Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ПОЯСНЮВАЛЬНА ЗАПИСКА
Дипломної роботи

магістра
(назва освітньо-кваліфікаційного рівня)

галузь знань *12 Інформаційні технології*
(шифр і назва галузі знань)

напрям підготовки *125 Кібербезпека*
(код і назва напряму підготовки)

освітній рівень ***магістр***

кваліфікація _____
(назва освітнього рівня)

на тему: Програмне забезпечення системи ідентифікації користувачів інформаційних систем по клавіатурному почерку

Виконавець: студент II **курсу, групи** КБМ-21

_____ Кузьменко Олександр Олександрович
(підпис) (прізвище ім'я по-батькові)

	Прізвище, ініціали	**Оцінка**	**Підпис**
Науковий керівник	Наконечний В.С.		
Рецензент			
Нормоконтроль			

Київ
2021

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
завідувач кафедри
кібербезпеки та захисту інформації
_____ Лукова-Чуйко Н.В.
«_____» _____ 2021 року

ЗАВДАННЯ
на виконання дипломної роботи

спеціальності *125 Кібербезпека*
(код і назва напряму підготовки)

студенту КБМ-21 (група) Кузьменка Олександра Олександровича (прізвище ім'я по-батькові)

Тема дипломної роботи Програмне забезпечення системи ідентифікації користувачів інформаційних систем по клавіатурному почерку

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Рішення засідання кафедри кібербезпеки та захисту інформації факультету інформаційних технологій протокол №2 від 08.10.2020 р.

2. МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень процес автентифікації користувача інформаційної системи у веб-середовищі по клавіатурному почерку.

Предмет досліджень клавіатурний почерк, як спосіб автентифікації користувача в інформаційній системі.

Мета розробка ефективного програмного забезпечення для автентифікації користувача в інформаційній системі за допомогою клавіатурного почерку

Вихідні дані для проведення роботи клавіатурний почерк, як біометрична ознака людини, веб-застосунок.

3. ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна використання клавіатурного почерку, як способу автентифікації людини в інформаційній системі, яка знаходиться у веб-середовищі

Практична цінність клавіатурний почерк ефективний механізм автентифікації користувачів в інформаційних системах з мінімальними вимогами до впровадження та супроводу

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Практичне доведення ефективності клавіатурного почерку за допомогою розробленого програмного забезпечення

5. ЕТАПИ ВИКОНАННЯ РОБОТИ

Найменування етапів робіт	Строки виконання робіт (початок-кінець)
Уточнення постановки задачі	12.10.2020 – 16.10.2020
Пошук та аналіз літератури	17.10.2020 – 10.12.2020
Збір даних	11.12.2021 – 12.01.2021
Обґрунтування вибору рішення	13.01.2021 – 26.02.2021
Розробка програмного забезпечення	27.02.2021 – 16.04.2021

Найменування етапів робіт	Строки виконання робіт (початок-кінець)
Проведення аналізу отриманих результатів	17.04.2021 – 07.05.2021
Робота над висновками	08.05.2021 – 12.05.2021
Оформлення презентації	13.05.2021 – 16.05.2021

6. РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект мінімальні витрати на впровадження, використання та супроводження системи автентифікації за клавіатурним почерком

Соціальний ефект Впровадження результатів роботи дозволить ефективно проводити автентифікацію користувача в інформаційній системі за допомогою клавіатурного почерку.

7. ДОДАТКОВІ ВИМОГИ

Завдання видав _____ (підпис) Наконечний В.С. (ініціали, прізвище)

Завдання прийняв до виконання _____ (підпис) Кузьменко О.О. (ініціали, прізвище)

Дата видачі завдання:

Термін подання дипломної роботи до ЕК _____

РЕФЕРАТ

Пояснювальна записка: 82с., 24 рис., 9 табл., 6 додатків, 28 джерел, 18 формул.

Об’єкт дослідження: процес автентифікації користувача інформаційної системи у веб-середовищі по клавіатурному почерку.

Мета роботи: розробка програмного забезпечення для автентифікації користувача в інформаційній системі за допомогою клавіатурного почерку та доведення ефективності використання даної біометричної ознаки.

Методи дослідження: математичні методи аналізу даних (ММАД), методи математичної статистики (ММС), відстань Евкліда.

В основній частині дана характеристика клавіатурному почерку, опис програмного забезпечення, яке використовується для навчання системи та розпізнавання клавіатурного почерку.

У роботі досліджено клавіатурний почерк як біометрична ознака людини для автентифікації в інформаційних системах. Проведено аналіз переваг та недоліків використання клавіатурного почерку, а також ММАД. Запропоновано використовувати автентифікацію клавіатурного почерку як додатковий захід безпеки. Побудовані алгоритми, за допомогою яких ПЗ здатне запам’ятовувати та розпізнавати варіанти клавіатурних почерків різних користувачів. Розроблено ПЗ, яке має два режими роботи: навчання та розпізнавання.

Практичне значення роботи полягає у доведенні ефективності клавіатурного почерку в інформаційних системах як інструмент автентифікації особи та попередження несанкціонованого доступу.

Результати здійснених у дипломній роботі досліджень можуть бути використані фахівцями з розробки та впровадження систем захисту інформації.

Наукова новизна полягає у застосуванні клавіатурного почерку, як способу автентифікації користувачів в ІС, яке може функціонувати у веб-середовищі і забезпечувати додатковий механізм безпеки від несанкціонованого доступу до інформаційної системи.

Напрямки подальших досліджень пошук шляхів для вдосконалення розробленого алгоритму, подальше впровадження розробленої системи автентифікації та оцінка якості стану інформаційної безпеки до та після її впровадження.

Ключові слова: автентифікація користувача, аналіз даних, біометричні ознаки людини, клавіатурний почерк, веб-застосунок, несанкціонований доступ.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

БД – база даних

ІС – інформаційна система

ІТ – інформаційні технології

ІТС – інформаційно-телекомунікаційні системи

ММС – математичні методи статистики

ММАД – математичні методи аналізу даних

НСД – Несанкціонований доступ

ПЗ – програмне забезпечення

СКУД – системи контролю та управління доступом

ЧУК – час утримання клавіші

ШНМ – штучна нейронна мережа

ВСТУП

На даний час захист інформації від несанкціонованого доступу грає дуже важливу роль і є необхідним заходом запобігання матеріального та нематеріального збитку її власника.

Щоб попередити небажаний доступ до секретної або конфіденційної інформації, дуже важливо брати у розрахунок ефективність роботи систему захисту даних та управління доступом.

Сучасні методи автентифікації користувачів поділяють на три основні групи [1]:

- парольні – засновані на використанні унікальної інформації (пароль, кодова фраза тощо);
- атрибутні – засновані на використанні унікального предмету (використання токену, електронної карти, апаратного ключа тощо);
- біометричні – засновані на використанні унікальності біологічних, психологічних та фізіологічних даних користувача (відбиток пальця, почерк, голос, райдужка ока тощо).

Стрімкий розвиток інформаційних технологій (ІТ) відправляє парольні та атрибутні способи ідентифікації на звалище минулого.

Центральною проблемою парольних та атрибутних способів є неточність ідентифікації користувача у системі та велика ймовірність щодо порушення її безпеки в результаті крадіжки, імітації певного атрибуту або злому пароля.

Відсутність функціоналу для виявлення підміни авторизованого легітимного користувача також виступає значним мінусом, тобто порушник може незаконно потрапити у систему в момент, коли законний користувач залишає її без контролю після етапу проходження авторизації.

Біометричні характеристики користувача, зокрема клавіатурний почерк, як спосіб ідентифікації, можуть гарантувати:

- високий рівень безпеки;
- неможливості відмови від авторства;
- комфорт для користувачів, враховуючи невід'ємність біометричних даних від певної людини.

Неперервний прихований моніторинг або введення кодового слова дає можливість своєчасно виявити відсутність законного користувача та перекрити доступ до системи для зловмисника.

Тому на сьогоднішній день стає вкрай важливим питання вивчення моделей, способів й алгоритмів визначення клавіатурного почерку користувачів інформаційних систем (ІС).

З вивченням клавіатурного почерку користувача були визначені головні його властивості [2]:

- швидкість вводу – співвідношення кількості введених символів до часу на їх друкування;
- динаміка вводу – показники інтервалів між натисканням клавіш та тривалості їх натиску;
- помилки при введенні тексту та частота їх появи;
- статистика використання певних клавіш;
- сила, з якою користувач натискає на клавіші.

Варто зауважити, що клавіатурний почерк – це нестатична біометрична характеристика людини і вона може змінюватися залежно від психологічного та фізіологічного стану користувача.

Тому існуючі програмні реалізації визначення клавіатурного почерку мають низький рівень ефективності та точності ідентифікації, а також велику вірогідність утворення похибок першого та другого роду.

Для математичного дослідження часу утримання клавіш (ЧУК), який притаманний для клавіатурного почерку користувача, з'являється необхідність у використанні методів математичної статистики (ММС).

В ході аналізу тривалості утримання клавіш користувачами з’явилась можливість визначити клавіатурний почерк людини та виявляти підміну законного користувача.

В ході роботи було розроблені та реалізовані алгоритми навчання системи та безпосереднього розпізнавання почерку користувача.

Метою роботи є реалізація механізму ідентифікації по клавіатурному почерку за допомогою ПЗ, що здійснює збір, аналіз та використання аналітичної моделі для порівняння отриманого шаблону при реєстрації з поточним шаблоном.

Основні задачі роботи:

- проаналізувати існуючі методи аналізу даних і обрати найбільш підходящі, виходячи зі складності реалізації і точності отриманого результату;
- розробити два алгоритми навчання системи і розпізнавання клавіатурного почерку;
- обрати відповідні технології для реалізації попередньо розроблених алгоритмів;
- побудувати архітектуру веб-застосунка та реалізувати її у вигляді програмного забезпечення (ПЗ);
- протестувати ПЗ та проаналізувати його ефективність;
- надати практичні рекомендації при використанні даних клавіатурного почерку.

Об'єкт дослідження – процес автентифікації користувача інформаційної системи у веб-середовищі по клавіатурному почерку.

Предмет дослідження – клавіатурний почерк, як метод автентифікації користувача.

Актуальність роботи визначається тим, що традиційні методи ідентифікації і автентифікації, засновані на використанні переносних ідентифікаторів, а також паролів і кодів доступу, мають ряд суттєвих недоліків, пов'язаних з тим, що для встановлення автентичності користувача застосовуються атрибутивні і засновані на

знаннях розпізнавальні характеристики, які можна підробити або вкрасти. Також варто додати, що зі стрімким розвитком веб-технологій та хмарних сховищ даних, постає питання використання надійних методів біометричної ідентифікації користувачів у веб-середовищі. З цього виникає питання, яким чином інтегрувати автентифікацію користувача за клавіатурним почерком у системі, яка знаходиться на віддалених серверах та базах даних (хмарах).

У зв'язку з цим зростає інтерес до використання біометричних методів ідентифікації у веб-технологіях без використання дорогого обладнання та встановлення додаткового ПЗ.

Веб-технології стрімко розвиваються, тому важливо використовувати біометричні характеристики людини, які є її невід'ємною частиною, тому що їх неможливо підробити, забути або втратити.

РОЗДІЛ 1
ЗАГАЛЬНІ ПРИНЦИПИ БІОМЕТРИЧНОЇ ІДЕНТИФІКАЦІЇ

1.1 Загальні відомості про види ідентифікації користувача

Зі стрімким розвитком комп'ютерних технологій все частіше виникає проблема захисту інформації в інформаційно-телекомунікаційних системах (ІТС). Тому на сьогодні актуальними є теоретичні аспекти в області захисту інформації за допомогою біометричних систем та їх практичне застосування безпосередньо в конкретних ІТС у веб-середовищі.

Управління доступом є одним із найголовніших методів захисту інформації, який регулює та контролює доступ до інформаційних ресурсів системи.

Методи захисту інформації, що спираються на управління доступом, включають наступні функції захисту інформації [1]:

- ідентифікація користувачів, ресурсів і персоналу системи інформаційної безпеки;
- ідентифікація і встановлення достовірності користувача за присвоєним логіном та паролем.

За останні роки все більше увага науковців та розробників у галузі інформаційної безпеки зростає до біометричної ідентифікації особи користувача у веб-середовищі. Найчастіше розглядаються вдосконалення найпопулярніших способів біометричної ідентифікації за відбитком пальцю, сітківкою ока тощо.

На думку автора цієї роботи з швидкою міграцією ІТС до веб-середовища, варто забезпечити прості, надійні та швидкі рішення, які забезпечать найбільший ступінь захисту у всесвітній мережі без придбання та використання дорогого апаратного забезпечення або розроблення та встановлення дорогого ПЗ.

Тому варто сказати, що існують інші способи ідентифікації користувача, які варті уваги [1].

Перед тим, як розпочати аналіз ефективності потрібно навести види автентифікації користувача в ІТС [1]:

- парольна - використання унікального паролю для входу у систему (цей пароль може бути вигаданий користувачем або згенерований спеціальною програмою та наданий користувачеві);
- атрибутна – використання унікального фізичного предмету (апаратне забезпечення) для входу у систему (токен, електронно-цифровий підпис тощо);
- біометрична – використання фізіологічних ознак для входу у систему (відбиток пальцю, сканування сітківки ока, геометрія руки тощо).

Варто розпочати з того, що парольні системи контролю та управління доступом (СКУД) найбільше використовуються засобами захисту інформації в ІТС. Їх популярність пояснюється тим, що використання даної системи є значно простішою в порівнянні з іншими, але подібні системи мають невисокий рівень безпеки, в зв'язку з наявністю великої кількісті недоліків [2]:

- можливість підбору пароля;
- невиконання інструкцій по створенню безпечного пароля користувачем (недбале ставлення до процедури вибору пароля);
- існування і наявність у вільному доступі спеціалізованих додатків для підбору і злому паролів;
- отримання паролю шляхом застосування насильства до користувача;
- викрадення або перехоплення при введенні власником;

Згідно даних дослідницької компанії RSA, яка провела опитування серед користувачів про місце зберігання паролю на випадок його забуття і опублікувала наступні результати [3]:

- 27% користувачів зберігають паролі у вигляді звичайного тексту на робочій машині;
- 25% користувачів використовують мобільний телефон для зберігання паролів;

- 48% користувачів записують паролі на листках паперу та зберігають в гаманцях/записниках/книгах тощо.

Веб-ресурс NordPass у 2020 році провів дослідження і опублікував список найбільш популярних паролів, які були використані при спробі злому, викриті тощо рис. 1.1 [4]:

Position	Password	Number of users	Time to crack it	Times exposed
1. ↑ (2)	123456	2,543,285	Less than a second	23,597,311
2. ↑ (3)	123456789	961,435	Less than a second	7,870,694
3. (new)	picture1	371,612	3 Hours	11,190
4. ↑ (5)	password	360,467	Less than a second	3,759,315
5. ↑ (6)	12345678	322,187	Less than a second	2,944,615
6. ↑ (17)	111111	230,507	Less than a second	3,124,368
7. ↑ (18)	123123	189,327	Less than a second	2,238,694
8. ↓ (1)	12345	188,268	Less than a second	2,389,787
9. ↑ (11)	1234567890	171,724	Less than a second	2,264,884
10. (new)	senha	167,728	10 Seconds	8,213

Рисунок. 1.1 - Рейтинг найпопулярніших паролів у 2020-му році

Беручи до уваги результати та звіти згаданих досліджень можна зробити висновок, що користувачі не мають знань, яким чином створювати потужні паролі, щоб їх було легко запам’ятати (щоб не шукати іншого місця для збереження на випадок забуття), але важко підібрати.

Атрибутний метод аутентифікації за допомогою унікального апаратного пристрою дозволяє забезпечити більш надійний захист інформації, ніж парольний, але атрибутна автентифікація, як з «пасивними», так і з «активними» унікальними предметами володіє наступними недоліками [5]:

- можливість крадіжки апаратного ідентифікатора у користувача;
- необхідність у спеціальному обладнанні для роботи з магнітними картками, смарт-картами та іншими;

- можливість виготовлення копії унікального предмета;
- можливість підробки унікального предмета.

Отже, атрибутний метод забезпечує більш надійний спосіб для автентифікації користувача, але з іншого боку виникає проблема захисту апаратного пристрою від втрат, викрадення тощо.

Біометричні засоби забезпечують високий ступінь захисту серед парольних та атрибутних методів, тому що відбитки пальців, які є унікальними і неповторними характеристиками для кожної людини, таким чином забезпечується максимальний рівень безпеки. Втрата таких ідентифікаторів, викрадення або підробка мінімальна.

Незважаючи на величезні переваги біометрії, існують її певні недоліки. Наприклад, система відбитків пальців менш зручна, якщо користувачі системи контролю доступу часто змінюються, наприклад в офісах бізнес-центрів. Витрати на обслуговування є низькими, але вартість встановлення спеціального ПЗ є високою в порівнянні з іншими системами контролю доступу. Така ж сама ситуація зі встановленням програмно-апаратного забезпечення для інших фізіологічних ознак, наприклад геометрії руки, ідентифікації за сітківкою ока тощо.

В даному підрозділі були описані та проаналізовані основні методи та способи автентифікації користувача за допомогою різних способів, а саме парольного, атрибутного та біометричного. Проаналізовані переваги й недоліки кожного способу автентифікації та наведені їх короткі характеристики.

1.2 Загальні відомості про види біометричної автентифікації користувача, їх переваги та недоліки

Більш детально розглянемо види біометричних засобів автентифікації, та визначимо, які механізми використовуються для цього.

Статичні ознаки – це ознаки, які практично не змінюються з плином часу, починаючи з народження людини, тобто це її фізіологічні характеристики [6]:

- відбиток пальця - за допомогою сканера одержують зображення відбитку, потім це зображення за складним алгоритмом перетворюється на спеціальний цифровий код, який далі порівнюється з еталонними кодами, що зберігаються в базі даних;
- розташування вен на долоні - прилад, який зчитує інформацію в цьому випадку, є інфрачервона камера. В результаті на вході програми при формуванні цифрового коду з'являється малюнок вен на руці людини. Не потребує контакту людини з пристроєм для сканування. Має високі показники надійності і достовірності;
- сітківка ока - в цьому випадку сканується малюнок кровоносних судин очного дна, який має нерухому структуру, незмінну в часі. За допомогою ПЗ із зображення виділяється малюнок потрібної райдужної оболонки. Цей метод є одним з найбільш точних серед біометричних методів;
- райдужна оболонка ока - малюнок райдужної оболонки ока - унікальний для кожної людини. За допомогою ПЗ із зображення виділяється малюнок потрібної райдужної оболонки. Цей метод є одним з найбільш точних серед біометричних методів;
- форма кисті - ідентифікація ґрунтується на розпізнаванні геометричних особливостей кисті руки. Спеціальний сканер формує тривимірний малюнок кисті. При аналізі цього малюнка виконуються вимірювання, за допомогою яких формується відповідний цифровий код;
- форма обличчя - двовимірне розпізнавання обличчя на сьогодні - один із самих неефективних методів біометрії, тому має обмежене коло застосування або використовується тільки в сукупності з іншими методами.

