np.array([])
        return model.predict([Xk, Xh, Xd], verbose=0).flatten()
    pg = predict_block(Xk_test[mask_g], Xh_test[mask_g],
Xd_test[mask_g])
    pi = predict_block(Xk_test[mask_i], Xh_test[mask_i],
Xd_test[mask_i])
    pe = predict_block(Xk_test[mask_e], Xh_test[mask_e],
Xd_test[mask_e])
    acc_genuine = accuracy_score(np.zeros(len(pg)), (pg >=
0.5).astype(int)) if len(pg) else np.nan
```

Продовження додатку Г

```
acc_total = accuracy_score(np.concatenate([np.zeros(len(pg)),
np.ones(len(pi)+len(pe))]),
        np.concatenate([(pg >= 0.5).astype(int),
(np.concatenate([pi,pe]) >= 0.5).astype(int)])
    ) if len(pg)+len(pi)+len(pe) else np.nan
    FRR = np.mean(pg >= 0.5) if len(pg) else np.nan
    iFAR = np.mean(pi < 0.5) if len(pi) else np.nan
    eFAR = np.mean(pe < 0.5) if len(pe) else np.nan
    FAR = np.nanmean([iFAR, eFAR])
    all_scores = np.concatenate([pg, pi, pe])
    all_labels = np.concatenate([np.zeros(len(pg)),
np.ones(len(pi)), np.ones(len(pe))])
    if len(np.unique(all_labels)) < 2: eer = np.nan
    else:
        fpr, tpr, thr = roc_curve(all_labels, all_scores)
        fnr = 1 - tpr
        idx = np.argmin(np.abs(fpr - fnr))
        eer = (fpr[idx] + fnr[idx]) / 2
    fmt = lambda x: f"{x:.4f}" if not np.isnan(x) else "nan"
    print("\nResults for single user")
    print(f"User id: {USER_ID}")
    print(f"Genuine: {len(pg)}, Internal impostors: {len(pi)},
External impostors: {len(pe)}")
    print(f"ACC_genuine = {fmt(acc_genuine)}")
    print(f"ACC_total = {fmt(acc_total)}")
    print(f"FRR = {fmt(FRR)}")
    print(f"iFAR = {fmt(iFAR)}")
    print(f"eFAR = {fmt(eFAR)}")
    print(f"FAR = {fmt(FAR)}")
    print(f"EER  = {fmt(eer)}\n")
    model.save(os.path.join(TRAIN_DIR, f"model_{USER_ID}.h5"))
```