Динамічні ознаки – це поведінкові характеристики, які побудовані на підсвідомих особливостях рухів у процесі відтворення будь-якої дії [6]:

- голос - враховуються унікальні частотні характеристики голосу людини;
- почерк - досліджується почерк людини.

Перевіряються наступні динамічні характеристики, на яких будується спеціальний цифровий код: графічні параметри, сила натиску на поверхню, швидкість написання.

Клавіатурний почерк - метод аналогічний ідентифікації за почерком. Замість того, щоб ставити автограф, людині необхідно надрукувати кодове слово. Цифровий код будується по динаміці набору певного слова або фрази.

В даному підрозділі було надано інформацію щодо біометричних способів автентифікації користувача з використанням статичних та біометричних ознак людини: за відбитком пальцю, геометрією руки, обличчя, клавіатурного почерку тощо, а також наведені короткі характеристики процесів автентифікації за кожною ознакою.

1.3 Аналіз сучасних біометричних систем контролю доступу

Проаналізуємо найбільш розповсюдженні біометричні системи контролю доступу засновані на розпізнаванні фізіологічних і поведінкових характеристик користувача:

- відбитки пальців;
- райдужна оболонка ока;
- геометрія руки та обличчя;
- малюнок вен на руках;
- голос;
- почерк або підпис.

У класичному алгоритмі автентифікації пред'явлений користувачем зразок порівнюється зі раніше наданим шаблоном (зразком), де ураховується деяка похибка.

Похибка залежить від необхідного оптимального співвідношення помилок невірних прийнять (FAR) і помилкової відмови (FRR), які відповідають точності і надійності роботи системи [7]:

- FAR – це коефіцієнт вірогідності помилкового пропуску.
- FRR – це коефіцієнт вірогідності помилкової відмови.

При порівнянні зразків з еталоном можливі наступні варіанти [9]:

- зразки належать одному і тому самому користувачу і система ідентифікує поточні зразки як схожі;
- зразки належать іншим користувачам і система ідентифікує поточні зразки як не схожі;
- зразки належать одному і тому самому користувачеві, але система ідентифікує поточні зразки як несхожі – FRR, так як спростовується вірна гіпотеза;
- зразки належать різним користувачам, але система ідентифікує поточні зразки як схожі FAR, так як приймається невірна гіпотеза.

Більш високі значення FAR зазвичай краще в системах, де безпека не має першорядної важливості, тоді як більш високі значення FRR є кращим в додатках з високим ступенем захисту.

Компроміс між FAR і FRR повинен визначатися цілями конкретної прикладної задачі. Наприклад, малий пропуск нелегальних користувачів (FAR) відповідає високому граничного значення (чутливості), але призводить до великого відхилення зареєстрованих користувачів (FRR), тобто до їх низького відсотку пропуску.

В таблиці 1.1 представлена порівняльна характеристика біометричних систем контролю доступу, де наведено переваги, недоліки, імовірність помилкового розпізнання та способи виявлення підміни авторизованого користувача [8].

Таблиця 1.1 - Порівняльна таблиця біометричних систем контролю доступу

Біометричний параметр	Імовірність помилки FAR,%	Переваги	Недоліки	Можливість застосування для виявлення підміни авторизованого користувача
Відбиток пальця	0.001	1. Висока достовірність. 2. Стійкість параметра. 3. Малий ідентифікаційн й	1.Безпосередній контакт з обладнанням. 2. Складність алгоритмів. 3. Легкість	Проводиться впровадження сканерів відбитків пальців у мишки і клавіатури, в корпуси ноутбуків, але

		код. 4. Компактний зчитувач. 5. Низька вартість. 6. Застосування додаткових датчиків (температури, сили натискання).	пошкодження папілярного візерунка пальців, що ускладнює ідентифікацію. 4. Значна залежність якості зчитування від стану шкіри. 5. Можливість підробки відбитка пальця.	більшість з них служать тільки для забезпечення процесу авторизації.
Райдужна оболонка ока	0.0001	1.Стійкість параметра. 2. Висока точність. 3. Надзвичайна складність підробки. 4.Відсутність безпосереднього контакту з обладнанням. 5. Висока швидкодія.	1.Складність алгоритмів. 2. Висока вартість. 3.Низька доступність високих рішень.	Ускладнюється необхідністю постійного напрямку погляду оператора в бік камери, яка володіє малими кутами сканування. Пристрій EyeLock компанії Hoyos Group розроблено для забезпечення процесу авторизації.
Геометрія руки	0.2	1.Стійкість параметра. 2.Простота алгоритмів.	1.Безпосередній контакт з обладнанням. 2.Незручна процедура сканування. 3.Великі розміри зчитувача.	Постійний моніторинг неможливий, якщо руки оператора розташовані поза зоною дії сканера.
Сітківка	0.000001	1. Незмінність параметра з плином часу. 2. Висока точність. 3.Відсутність безпосереднього контакту з обладнанням.	1.Складність зчитування. 2.Складність алгоритмів. 3.Високий час обробки шаблону. 4.Висока вартість системи.	Відсутня в зв'язку з необхідністю виконання певних умов для зчитування характеристики.
Геометрія особи	Від 5 до 0.0047	1.Можливість безперервної аутентифікації. 2. Відсутність безпосереднього контакту з обладнанням. 3. Низька вартість.	1. Залежність від умов освітлення, положення голови. 2.Залежність від міміки обличчя. 3.Залежність від перешкод (окуляри, головний убір, зміна зачіски).	Можливість застосування для постійного моніторингу підміни оператора є, але існує ряд обмежень, спричинених недоліками методу. Основне застосування –

				процес авторизації.
Вена руки	0.0008	1. Висока точність. 2.Відсутність безпосереднього контакту обладнанням. 3.Прихованість характеристики.	1.Чутливість сканера до природного і штучного освітлення.	Постійний моніторинг неможливий, якщо руки оператора розташовані поза зоною дії сканера.

Виходячи з наведеної характеристики з найбільш популярних систем контролю доступу найнадійнішими є автентифікація користувача за сітківкою ока та райдужної оболонки ока, але ці фізіологічні ознаки також мають недоліки.

Таким чином в даному підрозділі було проведено аналіз активно експлуатованих біометричних систем ідентифікації користувачів і проведена оцінка можливості їх застосування для виявлення підміни авторизованого законного користувача. Реалізація механізмів ідентифікації для входу в системи засновані на біометричній ідентифікації можуть суттєво підвищити ступінь захисту в ІС.

Зауважимо, для того, щоб механізм біометричної ідентифікації, якомога менше створював дискомфорт для користувача системи потрібно, щоб він був простий у використанні.

Додатково були наведені можливі сценарії поведінки системи, яка порівнює зразки з еталонним значенням.

Висновки до розділу 1

В даному розділі були розглянуті переваги та недоліки парольних, атрибутивних та біометричних систем контролю доступом до інформації.

Було наведено результати дослідження американської організації RSA в якому висвітлювався процентний розподіл місць зберігання користувацьких паролів, що свідчить про неефективність їх використання.

Виходячи з описаного, зроблено висновок, що парольні та атрибутивні системи контролю доступу уже не є ефективними, а біометричні є альтернативою, яка є дуже стійкою для контролю доступу до системи.

Проведений аналіз сучасних біометричних систем контролю доступу, наведена порівняльна таблиця в якій описані основні біометричні параметри, імовірність помилки FAR, переваги, недоліки та можливості застосування для виявлення підміни авторизованого користувача.

РОЗДІЛ 2
АНАЛІЗ ДАНИХ КЛАВІАТУРНОГО ПОЧЕРКУ КОРИСТУВАЧА

2.1 Характеристики клавіатурного почерку

Клавіатурний почерк - це набір динамічних характеристик роботи на клавіатурі [9]. Автентифікація по клавіатурного почерку дозволяє забезпечити підвищену точність, неможливість відмови від авторства, зручність для оператора. Також автентифікація по клавіатурному почерку не потребує придбання додаткового апаратного та програмного забезпечення, потрібна лише клавіатура [9].

Аналіз клавіатурного почерку бере початок з тих часів, коли широке поширення мав радіозв'язок з використанням азбуки Морзе. Досвідченого радиста розпізнавали по швидкості, своєму стилю і якості передачі сигналів, на відміну від початківця. Так само, як і любитель музики може дізнатися виконавця пісні на слух. Приблизно ті ж самі принципи покладені в основу розпізнавання людини, яка працює на клавіатурі комп'ютера.

Ідентифікації за допомогою клавіатурного почерку має характерний ряд переваг і недоліків [10].

До переваг відносяться:

- стабільність клавіатурного почерку конкретної людини, що дозволяє з більшою вірогідністю ідентифікувати користувача, працюючого з клавіатурою;
- відносна дешевизна впровадження системи розпізнавання;
- можливість контролювати як доступ до ресурсів, так і фізичний стан співробітника.

До недоліків методу можна віднести:

- використання можливо тільки для розпізнавання користувачів зі сформованим клавіатурним почерком.

Можливості аналізу клавіатурного почерку дозволяють вирішувати такі завдання [10]:

- контролювати фізичний стан користувачів;
- відмовитись від використання паролів;
- надати користувачам більш простий спосіб входу в мережу.

Крім того, на сьогоднішній день це єдина технологія, яка може використовуватися за двома призначеннями [10]:

- для ідентифікації користувача, який претендує на доступ до комп'ютерної системи;
- для проведення таємного клавіатурного моніторингу працюючих користувачів.

Системи, що вирішують ці завдання, різняться тим, що в першому випадку ідентифікація користувача здійснюється по короткій парольній фразі, а в другому випадку – за довільним текстом.

Основними характеристиками клавіатурного почерку користувача є [10]:

- час утримання клавіші (ЧУК);
- паузи між натисканнями;
- наявність накладень;
- кількість помилок при вводі;
- ступінь ритмічності під час набору;
- швидкість набору;
- особливості використання службових клавіш.

На рис. 2.1 показані основні характеристики клавіатурного почерку (КП): час утримання клавіші та пауза між натисками.

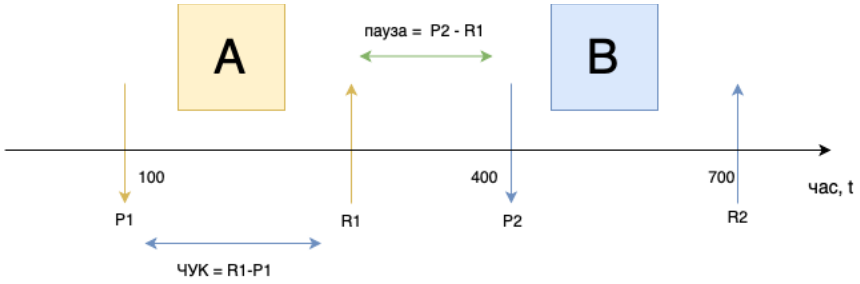


Рисунок 2.1 - Демонстрація характеристик КП

Накладення натискань клавіш відбувається тоді, коли одна клавіша ще не відпущена, а інша вже натискається. З підвищенням швидкості набору тексту збільшується число накладень. Більш докладна схема з урахуванням усіх характеристик почерку зображена в Додатку Б.

В даному підрозділі було описано, що таке клавіатурний почерк, його переваги та недоліки, а також наведені його найголовніші характеристики графічно та у вигляді списку.

2.2 Переваги та недоліки використання клавіатурного почерку

Клавіатурний почерк, як спосіб автентифікації має переваги та недоліки.

Основними перевагами є простота у реалізації та впровадження, тобто виключно програмна реалізація. Користувач вводить контрольну фразу або текст з стандартного пристрою вводу (клавіатури), а значить немає потреби у придбані та використанні додатково апаратного забезпечення. Автентифікація за клавіатурним почерком є дешевим способом автентифікації за біометричних ознаками суб’єкту доступу.

Відсутня потреба користувача у будь-яких додаткових діях, крім звичних. Користувач у будь-якому випадку використовує пароль, який можна назначити контрольною фразою за допомогою якої буде проводитися автентифікація.

Присутність можливості скритної автентифікація користувача – людина може не знати, що активована додаткова перевірка, а значить не зможе повідомити про це потенційному зловмиснику.

Варто сказати про недоліки даного способу автентифікації користувача.

Кожна система та застосунок потребує навчання, тобто потрібно зберігати початкові дані за допомогою яких буде відбуватися подальше порівняння.

Існує сильна залежність від ергономічності клавіатури, у випадку зміни клавіатури, система або застосунок потрібно навчати заново.

Також присутня залежність від психофізичного стану користувача. Якщо людина захворіла або відчуває себе недобре, то вірогідно не зможе пройти автентифікацію.

В даному підрозділі було описано переваги та недоліки використання клавіатурного почерку як біометричного способу автентифікації користувача у системі.

Перевагами є простота у реалізації, відсутність у додатковому апаратному забезпеченні, а недоліками є залежність від клавіатури та психофізичного стану оператора-користувача.

2.3 Огляд існуючих систем прихованого клавіатурного моніторингу

В даний час можна виділити ряд розробок, що реалізують розпізнавання клавіатурного почерку саме у десктопній версії:

Система контролю співробітників «Стахановець» [11]. Даний комплекс включає перехоплення дзвінків співробітників, контроль використання принтерів, спостереження за співробітниками за допомогою веб-камер, облік часу роботи і інші інструменти. У тому числі система містить модуль для аналізу клавіатурного почерку.

Моніторинг почерку здатний [11]:

- ідентифікувати користувача за набраним текстом;
- визначити, чи дійсно за комп'ютером закріплений співробітник, або дії за нього робить хтось інший;
- виявити несанкціонований доступ до важливої інформації;

- знайти автора конкретного тексту;
- визначити стан користувача (алкогольне або наркотичне сп'яніння, паніка, страх, стрес та інші змінені психічні стани).

Розробка Горбунова І. [12]. Користувачам видаються логіни і паролі для віддаленого доступу до комп'ютерів. При введенні даних перевіряється не тільки факт збігу з логіном і паролем з бази даних, а й характеристики клавіатурного почерку (час утримання клавіш і паузи між натисканнями).

Розробка Савінова А. [13]. Програма враховує швидкість натискання клавіш і дозволяє визначити, хто знаходиться за комп'ютером. При цьому набраний текст аналізується, вибираються найбільш часто використовувані слова. На основі порівняння обраних слів зі словником, програма визначає професію користувача.

Також присутня можливість визначити темперамент і настрій користувача. Наприклад, за словами автора, якщо символи на клавіатурі набираються рівномірно, то, значить, це акуратний і пунктуальний чоловік, здатний до рутинної роботи. А рівні паузи між словами свідчать про впевненість і відсутність негативних емоцій.

Якщо людина довго шукає клавіші, то, навпаки, можна говорити про її депресивний стан і негативне сприйняття. Висока швидкість друку на початку роботи і її зниження до кінця вказують на стомленість.

Іншим ПЗ є додаток «Клавіатурний почерк 1.0» [14], який призначений для підрахунку різних характеристик клавіатурного почерку: швидкість натискань, швидкість введення символів, відсоток помилок, відсоток перекритів, відсоток ритмічності. Він не дозволяє зберігати дані про клавіатурні почерки декількох користувачів і проводити їх порівняння, щоб ідентифікувати користувача рис. 2.2:

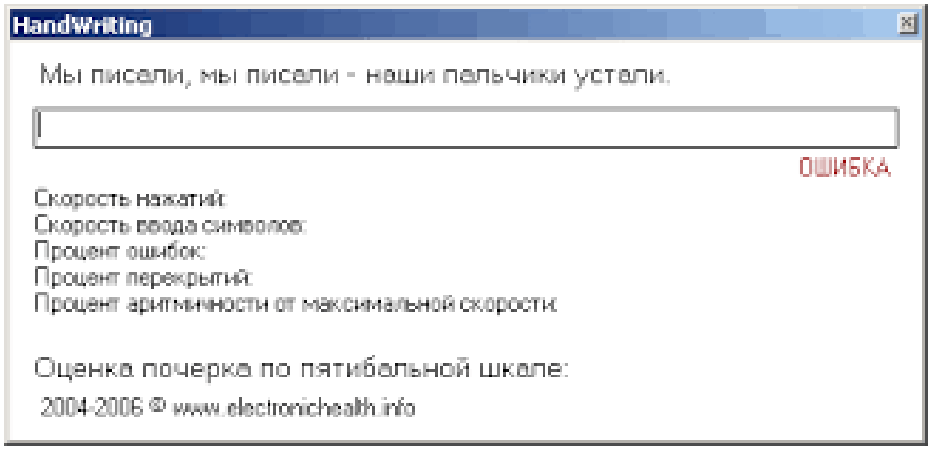


Рисунок 2.2 - Інтерфейс програми «Клавіатурний почерк 1.0»

В даному підрозділі було проведено огляд існуючих програм для клавіатурного моніторингу та аналізу почерку. Наведені основні характеристики програм, переваги та недоліки.

2.4 Огляд математичних алгоритмів розпізнавання клавіатурного почерку

Алгоритми розпізнавання клавіатурного почерку можна розділити на три групи [15]:

- алгоритми, які аналізують почерк під час введення пароля;
- алгоритми, які аналізують почерк після введення додаткового текстового фрагмента або фрази;
- алгоритми, які постійно проводять прихований моніторинг клавіатурного почерку користувача.

Алгоритми першої групи забезпечують найбільшу швидкодію: користувачеві потрібно лише ввести свій пароль. Однак точність в цьому випадку невисока, особливо в разі короткого пароля.

Алгоритми другої групи дозволяють забезпечити більшу точність у порівнянні з першою групою. Однак на введення додаткового фрагмента тексту потрібен час,

що може викликати негативні емоції у користувача, особливо в разі, якщо йому часто доводиться проходити процедуру автентифікації.

Алгоритми третьої групи можуть забезпечити високу точність. При цьому вони вимагають більше ресурсів. Перевагою цієї групи є можливість розпізнати зловмисника, який використовує комп'ютер, на якому раніше авторизувався оператор. В цьому випадку система може заблокувати комп'ютер для запобігання доступу до конфіденційної інформації.

При моніторингу клавіатурного почерку розглядаються невеликі фрагменти фраз, що містять приблизно 10-40 символів. Ця кількість обумовлена інтервалом копіювання користувача.

Під інтервалом копіювання розуміється число символів, які можуть бути надруковані в точності після однократного перегляду тексту [15]. Його величина залежить від рівня професійної підготовки оператора.

Дослідження показують, що чим уривчастий удар і чим більше ритмічний друк, тим менше спостерігається накладень. Коли клавіші швидше натискаються, ніж вдаряються, накладень виходить більше. У разі дуже уривчасто друку час утримання клавіші може становити 65 мс і менше, а при великій кількості накладень час утримання збільшується до 120 мс і більше. Як правило, середній час утримання клавіші становить 80-100 мс [15].

При розробці інформаційної системи, що забезпечує перевірку клавіатурного почерку, необхідно враховувати, що описані методи ефективні тільки для користувачів з великим досвідом роботи на комп'ютері та зі сформованим клавіатурним почерком. Достатня ймовірність ідентифікації користувача може бути досягнута, якщо термін активного використання комп'ютера становить як мінімум 6 місяців [15].

Для порівняння двох зразків клавіатурного почерку між собою найчастіше використовуються наступні характеристики [16]:

- час утримання клавіші;
- тривалість паузи між натисканнями;
- наявність накладень;

- кількість помилок введення;
- системні клавіші, яким надається перевага: наприклад, для введення великих літер, деякі користувачі використовують клавішу «Caps Lock», а інші користувачі натискають кнопку «Shift».

Робота системи розпізнавання користувача по клавіатурному почерку складається з двох етапів.

На першому етапі відбувається навчання системи, тобто накопичення інформації про клавіатурні почерки різних користувачів, наприклад, співробітників однієї організації.

На другому етапі відбувається розпізнавання користувача на підставі отриманого зразка почерку і еталонів, відомих системі після завершення першого етапу.

Одним із способів підвищення точності роботи алгоритму є постійне оновлення еталона для користувача, що успішно пройшов автентифікацію. Це дозволить варіантам клавіатурного почерку не застаріти і завжди відповідати цьому рівню швидкості друку користувача.

Для визначення ступеня відповідності двох зразків клавіатурного почерку існує ряд алгоритмів, які забезпечують автентифікацію користувача:

- ймовірнісно-статистичний метод;
- гістограмний метод;
- метод на основі нейронних мереж.

В даному підрозділі було проведено огляд основних алгоритмів аналізу та розпізнавання клавіатурного почерку, а також висвітлені основні характеристики почерку, які використовуються алгоритмами для більш ґрунтовного аналізу.

2.4.1 Огляд ймовірнісно-статистичного методу

У роботі [17] вдалося встановити, що ЧУК при наборі тексту є бімодальним розподілом. У другий нормальний розподіл елементи потрапляють у разі, якщо

набір відбувався з накладенням клавіш. Отже, необхідно розділяти ЧУК з накладеннями і без накладень.

Таким чином, можливо отримати дві незалежні вибірки і дві величини математичного очікування ЧУК - з накладеннями і без накладень. Такий підхід дозволяє зменшити число помилок 1-го і 2-го роду, збільшивши точність ідентифікації користувача. Кожна з двох вибірок підпорядковується нормальному закону розподілу.

Робота ймовірнісно-статистичного алгоритму складається з декількох етапів: збір статистики характеристик клавіатурного почерку різних користувачів. Для цього проводиться ідентифікація оператора по його унікальному логіну. Далі заповнюється масив KeyEventsArr, поки не буде натиснуто достатню кількість клавіш. Структура масиву представлена в табл. 2.1 [17]:

Таблиця 2.1 – Структура масиву KeyEventsArr

Клавіша 1	…	Клавіша N
Подія KeyUp або KeyDown	…	Подія KeyUp або KeyDown
Ітерація лічильника, на якому відбулась подія	…	Ітерація лічильника, на якому відбулась подія

Наступним етапом є підрахунок ЧУК. Для кожної події натискання клавіші знаходиться подія відпускання цієї клавіші. Величина ЧУК становить різницю між тиком відпускання і тиком натискання, діленим на частоту лічильника з високою роздільною здатністю. Це дозволяє отримати значення ЧУК в мілісекундах. Для зберігання характеристик клавіатурного почерку користувача використовується структура такого вигляду [17]:

Таблиця 2.2 – Вибірка клавіатурного почерку

Клавіша	*ЧУК-1*	*ЧУК-2*	*Кількість-1*	*Кількість - 2*	*Вибірка нормального розподілу - 1*	*Вибірка нормального розподілу -2*
А						
Б						
…						

В поле «Вибірка» заносяться значення ЧУК, розділені символом «;». У полі «Кількість» зберігається кількість елементів у вибірці. При отриманні достатньої для аналізу кількості ЧУК, підраховується математичне очікування ЧУК і заноситься в поле «ЧУК».

Наступним етапом роботи системи є автентифікація користувача по клавіатурному почерку. У розглянутому алгоритмі порівняння почерків відбувається за принципом «один до одного», тобто завантажується еталон конкретного оператора, логін якого ввів користувач. У разі успішної автентифікації відбувається процес авторизації, інакше користувач отримує відмову.

Для визначення клавіатурного почерку тут використовується порівняно невелика кількість клавіш - не більше 20.

Для порівняння отриманого зразка клавіатурного почерку користувача з еталонним відбувається шляхом підрахунку міри Евкліда для кожної клавіші. При цьому ЧУК-1 і ЧУК-2 аналізуються як окремі елементи.

Для підрахунку відстані Евкліда використовується формула 2.1:

$$\mathbf{M} = \sqrt{\sum_{\mathbf{i=1}}^{\mathbf{V}} (\mathbf{A_i} - \mathbf{B_i})^{\mathbf{2}}}, \qquad (2.1)$$

де M – розраховане значення Евклідової відстані;

V – кількість вибірок часу Hold, DownDown, UpDown, що відповідає кількості аналізованих клавіш;

A_i – час утримання клавіші з поточного зразка клавіатурного почерку користувача, що претендує на доступ;

B_i – час утримання клавіші, що зберігається в шаблоні почерку.

Якщо несхожість менша, ніж поріг доступу, то вважається, що почерки співпадають і користувач пройшов автентифікацію, інакше автентифікація не пройдена.

В даному підрозділі було проведений огляд ймовірнісно-статистичного методу, визначені його основні етапи і основні математичні формули, що використовуються для аналізу даних.

2.4.2 Огляд гістограмного методу

Гістограмний метод розглянуто в [18]. У гістограмному методі автентифікація користувача відбувається шляхом аналізу клавіатурного почерку при введенні певної контрольної фрази. За результатами аналізу виноситься рішення про успішну автентифікації або відбувається відмова в доступі.

В якості основних параметрів клавіатурного почерку використовується час утримання і час пауз між натисканнями. Нехай контрольна фраза містить q символів і час, витрачений користувачем на введення цієї фрази, становить T. Тоді необхідно проаналізувати r подій клавіатури формула 2.2 [18]:

$$\boldsymbol{r = q + p} \tag{2.2}$$

де q – кількість утримань (натиску) клавіш; $p = q–1$ – кількість пауз між утриманнями.

При введенні можливі накладення натискань. Такий вид події клавіатури слід інтерпретувати як негативні значення тривалості пауз між натисканнями.

Для опису роботи алгоритму введено такі позначення:

τ_i - значення тривалості утримання клавіші i, причому$\tau_i > 0$;

τ_{ij}- алгебраїчне значення тривалості паузи між утриманнями клавіш i та j.

Тобто $\tau_{ij} \geq 0$ для звичайної паузи та $\tau_i < 0$ при накладення часових рамок утримань клавіш.

Як приклад можна розглянути введення користувачем контрольної фрази з шести символів. В цьому випадку $r = 11$, $q = 6$, $p = 5$.

Часова розкладка процесу введення контрольної фрази індивідуальна для кожного користувача. Для кожного результату введення в відповідність ставиться r - мірний вектор клавіатурних параметрів формула 2.3 [18].

$$\boldsymbol{V} = \{\boldsymbol{V_j}\}, \boldsymbol{j} = \overline{\boldsymbol{1, r}} \quad (2.3)$$

де кожен компонент V_j відповідає тривалості події клавіатури: утримання клавіші або паузи між утриманнями, що сталося за період часу *T*.

Таким чином вектор біометричних параметрів *V* можна інтерпретувати як зразок клавіатурного почерку користувача.

Для кожного користувача необхідно скласти навчальну вибірку зразків *s*-класу. Для цього використовується серія з *L* зразків клавіатурного почерку цього користувача формула 2.4 [18].

$$\boldsymbol{\psi^{(s)}} = \{\boldsymbol{V_i}\}, \boldsymbol{i} = \overline{\boldsymbol{1, L}} \quad (2.4)$$

Допускається наявність в системі безлічі $K = \{k_1, k_2, \dots, k_n\}$ користувачів, для кожного з яких представлений власний еталон клавіатурного почерку. Даний еталон співвідноситься з одним з класів безлічі $S = \{s_1, s_2, \dots, s_n\}$. Отже, сукупність користувачів системи $\{K\}$ однозначно відображається на безліч класів $\{S\}$. Таким чином, необхідно *M* навчальних вибірок для того щоб сформувати еталони *M* легітимних користувачів формула 2.5 [18]:

$$\boldsymbol{\psi^{(s_1)}}, \boldsymbol{\psi^{(s_2)}}, \boldsymbol{\psi^{(s_n)}} \quad (2.5)$$

Коли система працює в режимі автентифікації, невідомий користувач повинен пред'явити їй зразок свого клавіатурного почерку.

Почерк характеризується вектором біометричних параметрів $V^{(S_x)} = \{V_j\}, j = \overline{1, r}$. На основі отриманого вектора система формує еталонний опис невідомого *x*-класу, після чого порівнює його з еталонами всіх відомих системі $\{k_1, k_2, \dots, k_n\}$ користувачів. В результаті порівняння виноситься рішення про особу користувача. При цьому необхідно вирішити задачу класифікації вектора $V^{(S_x)}$ на $M + 1$ взаємовиключних класів. З них *M* класів відносяться до безлічі $S = \{s_1, s_2, \dots, s_n\}$, тобто позначають відомих системі користувачів, а $M + 1$ клас відповідає за опис всіх інших користувачів, тобто чужих.

Таким чином, мета навчання системи розпізнавання клавіатурного почерку - це формування еталонних описів для класів, відповідним користувачам. Найчастіше, використовується вид вирішальних правил, запозичений з теорії статистичних рішень. Необхідно сформувати ставлення правдоподібності для умовних густин розподілу і порівняти з порогом C_n формула 2.6 [18]:

$$\frac{\boldsymbol{w_r(V|s_1)}}{\boldsymbol{w_r(V|s_2)}} \geq \boldsymbol{C_n} \tag{2.6}$$

де $w_r(V|s_j)$ – умовна сумісна r-вимірна густота вірогідності вибіркових значень $\{V_j\}, j = \overline{1,r}$ за умови їх приналежності до класу s_j.

Для апроксимації розподілу векторів можна використовувати гауссового розподіл. Однак необхідно враховувати, що на параметри клавіатурного почерку впливають психофізичний стан користувача, добові біоритми і інші фактори.

Таким чином, для клавіатурного почерку характерні флуктуації параметрів, тому має сенс використовувати змішаний гауссовий розподіл з декількома центрами.

При цьому вирішальне правило має вигляд формула 2.7 [18]:

$$\boldsymbol{S} = \begin{cases} \boldsymbol{S_c}, \text{якщо } \boldsymbol{C \geq C_n} \\ \boldsymbol{S_x}, \text{якщо } \boldsymbol{C < C_n} \end{cases} \tag{2.7}$$

де C_n - значення порога, яке вибирається з урахуванням необхідної точності розпізнавання.

Таким чином система визначає, чи відповідає пропонований зразок клавіатурного почерку користувача еталону. У разі успішної автентифікації користувач може перейти до авторизації. В іншому випадку вважається, що користувач не автентифікований.

В даному підрозділі було проведено огляд гістограмного методу, визначені його основні етапи і основні математичні формули, що використовуються для аналізу даних.

2.4.3 Огляд методу на основі нейроних мереж

У роботах [19], [20] розглядається можливість використання штучної нейронної мережі для аналізу клавіатурного почерку.

Штучна нейронна мережа (ШНМ) - мережа, що складається з штучних нейронів [21]. ШНМ - предмет дослідження нейроінформатики і одна з гілок вивчення і моделювання штучного інтелекту.

Штучні нейронні мережі і нейрони - це математичні моделі біологічних нейронних мереж і нейронів (клітин, з яких складається нервова система людини) [21].

Використання нейромережевого підходу дозволяє досягти більшої точності, в порівнянні з ймовірно-статистичними методами. Проте, необхідно враховувати дві групи принципових проблем.

По-перше, до них відносяться власні проблеми штучних нейронних мереж, пов'язані можливістю виникнення невизначено довгого процесу навчання, тупиків, стану «паралічу».

По-друге, можуть виникнути проблеми, які визначаються біометричною природою розпізнавання образів, головна з яких навчання.

Для аналізу характеристик клавіатурного почерку в якості вихідних даних пропонується використовувати два вектори, один з яких містить значення тривалості натискань окремих клавіш, а другий описує інтервали між натисканнями.

На основі описаних часових рядів формується фазова траєкторія на площині, утворена двома осями: віссю часу натискання клавіш і віссю часових інтервалів між двома сусідніми натисканнями.

Фазова траєкторія - це крива в фазовому просторі, складена з точок, що представляють стан динамічної системи в послідовні моменти часу протягом усього часу еволюції [21].

Дана фазова траєкторія аналізується за допомогою нейромережевої структури, яка самоорганізовується. Навчання нейронної мережі пропонується проводити на основі алгоритму нейронного газу [21].

Зазначена нейронна мережа є одношаровою, і в ній кожен з нейронів з'єднаний з кожним елементом вхідного вектору. Після початкового визначення вагових коефіцієнтів розраховується відстань за Евклідом між вхідним вектором і векторами вагових коефіцієнтів, що належать відповідним нейронам.

На кожній ітерації роботи алгоритму нейрони сортуються залежно від їх відстані до вхідного вектору. Таким чином, після сортування нейрони розташовуються в порядку, відповідному збільшенню віддаленості формула 2.8. [21]

$$\boldsymbol{d_0 < d_1 < d_2 < d_{n-1}} \tag{2.8}$$

де d_i - віддаленість i-го нейрона, що займає в результаті сортування m-ну позицію в послідовності, яку очолює нейроном- переможцем;

d_0 - віддаленість нейрона-переможця.

Переможцем визнається той нейрон, якому відповідав би вектор вагових коефіцієнтів з найменшою відстанню евкліда до вхідного вектору.

Значення функції сусідства для і-го нейрона визначається відповідно до формули 2.9 [21]:

$$\boldsymbol{G(x,i) = \exp(-\frac{m(i)}{\lambda})} \tag{2.9}$$

де $G(x,i)$- значення функції сусідства;

і - номер нейрона;

$m(i)$ - черговість, отримана в результаті сортування;

λ - параметр, значення якого з часом зменшується.

При $\lambda = 0$ адаптації піддається тільки нейрон-переможець, але при $\lambda \neq 0$ уточненню підлягають ваги багатьох нейронів, при цьому рівень уточнення залежить від величини $G(x,i)$.

Нейрон-переможець і всі нейрони, що лежать в межах його околиці, піддаються адаптації, в ході якої їх вектори ваг змінюються за правилом Кохонена формула 2.10 [21]:

$$\boldsymbol{w_i(k+1) = w_i(k) + n_i(k)[x - w_i(k)]}, \quad (2.10)$$

де $n_i(k)$- коефіцієнт навчання.

Описаний алгоритм аналізу параметрів клавіатурного почерку дозволяє коректно обробити до 85% поданих на вхід векторів.

В даному підрозділі було проведений огляд методу на основі нейронних мереж, визначені його основні етапи і основні математичні формули, що використовуються для аналізу даних.

2.5 Недоліки існуючого методу аналізу клавіатурного почерку

Автори І.Г. Сидоркіна та А.Н. Савінов в своїй роботі [9] зробили аналіз клавіатурного почерку з використанням гістограмного методу. Алгоритми були реалізовані у вигляді відповідної програми з використанням неактуальних технологій, без використання сучасних методів розробки веб-застосунків. ПЗ вимагало встановлення спеціальної програми, яка проводила розпізнавання клавіатурного почерку локально на робочій машині. Це зумовлювало повільну роботу на комп'ютерах невеликої потужності, а також знижувало ефективність розпізнавання.

Певні складові частини алгоритмів ПЗ були зайвими та не виконували своїх функцій у повному обсязі для найкращої роботи програми.

Взявши за основу роботи авторів І.Г. Сидоркіної та А.Н. Савінова автор даної роботи пропонує виправити виявлені недоліки та застосувати розроблене ним відповідне ПЗ, а саме:

По-перше, розроблене ПЗ, яке буде працювати виключно у веб-середовищі, тобто у браузері. Тим самим відсутня необхідність у встановленні додаткового ПЗ на робочу машину, так як для ефективної роботи достатньо мати відмінне інтернет з'єднання та встановлений браузер останньої версії. В результаті чого швидкість виконання програмного коду та запропонованих алгоритмів зросте в рази.

По-друге, змінити запропоновані алгоритми в роботі [9], шляхом видалення деяких складових частин, тим самим підвищити ефективність виконання алгоритмів навчання та розпізнавання.

По-третє, зменшити поріг розпізнавання клавіатурних почерків в два рази, так як веб-технології мають високоточні механізми фіксування часу, які дозволять підвищити точність обчислення та порівняння зразків.

Висновки до розділу 2

В даному розділі були описані основні характеристики клавіатурного почерку, такі як час утримання клавіші, паузи між натисканнями, кількість помилок тощо.

Було проаналізовано ПЗ прихованого клавіатурного моніторингу, а саме:

- розробка Горбунова І.І. – автентифікація відбувається по характеристикам клавіатурного почерку після того як система надала логін та пароль для доступу в систему;
- розробка Савінова А.Н. – програма враховує швидкість натискання клавіш і дозволяє визначити, хто знаходиться за комп'ютером. При цьому набраний текст аналізується, вибираються найбільш часто використовувані слова. На основі порівняння обраних слів зі словником, програма визначає професію користувача.
- безкоштовне ПЗ «Клавіатурний почерк 1.0» - призначений для підрахунку різних характеристик клавіатурного почерку: швидкість натискань, швидкість введення символів, відсоток помилок, відсоток перекритів, відсоток ритмічності. Вона не дозволяє зберігати дані про клавіатурний почерк декількох користувачів і проводити їх порівняння, щоб ідентифікувати користувача.

В останній частині даного розділу був проведений загальний огляд основних математичних методів аналізу даних отриманих в результаті моніторингу клавіатурного почерку, а саме:

- ймовірнісно-статистичний метод;
- гістограмний метод;
- метод на основі нейронних мереж.

Для кожного методу були наведені основні характеристики та формули. Показано, що кожен з представлених методів має різний рівень складності реалізації.

Виявлені та описані основні недоліки ПЗ та алгоритмів, які були запропоновані авторами І.Г. Сидоркіною та А.Н. Савіним [9]. В результаті глибокого аналізу відповідних матеріалів автором даної роботи запропоновано шляхи виправлення описаних вище недоліків.

РОЗДІЛ 3
АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АНАЛІЗУ ДАНИХ КЛАВІАТУРНОГО ПОЧЕРКУ КОРИСТУВАЧА

3.1 Функціональні можливості системи

Традиційні методи ідентифікації і автентифікації, що засновані на використанні переносних ідентифікаторів, а також паролів і кодів доступу, мають ряд суттєвих недоліків, пов'язаних з тим, що для встановлення автентичності користувача застосовуються атрибутивні і засновані на знаннях розпізнавальні характеристики, які можна підробити або вкрасти.

Варто додати, що зі стрімким розвитком веб-технологій та хмарних сховищ даних, постає питання використання надійних методів біометричної ідентифікації користувачів у веб-середовищі. Тому постає питання, яким чином інтегрувати автентифікацію користувача за клавіатурним почерком у системі, яка знаходиться на віддалених серверах та БД.

В роботі [9] пропонується рішення, яке має деякі недоліки та не доопрацювання, яке автор представленої роботи пропонує виправити та вдосконалити, а саме:

По-перше, відмовитися від розробки декстопної версії за допомогою форм Windows - WinForms, а запропонувати мігрувати до веб-середовища, використовуючи сучасні мови програмування та фреймворки, такі як .NET Core та Javascript.

По-друге, змінити запропоновані І.Г. Сидоркіною та А.Н. Савіним алгоритми навчання та розпізнавання, а саме:

- в діаграмі варіантів використання прибрати файл журналізації, так як усі дані зберігаються в БД;
- змінити формат передачі даних, відмова від XML та використання більш зручного формату JSON;

- інформація щодо натисків клавіш передаються одним об’єктом, а не нескінченним потоком подій та обчислюються безпосередньо на сервері;
- в алгоритмі розпізнавання, якщо результат порівняння зразку клавіатурного почерку з еталонним не буде успішним, то система покаже повідомлення про те, що автентифікація не була пройдена, замість створення нового користувача з новим зразком почерку;
- в алгоритмі навчання не перевіряти введений зразок почерку з почерками усіх користувачів, тому що система була повідомлена про логін користувача. Порівняння буде здійснюватися лише зі зразками поточного користувача.

По-третє, знизити поріг розпізнавання, який був запропонований у роботі [9] в два рази, тому що сучасні технології з точним інструментарієм дозволив підвищити чутливість та ефективність розпізнання.

Запропоновані виправлення підвищать ефективність роботи та швидкість виконання програмного коду для алгоритму навчання та розпізнавання. Порівняння ефективності алгоритмів детально описано в підрозділі 3.2.

Зміни в алгоритмі та інтерфейсі користувача збільшить продуктивність та час розгортання ПЗ на робочі машини користувачів.

Розпочинати проектування системи потрібно з опису функціональних можливостей. Для цього потрібно використати UML-діаграми (діаграми варіантів використання).

UML-діаграма – це діаграма, яка відображає співвідношення між акторами та прецедентами і є основною частиною моделі прецедентів, яка дозволяє описати систему на концептуальному рівні [22].

Діаграма варіантів використання клавіатурного почерку для системи розпізнавання запропонована автором зображена на рис. 3.1:

На даній діаграмі показано, що при запуску даного ПЗ користувач має наступні можливості:

- зареєструватись у системі з власним логіном та паролем;

- додати будь-яке кодове слово (текст) по якому буде відбуватися навчання системи та розпізнавання почерку;
- переглядати дані клавіатурного почерку у табличному та графічному вигляді;
- навчати систему шляхом внесення зразку клавіатурного почерку в базу даних.

Рисунок. 3.1 UML-діаграма варіантів використання ПЗ

Отже UML-діаграма явно відображає можливості ПЗ, яке буде розпізнавати клавіатурний почерк користувача.

В даному підрозділі була надана інформація щодо функціональних можливостей системи. Перераховані основні функції за допомогою UML-діаграми.

3.2 Порівняння ефективності алгоритмів

Для порівняння ефективності алгоритмів було обчислене нотація великого «О» (ориг. – Big «O» notation) [23].

Нотація великого «О» - це математична нотація, яка описує поведінку функції, коли аргумент прагне до певного значення або нескінченності. Він є членом

сімейства нотацій, винайдених П. Бахманом, Е. Ландау та іншими, які в сукупності називаються нотаціями Бахмана-Ландау або асимптотичними нотаціями [23]. Простіше кажучи, нотація великого «О» описує складність коду з використанням алгебраїчних термінів.

Для того, щоб обчислити нотацію великого «О» потрібно розбити алгоритми на окремі операції та порахувати нотацію великого «О» для кожного індивідуального кроку. Для того щоб порахувати суму усіх нотацій, потрібно скористатися формулою 3.1 [23]:

$$\boldsymbol{f_1 = O(g_1), f_2 = O(g_2) \rightarrow f_1 + f_2 = O(\max(g_1, g_2))} \quad (3.1)$$

де f_1, f_2 – функції нотацій великого "О".

Тобто для обчислення потрібно обрати функцію результат якої дасть максимальний результат поміж інших функцій. В таблицях 3.1, 3.2, 3.3 та 3.4 показано порівняння між двома алгоритмами [9] та запропонованими автором даної роботи.

Таблиця 3.1 – Операції алгоритму навчання [9] та значення нотації великого «О»

№	Операція	Значення нотації великого «О»	Коментар
1	Ввод логіну та паролю та перевірка валідності	O(1)	Перевірка валідності логіну та паролю відбувається один раз при виконанні алгоритму. Час виконання цієї операції не залежить від кількості вхідних даних.
2	Ініціалізація динамічного масиву подій клавіатури	O(1)	Ініціалізація масиву відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.
3	Визначення події натиску клавіш на клавіатуру	O(1)	Визначення події натиску відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.
4	Визначення виду події KeyUp та	O(1)	Визначення виду події відбувається один раз при виконанні коду. Час

	KeyDown		виконання цієї операції не залежить від кількості вхідних даних.
5	Визначення клавіші з якої відбулась подія	O(1)	Визначення клавіші, яка створила подію відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.
6	Запис значення в масив	O(1)	Додавання значення в масив відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.
7	Обчислення ЧУК та ЧУК з накладеннями	O(n^3)	Потрійний цикл для ітерації усіх значень та їх обчислення, а також визначення наявності накладень між натисками.
8	Зберігання зразку до БД	O(1)	Зберігання об’єкту зразку до БД відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.

Значення нотації великого «О» для алгоритму навчання [11] дорівнює значенню відповідно до формули 3.2 .

$$\boldsymbol{f_1 = O(1), f_2 = O(n^3) \rightarrow f_1 + f_2 = O(\max(1, n^3)) = n^3} \quad (3.2)$$

Таблиця 3.2 – Операції алгоритму навчання запропонованого автором даної роботи та значення нотації великого «О»

№	Операція	Значення нотації великого «О»	Коментар
1	Ввод логіну та паролю та перевірка валідності	O(1)	Перевірка валідності логіну та паролю відбувається один раз при виконанні алгоритму. Час виконання цієї операції не залежить від кількості вхідних даних.
2	Формування об’єкту події натиску на клавіатуру	O(1)	Формування події натиску відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.

3	Додання об’єкту до фінального масиву	O(1)	Операція додання елементу до масиву відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.
4	Перевірка повноти набраного кодового слова (тексту)	O(1)	Перевірка повноти набраного тексту відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.
5	Передача фінального масиву на серверну частину	O(1)	Відправлення фінального масиву до серверу відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.
6	Обчислення ЧУК з накладеннями	$O(n^2)$	Потрійний цикл для ітерації усіх значень та їх обчислення, а також визначення наявності накладень між натисками.
7	Зберігання зразку до БД	O(1)	Зберігання об’єкту зразку до БД відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.

Значення нотації великого «О» для алгоритму навчання, який було запропоновано використовувати дорівнює відповідно до формули 3.3.

$$\boldsymbol{f_1 = O(1), f_2 = O(n^2) \rightarrow f_1 + f_2 = O(\max(1, n^2)) = n^2} \quad (3.3)$$

Таблиця 3.3 – Операції алгоритму розпізнавання [9] та значення нотації великого «О»

№	Операція	Значення нотації великого «О»	Коментар
1	Ввод логіну та паролю та перевірка валідності	O(1)	Перевірка валідності логіну та паролю відбувається один раз при виконанні алгоритму. Час виконання цієї операції не залежить від кількості вхідних даних.
2	Ініціалізація трьох вимірного динамічного масиву	O(1)	Ініціалізація масиву відбувається один раз при виконанні коду. Час виконання цієї операції не залежить

			від кількості вхідних даних.
3	Читання еталонних характеристик ЧУК користувача	O(1)	Отримання та читання еталоних даних з БД відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.
4	Визначення події натиску на клавіатурі	O(1)	Визначення події натиску відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.
5	Визначення виду події KeyUp або KeyDown	O(1)	Визначення виду події відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.
6	Визначення клавіші з якої відбулася подія	O(1)	Визначення клавіші, яка створила подію відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.
7	Додавання події до ініційованого масиву	O(1)	Операція додавання елементу до масиву відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.
8	Обчислення ЧУК та ЧУК з накладеннями	$O(n^3)$	Потрійний цикл для ітерації усіх значень та їх обчислення, а також визначення наявності накладень між натисками.
9	Отримання еталонного зразку з БД	O(1)	Запит до БД відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.
10	Порівняння зразків	$O(n^2)$	Подвійний цикл, який здійснює порівняння кожного значення еталонного зразку та отриманого в результаті процесу входу до системи.
11	Отримання результату - автентифікація успішна/неуспішна	O(1)	Отримання результату відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.

Значення нотації великого «О» для алгоритму розпізнавання [9] дорівнює формула Х.

$$\boldsymbol{f_1 = O(1), f_2 = O(n^2), f_3 = O(n^3) \rightarrow f_1 + f_2 + f_3 = O(\max(1, n^2, n^3)) = n^3} \quad (3.4)$$

Таблиця 3.4 – Операції алгоритму розпізнавання запропонованого автором даної роботи та значення нотації великого «О»

№	Операція	Значення нотації великого «О»	Коментар
1	Ввод логіну та паролю та перевірка валідності	O(1)	Перевірка валідності логіну та паролю відбувається один раз при виконанні алгоритму. Час виконання цієї операції не залежить від кількості вхідних даних.
2	Формування об’єкту події натиску на клавіатуру	O(1)	Формування події натиску відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.
3	Додання об’єкту до фінального масиву	O(1)	Операція додання елементу до масиву відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.
4	Перевірка повноти набраного кодового слова (тексту)	O(1)	Перевірка повноти набраного тексту відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.
5	Передача фінального масиву на серверну частину	O(1)	Відправлення фінального масиву до серверу відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.
6	Обчислення ЧУК та ЧУК з накладеннями	$O(n^2)$	Подвійний цикл для ітерації усіх значень та їх обчислення, а також визначення наявності накладень між натисками.
7	Отримання еталону для поточного користувача з БД	O(1)	Запит до БД відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.
8	Порівняння зразків (обчислення відстані	$O(n^2)$	Подвійний цикл для обчислення відстані Евкліда між ЧУК та ЧУК з

	Евкліда для зразків клавіатурного почерку)		накладеннями.
9	Отримання результату - автентифікація успішна/неуспішна	O(1)	Отримання результату відбувається один раз при виконанні коду. Час виконання цієї операції не залежить від кількості вхідних даних.

Значення нотації великого «О» для алгоритму розпізнавання, який було запропоновано використовувати, дорівнює формула 3.5:

$$\boldsymbol{f_1 = O(1), f_2 = O(n^2), f_3 = O(n^2) \rightarrow f_1 + f_2 + f_3 = O(\max(1, n^2, n^2)) = n^3} \quad (3.5)$$

В таблиці 3.5 показано яка кількість елементарних операцій відбудеться при застосуванні різних алгоритмів.

Таблиця 3.5 – Кількість елементарних операцій при різних вхідних значеннях

Кількість вхідних параметрів (n)	n^2	n^3
1	1	1
2	4	8
4	16	64
8	64	512
16	256	4096
32	1024	32768

Отже, алгоритм запропонований автором даної роботи дозволяє зменшити кількість елементарних операцій при виконані коду з різною кількістю вхідних даних, що свідчить про підвищення ефективності та швидкості виконання.

В даному підрозділі були оцінена ефективність та швидкість виконання алгоритмів навчання та розпізнавання за допомогою нотації великого «О», що дозволяє оцінити швидкість виконання коду за допомогою алгебраїчних виразів.

Результати порівняння показали, що видозмінений алгоритм запропонований автором цієї роботи має більшу швидкість обробки, виконуючи меншу кількість елементарних операцій при заданій кількості вхідних параметрів.

3.3 Проектування архітектури

Відповідно до необхідних потреб роботи розробки, автором було запропоновано використовувати клієнт-серверну архітектуру ПЗ, що зображена на рис. 3.2:

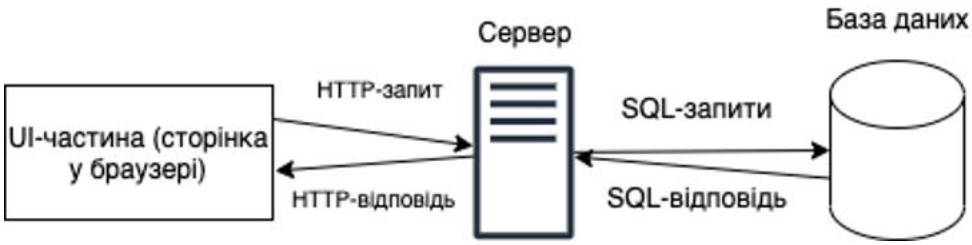


Рисунок. 3.2 – Клієнт-серверна архітектура ПЗ

Автор даної роботи відзначає фактори, що обумовлюють вибір такої архітектури:

- не потрібно дублювати програмний код на клієнтській стороні;
- так як усі розрахунки виконує сервер, то апаратні вимоги до клієнта мінімальні;
- усі дані зберігаються на сервері, який більш захищений, ніж клієнтська сторона.

Як видно з діаграми зображеної на рис 3.2, користувачеві не потрібно встановлювати додаткове ПЗ, достатньо перейти на потрібну сторінку у браузері.

Сервер приймає та відповідає на HTTP-запити, які надсилаються з клієнта і в залежності від логіки здійснює SQL-запити до бази даних (БД). Сервер виконує основні обчислення на основі даних надісланих с інтерфейсу користувача (ІК).

В даному підрозділі було описано основну архітектуру ПЗ на високому рівні функціонування. Охарактеризовано, яким чином взаємодіють компоненти між собою: клієнт надсилає HTTP-запити, сервер виконує обчислення на основі надісланих даних та зберігає результати в базі даних.

3.4 Вибір алгоритму

Наступним кроком проектування системи є розробка алгоритму. Беручи до уваги огляд існуючих алгоритмів розпізнавання, які були проаналізовані у розділі 2, можна зробити висновок, що ймовірнісно-статистичний метод є найбільш підходящим для системи що проектується.

Методу на основі нейронних мереж потрібно більше часу для навчання. Гістограмний метод використовує статистичні дані часових інтервалів без прив'язки до конкретних клавіш і це може негативно відобразитись на точності алгоритму в цілому. Потрібно враховувати, що кожен користувач має різну швидкість набору при використанні різних розкладок клавіатури.

Варто додати, що ймовірнісно-статистичний метод ґрунтується на простій для реалізації математичній моделі.

На основі вищесказаного було прийнято рішення використовувати ймовірнісно-статистичний метод розпізнання клавіатурного почерку, як простий та надійний у реалізації.

Виходячи з вище зазначеного, система має два етапи роботи:

- навчання;
- розпізнавання.

На клієнтський частині відбувається перша реєстрація користувача за допомогою логіну та паролю. Після цього користувач повинен ввести власне кодове слово (текст) по якому в подальшому буде відбуватися розпізнавання.

Маючи кодове слово (текст) користувач має можливість додати зразки клавіатурного почерку шляхом введення цього слова (тексту) декілька разів.

Клавіатурний почерк додається для поточного користувача в БД. Якщо кількість зразків клавіатурних почерків на момент додання нового перевищує 15, то система автоматично видаляє найстаріший зразок для поточного користувача. Такий підхід дозволяє враховувати коливання характеристик клавіатурного почерку одного того ж самого користувача в різні пори доби та в різних станах. Наприклад, можна відзначити той факт, що в стані втомленості або стресу людина набирає текст по-різному.

Блок-схеми алгоритмів етапів навчання та розпізнавання системи, що розроблена автором представлена на рис. 3.2 та 3.3 відповідно.

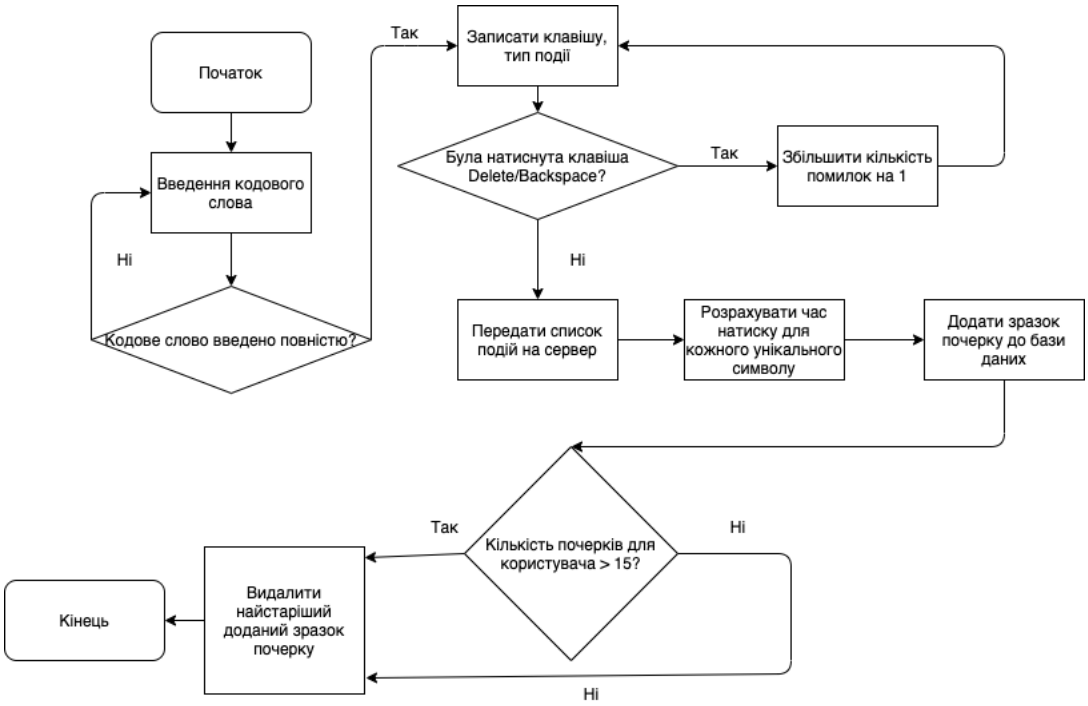


Рисунок 3.2 - Блок-схема алгоритму етапу-навчання

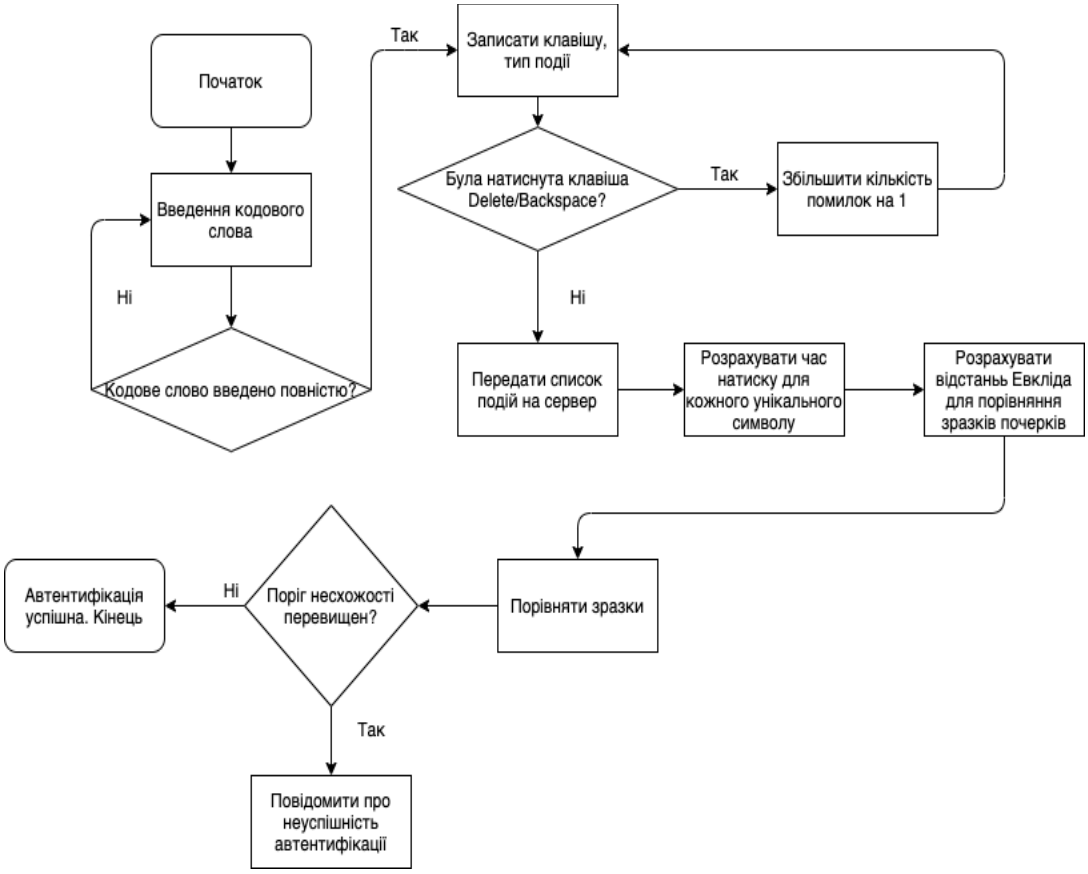


Рисунок 3.3 - Блок-схема алгоритму етапу-розпізнавання

Як видно з наданих блок-схем, початок алгоритму навчання та алгоритму розпізнавання практично ідентичні, крім другої частини, де відбувається безпосередньо розпізнавання та порівняння зразків почерку.

Результатом досконалого навчання є успішний процес додання зразка почерку до БД, а процесу розпізнавання – логін до системи після того як користувач ввів свій логін та правильний пароль. Якщо зразок введеного кодового слова (тексту) перевищує поріг схожості, то система повідомляє про помилку автентифікації та доступ до головної сторінки стає заблокованим.

3.5 Інструменти реалізації

Для розробки веб-застосунку було обрано наступні мови програмування та технології:

- інтерфейс користувача – Javascript, Vue JS 2.0 Framework;

- серверна частина – C# 7, .NET Core 2.2;
- база даних – MySQL 8+.

JavaScript — динамічна, об'єктно-орієнтована прототипна мова програмування. Найчастіше використовується для створення сценаріїв вебсторінок, що надає можливість на боці клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд вебсторінки [24].

C# та фреймворк .NET Core ідеально підходить для розробки серверної частини будь-якого веб-застосунку. Мова C# проста, безпечна в контексті типів та обєктно-орієнтовна. Платформа .NET забезпечує швидкість у розробці застосунків, при цьому зберігає за собою елегантність та простоту [25].

Vue JS 2.0 – один з найпрогресивніших Javascript фреймворків, який дозволяє будувати інтерфейси користувача будь-якої складності за допомогою вже готових компонетів [26].

MySQL — вільна система керування реляційними базами даних. MySQL був розроблений компанією «TcX» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування [27].

В даному підрозділі були описані інструменти за допомогою яких буде реалізовано ПЗ у веб-середовищі:

- Javascript та Vue JS 2.0 – для розробки клієнтської частини;
- C# та .NET Core – для розробки серверної частини;
- MySQL – для зберігання даних.

Для кожної технології була наведена коротке означення та список задач, які здатна вирішити кожна мова програмування.

3.6 Перехоплення подій клавіатури

Для аналізу та розрахування часу натиску клавіш потрібно організувати перехоплення подій клавіатури. В даному застосунку користувач має можливість ввести кодове слово та передати дані для подальшої обробки на сервер.

У мові Javascript та фрейворку Vue JS 2.0 є механізми, які дозволяють перехоплювати події клавіатури на певному елементі сторінки. Це події keyup, keydown та keypress, які дозволяють виконувати будь-які дії, коли користувач набирає текст всередині даного елемента. При перехоплені кожної події, програма додає об’єкт з інформацією про натиснуту клавішу, де вказується клавіша та час безпосереднього натиску.

Після того, користувач натискає відповідну клавішу, яка відправляє створений масив даних на сервер на подальшу обробку.

В даному підрозділі було коротко описано, за допомогою чого організоване перехоплення подій клавіатури для аналізу клавіатурного почерку.

3.7 Опис клієнтського застосунку

Інтерфейс користувача призначений для спрощеної взаємодії програми та людини. В даному випадку інтерфейс користувача призначений для збору даних про клавіатурний почерк користувача. Частина коду клієнтської частини винесена в Додаток Б.

Розроблений автором даної роботи клієнтський застосунок, складається з таких частин (сторінок):

- сторінка логіну та реєстрації;
- сторінка профілю;
- сторінка навчання (додання нового зразку почерку);
- сторінка активних зразків почерків у табличному вигляді;

- сторінка активних зразків почерків у графічному вигляді.

Більш докладний опис, склад та призначення сторінок буде розглянутий у подальших підрозділах.

3.7.1 Опис сторінок логіну та реєстрації

Сторінка логіну та реєстрації призначена відповідно для логіну та реєстрації користувача шляхом введення унікального логіну та паролю. Як видно з рис. 3.4 та 3.5, вищезгадані форми майже ідентичні за винятком деяких елементів.

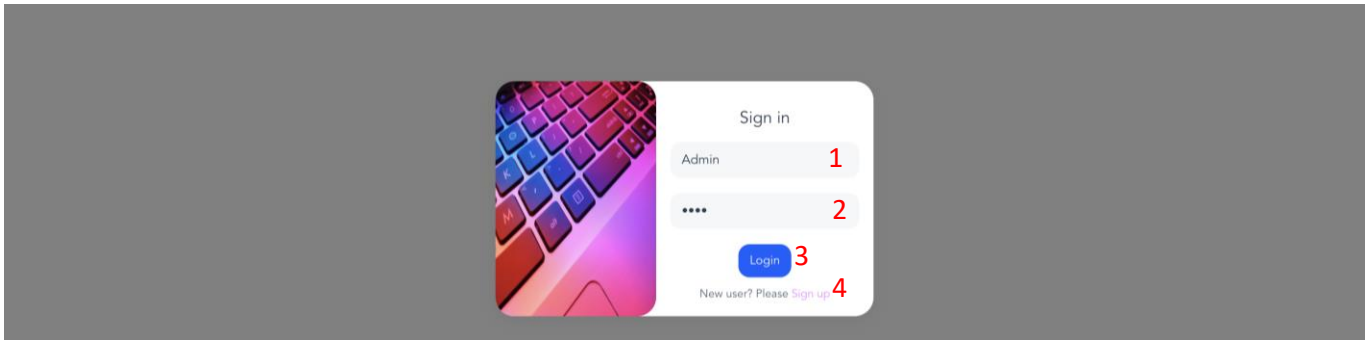


Рисунок. 3.4 - Форма логіну користувача

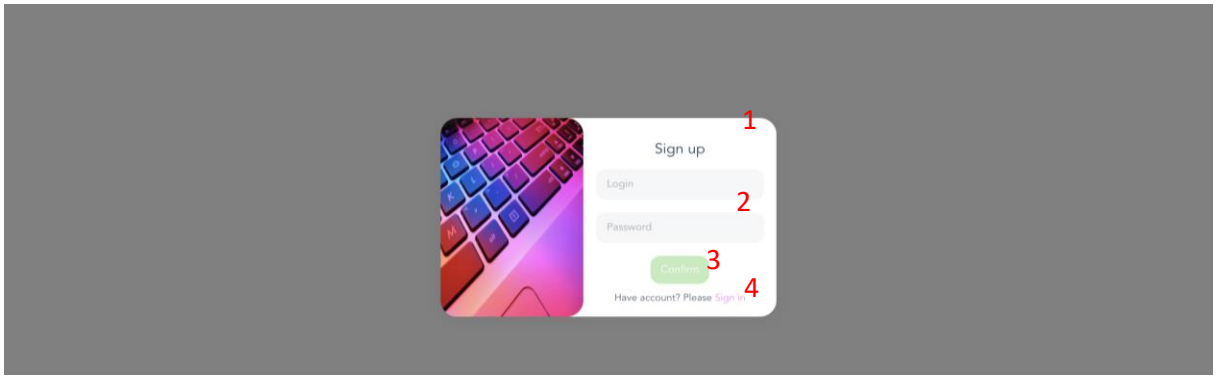


Рис. 3.5 - Форма реєстрації користувача

Склад форми:

1 – поле для вводу логіну;

2 – поле для вводу паролю;

3 - кнопка для відправки даних після введення логіну та паролю;

4 - кнопка для переключення між формами логіну та реєстрації.

3.7.2 Опис сторінки профілю

Результатом логіну до системи є перехід до сторінки профілю користувача. Склад сторінки зображено на рис. 3.6:

1- меню веб-застосунку;

2- кнопка виходу з системи;

3 -картка профілю користувача;

4- поточний логін користувача;

5- статус активованого розпізнавання по клавіатурному почерку;

6 -форма для додавання/зміни кодового слова;

7 - кнопка збереження кодового слова;

8 - кнопка для очищення форми для введення кодового слова.

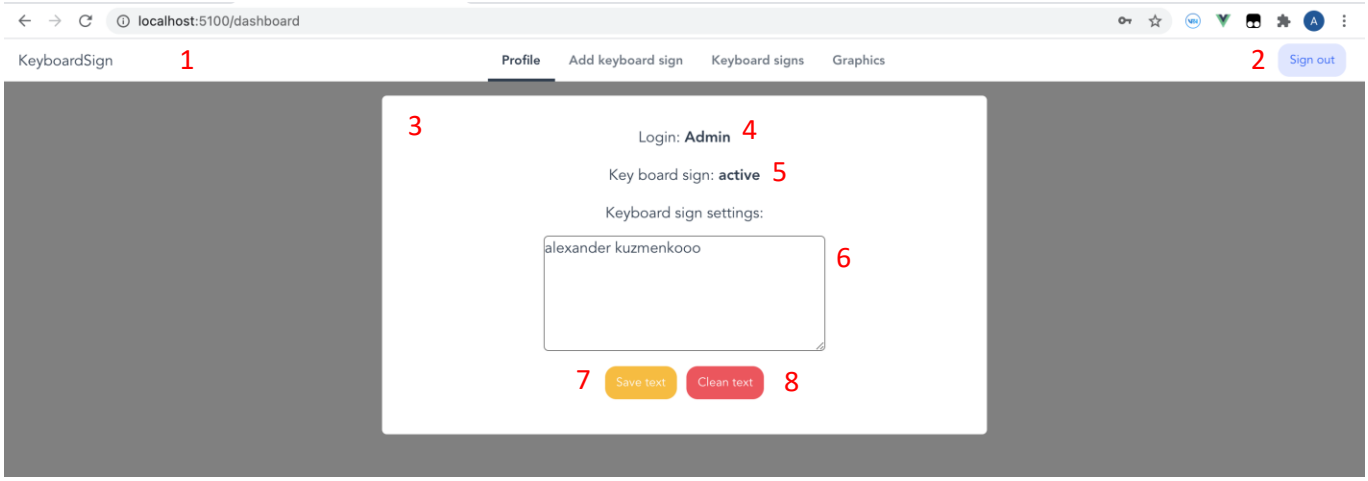


Рисунок 3.6 - Профіль користувача

Сторінка призначена для відображення інформації щодо поточного користувача та додання або редагування кодового слова (тексту) системи. За допомогою меню можливо переключатися між сторінками веб-застосунку.

3.7.3 Опис сторінки навчання (додання зразку почерку)

Сторінка навчання призначена для додання нового зразку почерку до системи шляхом введення кодового слова та відправки даних на сервер для подальшої обробки та збереження в БД.

Склад даної сторінки рис. 3.7 та 3.8 відповідно:

1 – картка стартового стану;

2 – кнопка для зміни режиму;

3 – карта режиму додання почерку;

4 – інформативне повідомлення;

5 – кодове слово, яке потрібно ввести;

6 – форма для вводу кодового слова;;

7 – кнопка збереження зразку почерку

8 – кнопка для очищення форми для вводу;

9 – кнопка відміни (переключення у стартове положення).

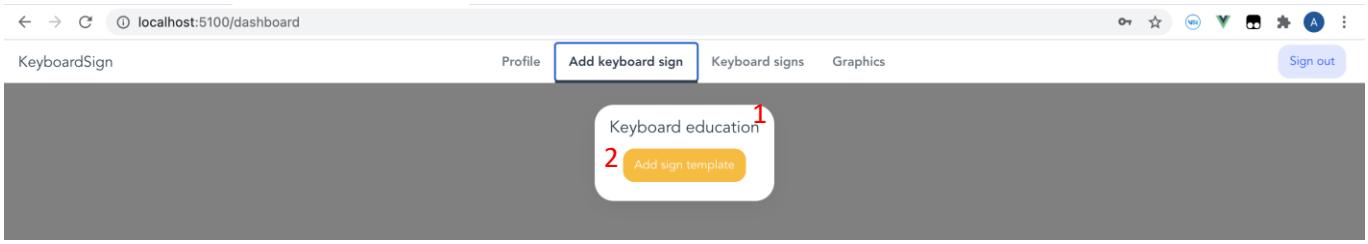


Рисунок 3.7 - Початковий стан форми

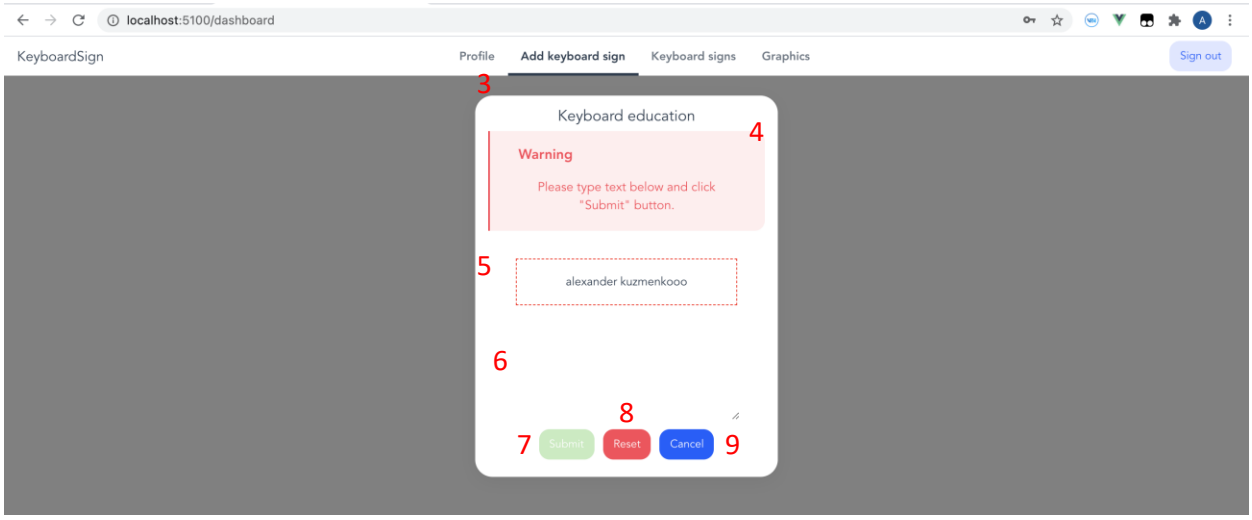


Рисунок 3.8 - Стан форми для додання зразку

Форма має інтуїтивно зрозумілий інтерфейс для швидкого додавання зразку клавіатурного почерку.

3.7.4 Опис сторінки активних почерків у табличному вигляді

Сторінка призначена для відображення даних по кожному активному клавіатурному почерку доданому користувачем через призначену для цього форму.

Користувачеві достатньо обрати будь-який зразок і система автоматично відобразить дані по обраному зразку. Склад форм зображено рис. 3.9 та рис. 3.10 відповідно:

1 – карта стартового положення;

2 – список активних зразків;

3 – обраний зразок;

4 – середній час паузи між натисканнями клавіш;

5 – кількість помилок;

6 – код клавіші в ASCII;

7 – назва клавіши;

8 – середній час утримання клавіши з накладенням;

9 – середній час утримання клавіш без накладення.

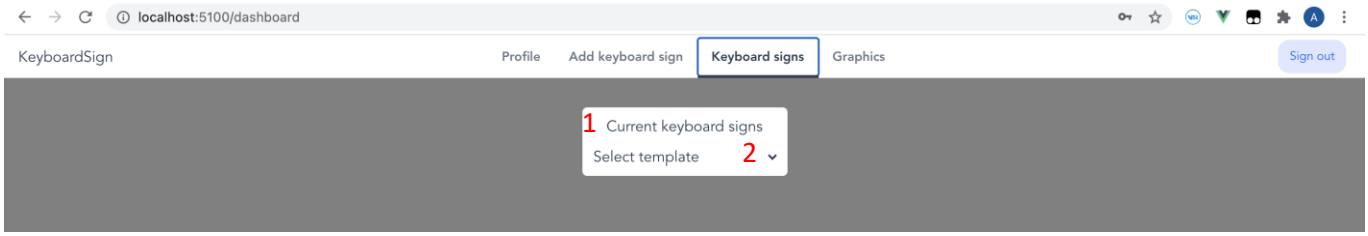


Рисунок 3.9 - Початковий стан форми

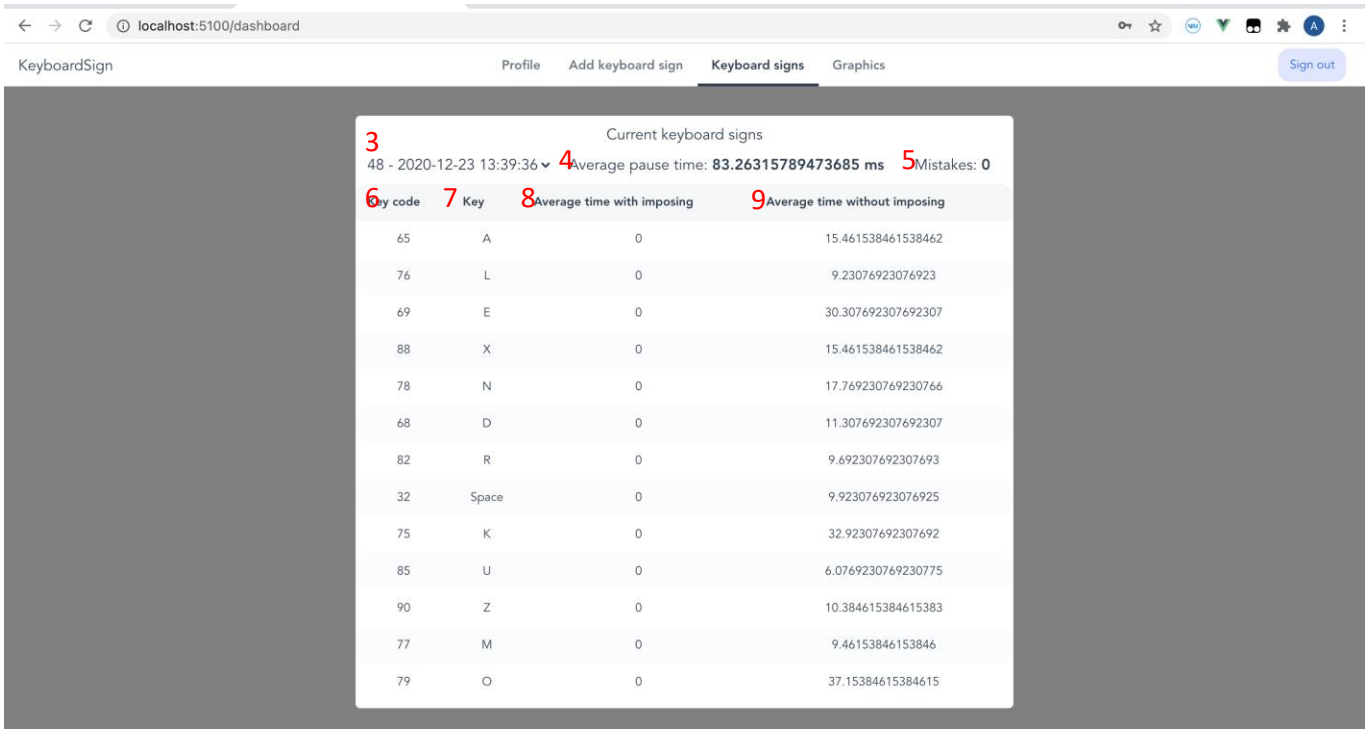


Рис. 3.10 - Табличне відображення інформації щодо обраного зразку поточного користувача

Варто зазначити, що список містить тільки 15 активних зразків клавіатурних почерків користувача. Значення часів утримання вказані в мілісекундах.

3.7.5 Опис сторінки активних почерків у графічному вигляді

Сторінка призначена для відображення інформації по активним зразкам клавіатурних почерків у графічному вигляді.

На даній сторінці зображено три графіка, які показують середній час паузи між натисками клавіш, середній час утримання клавіші з накладенням і середній час утримання клавіші без накладення рис. 3.11.

Рисунок 3.11 - Графічне відображення даних щодо усіх зразків поточного користувача

Аналогічно до сторінки, яка відображає дані по зразкам у табличному вигляді, тут зображені дані для 15 активних клавіатурних почерків.

3.8 Опис серверного застосунку

Серверна частина (зразок коду знаходиться в Додаток В) застосунку відповідає за обчислення даних, які надійшли з інтерфейсу користувача та збережені даних у БД.

Також вона виконує порівняння зразка почерку з еталонним, щоб визначити, що зразок клавіатурного почерку відповідає поточному користувачеві.

Діаграма класів зображена на рис. 3.12:

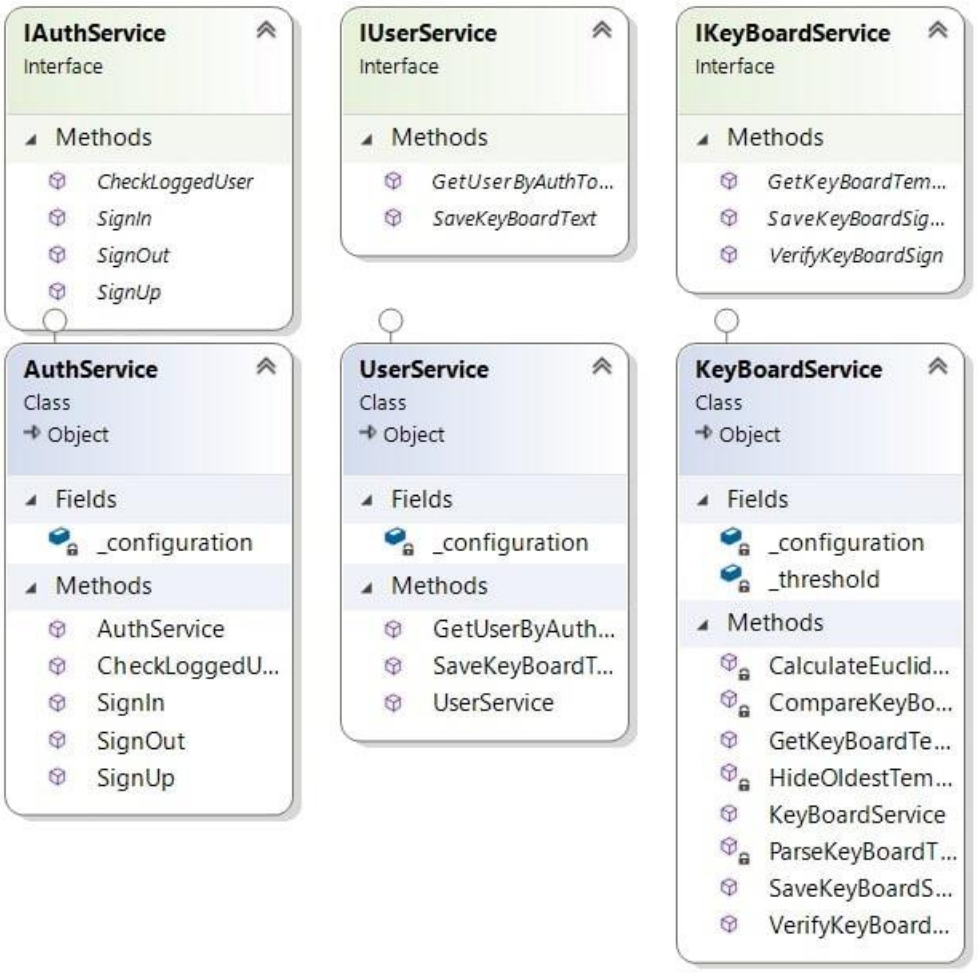


Рисунок 3.12 – Діаграма інтерфейсів та класів серверної частини застосунку

Діаграма містить у собі усі необхідні інтерфейси та класи, що реалізують їх, які відповідають за авторизацію користувача та оброку даних клавіатурного почерку надісланих з клієнтської частини.

3.9 Опис структури бази даних

База даних (скрипт зі створення бази даних винесений в Додаток Г) складається з однієї схеми, яка містить в собі дві таблиці: users та key_board_templates, які пов’язані зв’язком один-до-багатьох рис 3.12:

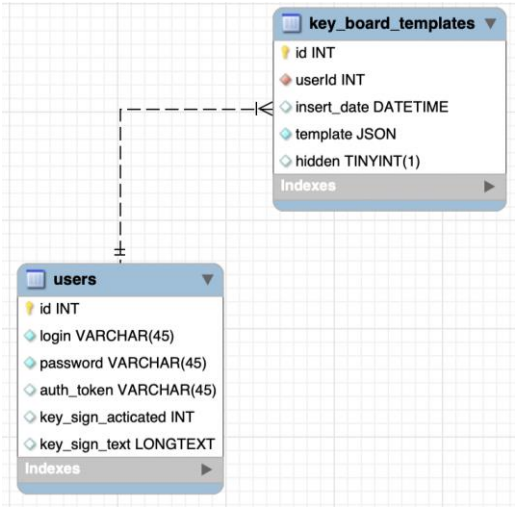


Рисунок 3.12 - Структура бази даних

Таблиця users зберігає дані про користувачів в системі, а key_board_templates – зразки клавіатурних почерків.

Автор представленої роботи пропонує наступну структуру таблиці users

- id – INT – унікальний номер запису в таблиці;
- login – VARCHAR(45) – логін користувача;
- password – VARCHAR(45) – пароль користувача;
- auth_token - VARCHAR(45) – авторизаційний токен;
- key_sign_activated – INT – статус активації клавіатурного розпізнання;
- key_sign_text – LONGTEXT - кодове слово (текст).

Авторизаційний токен – це згенерований випадковим чином рядок, який зберігається у localStorage браузера. При кожному запиті сервер порівнює токен, який зберігається у браузері та у БД. Таким чином система перевіряє чи поточний користувач має доступ до системи.

Крім того, автором даної роботи пропонується наступна структура таблиці key_board_templates:

- id – INT – унікальний номер запису в таблиці;
- userId – INT – унікальний ідентифікатор користувача з таблиці users;

- insert_date – DATETIME –дата і час збереження зразку клавіатурного почерку;
- template - JSON– об’єкт зразка клавіатурного почерку;
- hidden– INT – маркер, який показує чи активний зразок (0 – ні, 1 - так).

Отже, структура БД дозволяє зберігати усі необхідні дані, які потрібні веб-застосунку.

В даному підрозділі був наданий опис структури бази даних, кількість таблиць та полів, які зберігають різни типи даних.

3.10 Передача даних

Як було згадано раніше, передача даних між інтерфейсом користувача та сервером відбувається за допомогою HTTP-запитів. Усі необхідні класи конвертуються в JSON формат.

JSON формат надає універсальну структуру даних і зручний для організації взаємодії між різноманітними програмними модулями, в тому числі написаних на різних мовах програмування [28].

3.11 Алгоритм навчання

Сервер отримує дані від клієнта у вигляді масиву об’єктів, які в свою чергу містять інформацію щодо натиснутих клавіш (назва клавіші, код клавіші в ASCII, тип події та час виникнення події) рис. 3.13.

Автор роботи проаналізував, що існує три види подій: подія натиску клавіші, подія утримання клавіші та подія відпускання клавіші.

Після того, як сервер отримав дані, відбувається обчислення часу утримання клавіш шляхом віднімання часу події відпускання та часу події натиску клавіші. Якщо в натисках між клавішами присутні накладення, то система обчислює їх окремо.

Наступним кроком є те, що система вираховує середні значення часу утримання для кожної унікального символу в кодовому слові і формує новий список. Додатково обраховуються середні значення часу пауз між натисканнями і кількість помилок, які були допущені під час набору.

```
[
  {
    "code":"KeyA",
    "key":65,
    "type":1,
    "timeStamp":"2021-01-08T14:08:42.433Z"
  },
  {
    "code":"KeyA",
    "key":97,
    "type":3,
    "timeStamp":"2021-01-08T14:08:42.434Z"
  },
  {
    "code":"KeyA",
    "key":65,
    "type":2,
    "timeStamp":"2021-01-08T14:08:42.495Z"
  },
  {
    "code":"KeyL",
    "key":76,
    "type":1,
    "timeStamp":"2021-01-08T14:08:42.538Z"
  },
  {
    "code":"KeyL",
    "key":108,
    "type":3,
    "timeStamp":"2021-01-08T14:08:42.539Z"
  },
  {
    "code":"KeyL",
    "key":76,
    "type":2,
    "timeStamp":"2021-01-08T14:08:42.610Z"
  },
  {
```


Рисунок 3.13 - Приклад масиву який отримує сервер від клієнтської частини

Кожен об’єкт унікального символу з усіма середніми значеннями додається в масив об’єктів клавіатурного почерку, який конвертується в JSON формат і зберігається в базі даних. Перед збереженням система видаляє найстаріший доданий зразок для користувача і додає новий.

В даному підрозділі було описано алгоритм навчання системи, яким чином дані формуються та відправляються з клієнта на сервер та які операції робить серверна частина перед збереженням зразку клавіатурного почерку до БД.

3.12 Алгоритм розпізнавання

Після того, як користувач набрав кодове слово і відправив його на сервер для перевірки, сервер формує той самий об’єкт, що і при етапі навчання, тобто вираховує середні значення для кожного унікального символу. Цей об’єкт буде порівнюватися з еталонним об’єктом.

Критерієм порівняння двох зразків клавіатурного почерку є поріг несхожості. Поріг несхожості – це стале число, яке показує чи можна вважати зразок схожим на еталонний. В даному випадку поріг несхожості становить 15 відсотків відповідно до дослідження [9].

При визначенні несхожості зразків використовуються наступні умови (формули 3.6 -3.8) [9]:

$$\boldsymbol{d < t * a_p} \tag{3.6}$$

$$\boldsymbol{|m - a_m| < t * \ a_m} \tag{3.7}$$

$$\boldsymbol{\left|p - \ a_p\right| < t * a_p} \tag{3.8}$$

де d – евклідова відстань між послідовностями ЧУК (для двох зразків клавіатурного почерку);

t – порогове значення, яке становить 15 відсотків;

a_p- середній час паузи між натисканнями еталонного зразку;

m – кількість помилок в зразку, який було введено під час логіну до системи;

a_m – середнє значення помилок еталонного зразку;

p – середній часовий діапазон пауз між натисканнями зразка, який було введено під час логіну.

При одночасному виконанні вищезгаданих умов зразки почерків вважаються співпадаючими, тобто належать одному і тому ж користувачеві, в іншому випадку система видає помилку про те, що зразки не співпадають.

Висновки за розділом 3

В даному розділі було розкрито питання щодо архітектури ПЗ, яке здійснює головні процеси, а саме передачу, аналіз та збереження даних.

Дане ПЗ має наступні функціональні можливості:

- логін користувача;
- реєстрація користувача;
- внесення кодового слова;
- внесення зразків клавіатурного почерку по кодовому слову;
- порівняння зразків почерку.

ПЗ має клієнт-серверну архітектуру у веб середовищі, що спрощує використання даного застосунку, без встановлення додаткових програмних пакетів.

Сервер використовує ймовірнісно-статистичний метод, який найпростіший в реалізації та не потребує складних математичних обчислювань.

Для реалізації обраної архітектури та алгоритму було обрано найсучасніші мови програмування та технології, а саме Javascript, Vue JS 2.0 та .NET Core 2.2.

Обрані технології забезпечують простоту, швидкість, гнучкість у розробці та реалізації поставлених задач.

Було надано конкретний опис усіх форм інтерфейсу користувача, а саме:

- сторінок логіну та реєстрації;
- сторінки профілю користувача;
- сторінки навчання системи;
- сторінок табличного та графічного відображені інформації.

Аналогічно до інтерфейсу користувача, було надано опис серверної частини в якій було наведена UML-діаграма зі структурою класів серверного коду.

Структура бази даних відіграє важливу роль у кожній системі, тому було також надана структура таблиць, які зберігають усю необхідну інформацію, а саме інформацію про користувачів у таблиці users, а зразки клавіатурного почерку – key_board_templates. Додатково було наведено з яких полів складається кожна таблиця.

Наведені факти, щодо передачі даних між частинами веб-застосунку, у якому форматі передаються дані та яким чином зберігаються у базі даних.

В кінці розділу були докладно описані алгоритми навчання та розпізнавання клавіатурного почерку в рамках обраної архітектури та математичного алгоритму.

Були наведені визначення порогу схожості зразків, а також математичні формули та критерії за допомогою яких відбувається порівняння наданого почерку та еталонного.

РОЗДІЛ 4
ПРАКТИЧНІ РЕКОМЕНДАЦІЇ ЗАСТОСУВАННЯ ДАНИХ КЛАВІАТУРНОГО ПОЧЕРКУ КОРИСТУВАЧА

4.1 Аналіз результатів тестування програмного забезпечення

Тестування ПЗ та обраної методики проводилося у декілька етапів.

На першому етапі дослідження автором роботи були досліджені клавіатурні почерки шести користувачів, які зібрані за допомогою даного ПЗ. Для кожного зразка почерку було підраховано ЧУК з накладеннями, ЧУК без накладенньь, кількість пауз та кількість помилок.

На рис. 4.1 представлено зміна ЧУК без накладень в залежності від натиснутої клавіші (більш докладна таблиця з усіма значеннями ЧУК винесена в Додаток Е):

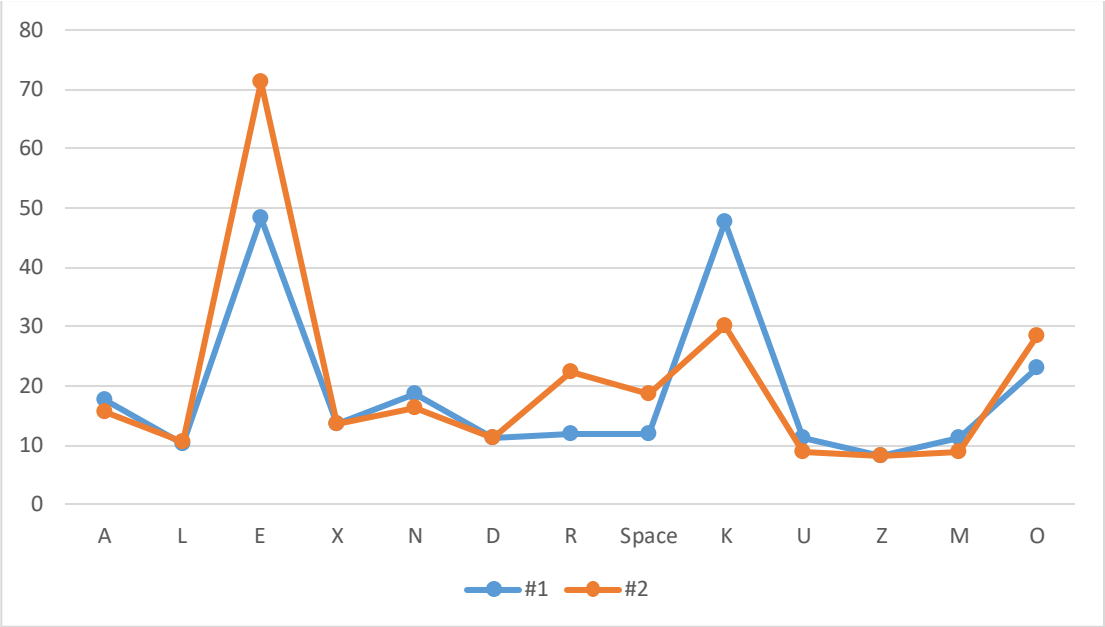


Рисунок 4.1 - ЧУК без накладень різних зразків для одного користувача

Час утримання клавіш для першого зразка коливаються від 8,23 мс до 48,38 мс, а для другого зразка – 8,08 мс до 71,15 мс. При цьому середнє значення часу

утримання клавіш для першого зразка становить – 18,92 мс, а для другого 19,35 мс. Таким чином середній час утримання клавіш є практично однаковим для різних зразків почерку одного і того ж користувача. Найбільша різниця між натисками клавіш «K» та «E».

На рисунку 4.2 зображено графік отриманий автором роботи у ході дослідження зміни ЧУК з накладенням в залежності від натиснутої клавіші для одного й того ж самого користувача.

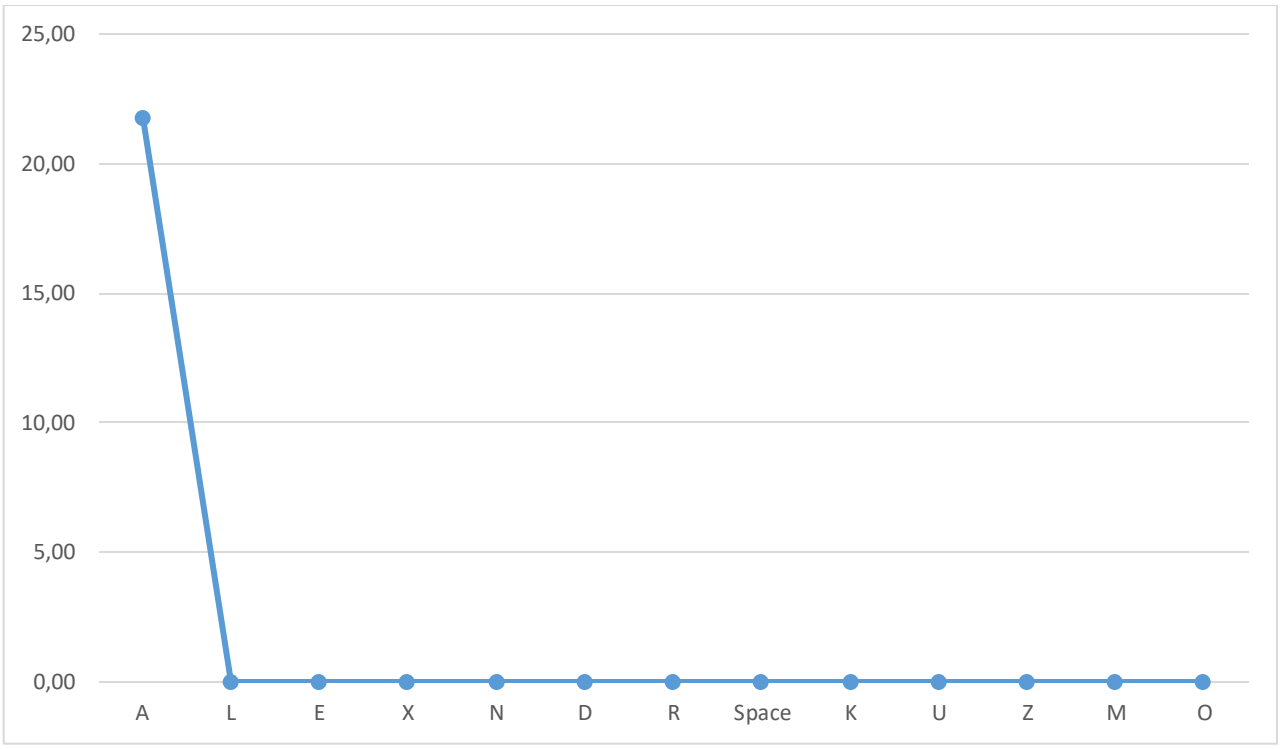


Рисунок 4.2 - ЧУК з накладеннями для одного користувача

З даного графіка можна зробити висновок, що користувач відпустив клавішу з літерою «A» пізніше, ніж натиснув наступну. Це і є накладення між натисками клавіш. Для інших клавіш накладень між натисками для даного зразку клавіатурного почерку не було, тобто кодове слова набиралося рівномірно з середньою швидкістю.

Середнє значення пауз між натисками клавіш для клавіатурного почерку одного і того ж користувача зображено на рис. 4.3:

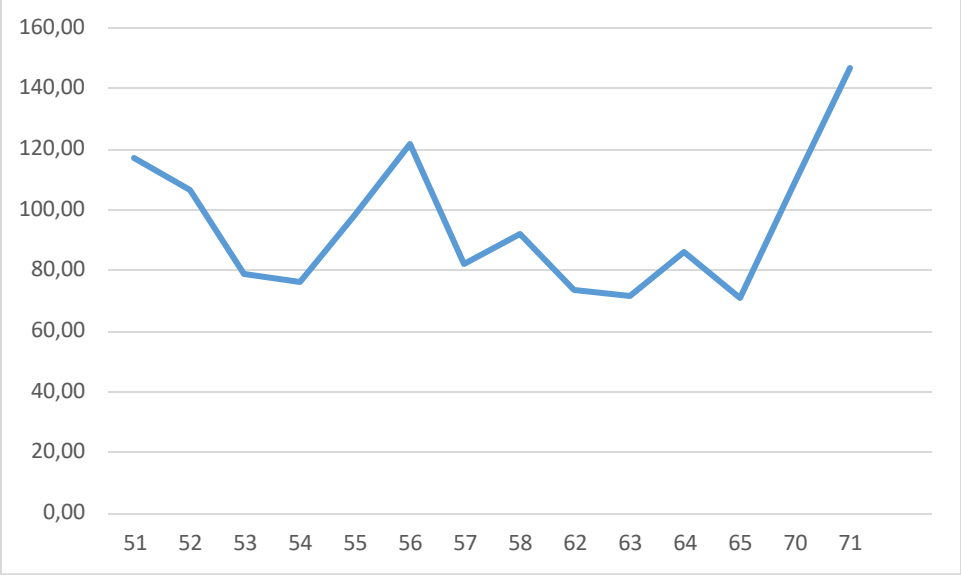


Рисунок 4.3 - Середнє значення пауз для зразків почерків одного користувача

Середні значення пауз коливаються від 70,84 мс до 147,17 мс. Це говорить про те, що паузи між натисканням клавіш є майже однаковими для одного і того ж користувача.

На рисунку 4.4 показано середнє значення ЧУК для декількох зразків клавіатурного почерку одного і того ж користувача:

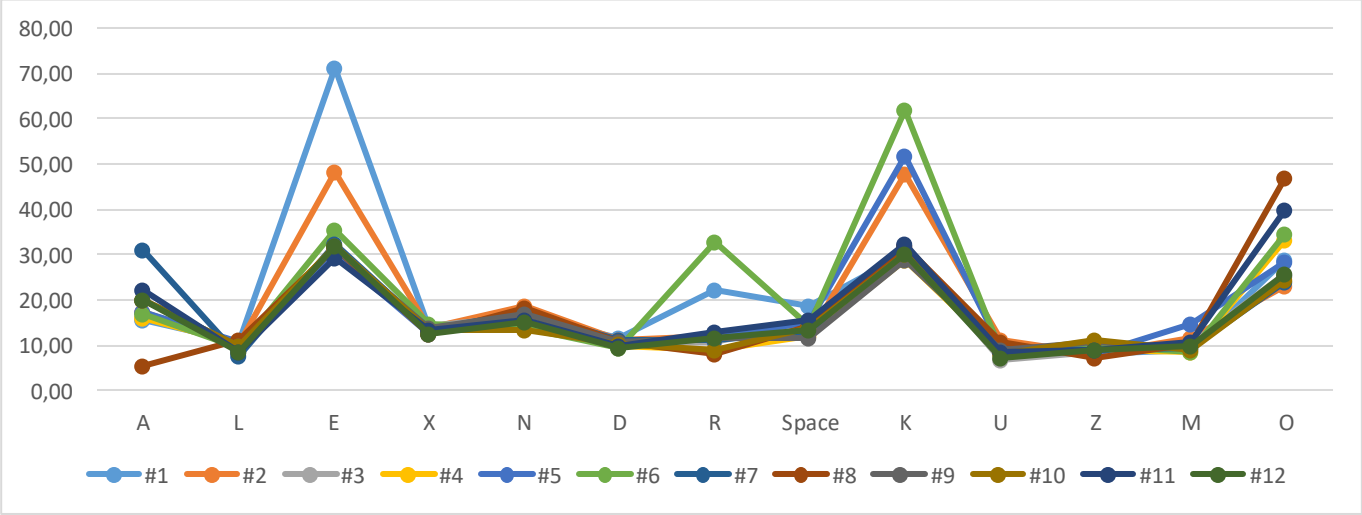


Рисунок 4.4 - Середнє значення ЧУК без накладень для різних зразків

Тепер варто порівняти клавіатурні почерки при введені однакового кодового слова (тексту) різними користувачами.

Для порівняння буде використовуватися середнє значення пауз між натисками, ЧУК з накладеннями та без накладень рис. 4.5.

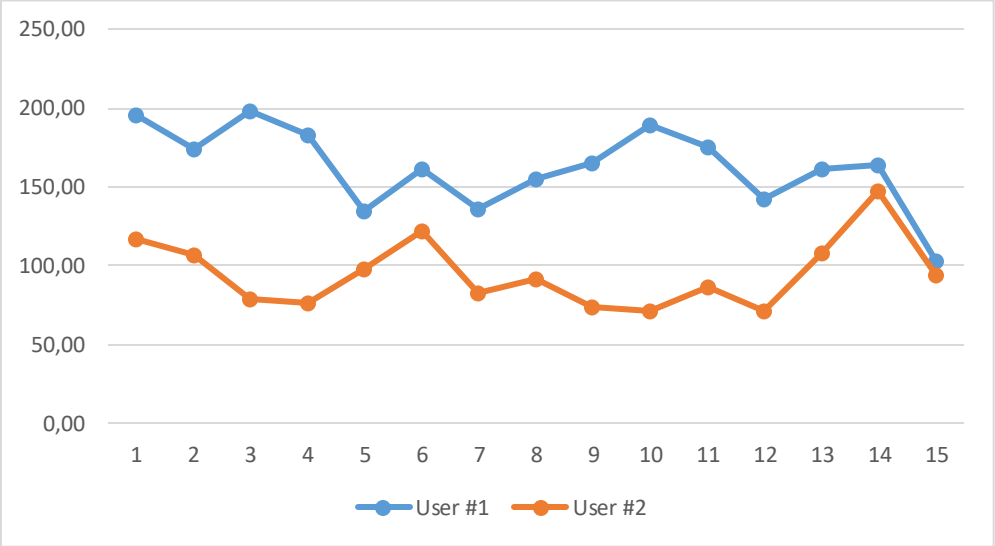


Рисунок 4.5 Порівняння середніх пауз при однаковому кодовому слові для різних користувачів

Результати аналізу середнього часу пауз зведені у табл. 4.1. В цій таблиці вказані максимальні та мінімальні значення пауз між натисками та середні значення пауз для усіх зразків почерку для кожного користувача.

Таблиця 4.1 – Порівняння часу пауз між натисками клавіш для різних користувачів

	t_{max}	t_{min}	$t_{average}$
Користувач 1,зразок1	198,42 мс	102,58 мс	162,63 мс
Користувач 2,зразок 1	147,17 мс	71,42 мс	94,95 мс

З даних зведеної таблиці можна зробити висновок, що зразки почерків мають суттєву різниці як у діапазоні пауз між натисками клавіш, та і у середньому значені. Таку ж саму різницю можна побачити на рис 4.6.

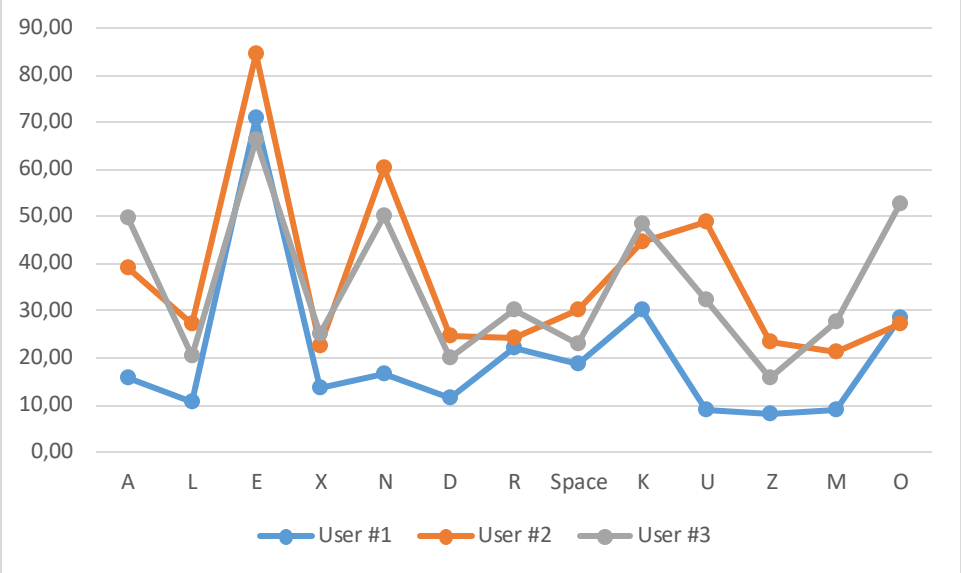


Рисунок 4.6 - Середнє значення ЧУК при однаковому кодовому слові для різних користувачів

В даному випадку ЧУК для першого користувача варіюються від 8,08 мс до 71,15 мс, для другого користувача – від 21,08 мс до 84,38 мс, а для третього користувача від 19,92 мс до 66,08 мс. Середнє значення ЧУК для усіх клавіш першого користувача становить – 20,36 мс, для другого – 36,72 мс, для третього – 45,49 мс.

В порівнянні зразків почерків для одного користувача різниця між ЧУК була менше 1 мс, в даному випадку різниця більше ніж 8-9 мс.

Мінімальні відмінності спостерігаються для літери «E» та «R» 10 - 15 мс, а максимальні для літер – 30 - 40 мс. В середньому різниця становить 30,85 мс.

Отримані результати дозволяють зробити висновок, що клавіатурний почерк різних людей сильно відрізняється, тобто є можливість проводити автентифікацію за допомогою математичної моделі.

Кожній людині характерний свій індивідуальний клавіатурний почерк, а почерки різних людей мають суттєві відмінності.

Підводячи підсумок, потрібно сказати, що автентифікацію на основі аналізу клавіатурного почерку можна вважати ефективною і використовувати як механізм

захисту в ІС, тому що характеристики клавіатурного почерку кожної людини суттєві відрізняються, що каже про їх унікальність та незмінність з часом.

В даному підрозділі був проведений аналіз зразків клавіатурного почерку, які були зібрані за допомогою програмного забезпечення, яке було розроблене та реалізоване автором роботи. Основним призначенням ПЗ є розпізнавання варіантів клавіатурного почерку. Дані були представлені у вигляді графіків та таблиць, які наочно відображають суттєву різницю між ЧУК різних користувачів при наборі кодового слова.

4.2 Практичні рекомендації застосування результатів аналізу клавіатурного почерку

Під час розробки та тестування ПЗ (веб-застосунку) було визначено, які дані потрібно використовувати для автентифікації користувача по клавіатурному почерку: достатньо розглядати часові інтервали між натисками клавіш, часові інтервали утримання клавіш, кількість помилок та присутність накладень.

Однак, вищесказані характеристики залежать від багатьох факторів, наприклад один і той же користувач може набирати текст з різною швидкістю в залежності від часу доби та ступеня втомленості. Зміна клавіатури також впливає на швидкість набору символів.

Також варто згадати про основну проблему використання автентифікації по клавіатурному почерку – почерк виробляється у людей, які регулярно працюють за комп'ютером, тобто даний метод не можна використовувати для новачків.

Вплив даних факторів суттєво знижується в процесі автентифікації, в якій використовується набір відомого або невідомого ключового слова (тексту). В такому випадку ми порівнюємо результати не тільки з еталонним зразком, а й з різницею між інтервалами часу при наборі знайомого або незнайомого кодового слова (тексту).

Такий спосіб дозволяє вірно ідентифікувати користувача, не дивлячись на те втомленість або інші психофізичні фактори, оскільки знайому фразу користувач

завжди буде набирати трохи швидше незнайомої. Якщо користувач є зловмисником, то всі фрази для нього будуть незнайомі і відмінність в інтервалах буде присутня.

Від зміни клавіатури залежать значення інтервалів між натиском на клавіші, і це практично не впливає на час утримання клавіш. Тому використання двох і більше характеристик зменшує вірогідність невірної ідентифікації.

Якщо вищесказані фактори будуть враховані, то ПЗ буде мати можливість автентифікувати користувачів з найвищою точністю.

Практично дані клавіатурного почерку можуть використовуватися у будь-якому веб-застосунку, де потрібно ідентифікувати користувача. В таких застосунках потрібно поєднувати парольний та біометричний захист, який ґрунтується на клавіатурному почерку, тому що такий захист не може виступати основним. Задача біометричної ідентифікації по клавіатурному почерку полягає в захисті інформаційної системи в ситуації компрометації паролю.

Дані клавіатурного почерку можна використовувати для скритного моніторингу за користувачами в системі, досліджувати поведінку та психофізичний стан тої чи іншої людини або потенційного зловмисника в точний момент часу. Використання даних скритного моніторингу дозволяє впровадити у системі додатковий шар безпеки, який може попередити несанкціонований доступ до інформації.

Отже, практичними рекомендаціями по використанню даних клавіатурного почерку є:

- використання даних, як біометричний спосіб автентифікації користувача у будь-якій програмі або веб-застосунку;
- використання даних після скритного моніторингу для визначення психофізичного стану користувача у конкретний момент часу, забезпечуючи додатковий шар безпеки від несанкціонованого доступу до інформації.

Застосування рекомендацій по використанню даних клавіатурного почерку дозволить впровадити потужний біометричний захист інформаційної системи від НСД та атак зловмисників.

Висновки за розділом 4

В даному розділі були наведені дані, які були отримані в ході тестування розробленого ПЗ. Аналіз отриманих результатів у табличному та графічному вигляді показав, що зразки клавіатурного почерку можуть відрізнятися, маючи при цьому унікальні характеристики, такі як час пауз між натисками клавіш, кількість накладень та частота появи однакових характеристик для одних тих самих клавіш.

Крім того, були надані основні рекомендації щодо використання цих даних для провадження додаткового механізму захисту ІС:

- користувач повинен бути в нормальному стані під час процесу навчання та ідентифікації;
- під час навчання системи та безпосереднього розпізнавання зразків бажано використовувати одну і ту ж саму клавіатуру.

Клавіатурний почерк може використовуватися для скритного моніторингу за користувачами в системі, досліджувати поведінку та психофізичний стан тої чи іншої людини або потенційного зловмисника в точний момент часу.

ВИСНОВОК

Ціллю роботи було створення ефективного ПЗ для автентифікації користувача в ІС за допомогою клавіатурного почерку.

Для досягнення поставленої цілі було проведений огляд існуючих біометричних засобів автентифікації користувачів, перераховані переваги та недоліки. Доведено, що парольні та атрибутні засоби автентифікації поступово стають історією. На їх заміну приходять біометричні засоби ідентифікації, використання яких можна поєднувати з парольними та атрибутними.

Була сфокусована увага до клавіатурного почерку як динамічної біометричної ознаки людини. Були наведені фактори, які враховуються для автентифікації користувачів по клавіатурному почерку (час утримання клавіші, час між натисками клавіш, час пауз, кількість помилок та накладень). Додатково були наведені переваги та недоліки використання клавіатурного почерку, а саме простота в реалізації, вплив клавіатури і психофізичного стану людини під час набору кодового слова (тексту).

Для побудови алгоритму були описані математичні способи (алгоритми), які можна використовувати для автентифікації користувача по клавіатурному почерку. Зокрема були розглянуті ймовірнісно-статистичний, гістограмний методи та метод на основі нейронних мереж.

На основі ймовірнісно-статистичного методу були створені алгоритми, які дозволяють системі обробляти зразки клавіатурного почерку різних операторів і порівнювати їх для автентифікації (алгоритми навчання та розпізнавання).

Після створення алгоритму було реалізоване ПЗ для веб-застосунку, який складався з клієнтської та серверної частини. Клієнтська частина відповідає за взаємодію з користувачем, збір інформації щодо клавіатурного почерку користувача та відправки у коректному форматі на серверну частину. Роль серверної частини полягає у обчисленнях даних зразку клавіатурного почерку, які надіслав користувач

з клієнтської частини, збереження зразку з необхідної додатковою інформацією та розпізнання зразку клавіатурного почерку з еталоном.

Для створення ПЗ використовувалися найактуальніші технології, які забезпечують гнучкість та швидкість у розробці.

Система була протестована з участю декількох користувачів. Аналіз та порівняння результатів різних користувачів у табличному та графічному вигляді довів ефективність використання клавіатурного почерку для автентифікації користувачів в інформаційних системах.

Реалізоване програмне забезпечення успішно виконує поставлені задачі, а саме навчання та розпізнавання варіантів клавіатурного почерку.

В кінці роботи були надані практичні рекомендації щодо використання даних клавіатурного почерку, а саме для біометричної ідентифікації користувачів та визначення стану людини за допомогою прихованого моніторингу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кульчицький О. С.; Грицюк В. В.; Зотова І. Г. Центр воєнно-стратегічних досліджень Національного університету оборони України імені Івана Черняховського, Київ. Аналіз існуючих підходів при ідентифікації і аутентифікації користувачів в інформаційно- телекомунікаційних системах.

2. В.А. Ворона. Системи контролю та управління доступом / В.А, Ворона, В.А. Тихонов. – М.: Горячая линия – Телеком, 2010. – 274 с.

3. Worst place to store password [Електронний ресурс]. Режим доступу: https://www.thersa.org/united-states/research - 3

4. Top 200 most common passwords the year 2020 [Електронний ресурс]. Режим доступу: https://nordpass.com/most-common-passwords-list/

5. Guven, A. Understanding users' keystroke patterns for computer access security. / A. Guven, I. Sogukpinar. // Computers & Security, 2003. – 695-706 с.

6. Загальна характеристика біотметричних технологій - [Електронний ресурс]. Режим доступу: https://www.biolink.ru/technology/biometric.php

7. Анализ причин возникновения ошибок первого и второго рода в системах авторизации основанных на распознавании клавиатурного почерка / А.Н. Савинов. - Программные системы и вычислительные методы – No1(1), 2012 – с. 53-59.

8. УДК 004:681.3 П. Бідюк, В. Бондарчук Інститут прикладного системного аналізу Національного технічного університету «Київський політехнічний інститут», Київ. Сучасні методи біометричної ідентифікації.

9. И.Г. Сидоркина. Три алгоритма управления доступом к КСИИ на основе распознавания клавиатурного почерка оператора: статья / И.Г. Сидоркина, А.Н. Савинов. – Вестник Чувашского университета, No3, 2013. – 9 с.

10. Система распознавания клавиатурного почерка защитит данные в дополнение к обычному паролю [Елекронний ресурс] / ИНО Томск. Режим доступу: http://inotomsk.ru/materials/news/v-tomske/sistema-raspoznavaniya- klaviaturnogo-

pocherka-zashchitit-dannye-v-dopolnenie-k-obychnomu-parolyu/, свободный. – Загл. с экрана. – Яз. рус.

11. Стахановец [Электронный ресурс] / Стахановец. URL: http://stakhanovets.ru/, свободный. – Загл. с экрана. – Яз. рус. Дата обращения: 21.04.2016 г.

12. Система распознавания клавиатурного почерка защитит данные в дополнение к обычному паролю [Електроний ресурс] / ИНО Томск. Режим доступу: http://inotomsk.ru/materials/news/v-tomske/sistema-raspoznavaniya- klaviaturnogo-pocherka-zashchitit-dannye-v-dopolnenie-k-obychnomu-parolyu/, свободный. – Загл. с экрана.

13. Говорящие клавиши. Характер человека определят по клавиатурному почерку [Електронний ресурс] / Smart News. Режим доступу: http://smartnews.ru/regions/yoshkarola/17278.html, свободный. – Загл. с экрана. – Яз. рус.

14. Клавиатурный почерк 1.0 [Електронний ресурс] / Soft For Free. Режим доступу: http://www.softforfree.com/programs/klaviaturnyi_pocherk-9813.html.

15. Banerjee, S. P., Woodard, D. L.: Biometric Authentication and Identification using Keystroke Dynamics: A Survey. Journal of Pattern Recognition Research. 7, 116—139 (2012).

16. Методы, модели и алгоритмы распознавания клавиатурного почерка в ключевых системах / Савинов А.Н. – 2013.

17. А.Н. Савинов. Решение проблемы измерения времени удержания клавиш при разработке системы анализа клавиатурного почерка: статья / 80 А.Н. Савинов, И.Г. Сидоркина. - ИКТ: образование, наука, инновации: труды III Междунар. науч.-практ. конф. Алматы: МУИТ, 2012. – 6 с.

18. Ю.А. Брюхомицкий. Гистограммный метод распознавания клавиатурного почерка: статья / Ю.А. Брюхомицкий. - Известия Южного федерального университета. Технические науки, No11, т.112, 2010. – 8 с.

19. Т.В. Жашкова. Нейросетевая идентификация типа личности человека по клавиатурному почерку: статья / Т.В. Жашкова, О.М. Шарунова, Э.Ш. Исянова. – Международный студенческий научный вестник, No3, ч.1, 2015. – 6 с.

20. Аутентификация в корпоративной компьютерной сети на основе анализа динамики клавиатурного почерка: статья / В.М. Колешко, С.А. Снигирев, Е.И. Богатов, Д.А. Гришанович, Ю.А. Безручко, С.С. Фильчук. – Минск, БГТУ, Материалы конференций факультета прикладной математики и информатики, 2009. – 3 с.

21. Искусственная нейронная сеть [Електронний ресурс] / Wikilogy. Режим доступу: http://wiki.witology.com/index.php/Искусственная_нейронная_сеть.

22. Введение в UML. Диаграммы прецедентов: крупным планом [Електронний ресурс] / Интуит. Режим доступу: http://www.intuit.ru/studies/courses/1007/229/lecture/5962.

23. https://en.wikipedia.org/wiki/Big_O_notation

24. Мова програмування JavaScript [Електронний ресурс] Режим доступу: https://uk.wikipedia.org/wiki/JavaScript.

25. Платформа .NET Core [Електронний ресурс]. Режим доступу: https://ru.wikipedia.org/wiki/.NET_Core.

26. Vue JS – The progressive Javascript framework [Електронний ресурс].Режим доступу: https://vuejs.org.

27. MySQL [Електронний ресурс]. Режим доступу: http://mysql.com

28. Формат JSON [Електронний ресурс]. Режим доступу: https://uk.wikipedia.org/wiki/JSON.

ДОДАТОК А

КЛАВІАТУРНИЙ ПОЧЕРК – МЕХАНІЗМ БІОМЕТРИЧНОЇ ІДЕНТИФІКАЦІЇ КОРИСТУВАЧА В ІНФОРМАЦІЙНІЙ СИСТЕМІ

Annotation. In the annotation of the report there is the analysis of keyboard handwriting as behavioural biometrics of the person, and suggested the mathematical methods of the analysis of the data which is used to define the identify user inside information system. Also in this report three popular methods are specified, which can be used to analyze data.

Determined approaches which are uses with mathematical methods, such as:

• Student's t-test

• Euclidean distance

The further steps of continuation of research in this direction are specified.

Key words: keyboard handwriting; biometric human identification; information system; data analysis.

Для запобігання витоку інформації або несанкціонованого доступу до інформаційних систем використовуються різні засоби ідентифікації користувача такі як:

- атрибутні
- парольні
- біометричні

Проблема атрибутних засобів в тому, що користувач, що має доступ до інформаційної системи може втратити свій атрибут – ключ, токен тощо. В іншому випадку атрибут доступу може бути втрачений або вкладений. Такою втратою може скористуватися зловмисник для отримання несанкціонованого доступу до системи.

Проблема парольних засобів полягає в тому, що користувач може забути пароль, пароль може бути підібраний, підглянутий під час набору або зчитаний за допомогою спеціального шкідливого програмного забезпечення.

Біометричні засоби можуть суттєво підвищити ступінь захисту інформаційних систем. Поведінкові характеристики користувача неможливо відтворити або вкрасти, тому сучасність вимагає відмовитись від використання атрибутних та парольних засобів на користь біометричних.

Використання методів поведінкової біометрії, яка заснована на клавіатурному почерку не вимагає придбання додаткових дорогих пристроїв та спецтехніки.

Для отримання зразка клавіатурного почерку достатньо наявність звичайної стандартної клавіатури. Це робить даний метод простим, недорогим і ненав'язливим для користувача, також цей метод може бути застосовано таємно, що дозволить покращити існуючі комп'ютерні інформаційні системи.

Тому клавіатурний почерк людини є поведінковою особливістю, яка розвивається з плином часу і не може бути змінена, втрачена або забута.

У будь-якій поведінковій біометричній характеристиці можуть спостерігатися великі зміни в особливостях характеристики. Однак, вони повинні надати достатньо інформації, щоб ідентифікувати і визначати справжність особистості за шаблоном почерку.

Тому на основі вищесказаного запропоновано вважати клавіатурний почерк індивідуальним та унікальним для кожної людини, що може бути використано процесі ідентифікації.

Для розпізнавання використовуються алгоритми розпізнавання клавіатурного почерку, які можна розділити на три групи[1]:

- алгоритми, які аналізують почерк вчасного введення пароля;
- алгоритми, які аналізують почерк після введення додаткового текстового фрагмента або фрази тощо;

- алгоритми, які постійно проводять прихований моніторинг клавіатурного почерку користувача.

Алгоритми першої групи забезпечують найбільшу швидкодію: користувачеві потрібно лише ввести свій логін і пароль. Однак точність в цьому випадку невисока, особливо в разі короткого пароля. Вхід може здійснюватися користувачем, а далі можлива підміна на іншу людину.

Алгоритми другої групи можуть забезпечити більшу точність, в порівнянні з першою групою. Однак на введення додаткового фрагмента тексту потрібен час, що може викликати негативні емоції у користувача, особливо в випадку, якщо йому часто доводиться проходити процедуру ідентифікації.

Алгоритми третьої групи дозволяють забезпечити високу точність. При цьому вони вимагають більше ресурсів. Перевагою цієї групи є можливість розпізнати зловмисника, який використовує комп'ютер, на якому раніше авторизувався студент.

Маючи шаблон клавіатурного почерку користувача стає можливо провести аутентифікацію та ідентифікацію користувача. Для цього необхідно провести процедуру порівняння поточного зразка почерку і збереженого раніше шаблону.

Запропоновано використовувати Евлідову відстань[2] та t-критерій Стьюдента[3] для порівняння часу Hold (час утримання клавіші), DownDown (час між натисканням сусідніх клавіш) та UpDown (час між опусканням однієї клавіші і натисненням наступної).

Евклідова відстань (Евклідова метрика) — формула традиційної відстані між двома точками (1.1):

$$M = \sqrt{\sum_{i=1}^{V} (A_i - B_i)^2} \qquad (1.1)$$

де M – розраховане значення Евклідової відстані; V – кількість вибірок часу Hold, DownDown, UpDown, що відповідає кількості аналізованих клавіш; A_i – час утримання клавіші з поточного зразка клавіатурного почерку користувача, що претендує на доступ; B_i – час утримання клавіші, що зберігається в шаблоні почерку.

Користувач буде успішно ідентифікований або його особа буде підтверджена, якщо розраховані значення Евклідової відстані менше встановленого в системі порога доступу.

Поріг доступу підбирається в залежності від вимог до розроблюваної системи. Основними вимогами для систем захисту інформації є ймовірності виникнення помилок першого і другого роду[4].

t-критерій Стьюдента – загальна назва для класу методів статистичної перевірки гіпотез (статистичних критеріїв), які засновані на розподілі Стьюдента. Найбільш розповсюджені випадки застосування t-критерію пов’язані з перевіркою рівності середніх значень у двох вибірках.

t-критерій Стьюдента використовується для визначення статистичної значущості відмінностей середніх величин. Може застосовуватися як у випадках порівняння незалежних вибірок, так і при порівнянні пов'язаних сукупностей.

Алгоритм Стьюдента порівняння двох вибірок:

1. Порахувати середнє значення вибірок.
2. Порахувати стандартне відхилення вибірок
3. Порахувати критерій Стьюдента за допомогою формули (1.2):

$$t = \frac{|M_1 - M_2|}{\sqrt{\frac{{\sigma_1}^2}{N_1} - \frac{{\sigma_2}^2}{N_1}}} \qquad (1.2)$$

де M_1- середнє арифметичне еталонної вибірки; M_2 - середнє арифметичне вибірки не авторизованого користувача; $\sigma_1 -$ стандартне відхилення еталонної вибірки; $\sigma_2 -$ стандартне відхилення еталонної вибірки; N_1/N_2 – об'єми вибірок.

Отримане значення t-критерію Стьюдента необхідно правильно інтерпретувати. Для цього необхідно знати кількість досліджуваних в кожній групі . Знаходимо число ступенів свободи f за формулою (1.3):

$$\mathrm{df} = (\mathrm{N}_1 + \mathrm{N}_2) - 2 \qquad (1.3)$$

де N_1/N_2 – кількість значеннь у вибірках;

Після цього визначаємо критичне значення t-критерію Стьюдента для необхідного рівня значущості (наприклад, p = 0,05) і при даному числі ступенів свободи f по таблиці критичних значень Стьюдента винесена у Додаток В

Порівнюємо критичне і розраховане значення критерію:

- якщо розраховане значення t-критерію Стьюдента дорівнює або більше критичного, знайденого по таблиці, робимо висновок про статистичної значущості відмінностей між порівнюваними величинами.
- якщо значення розрахованого t-критерію Стьюдента менше табличного, отже відмінності порівнюваних величин статистично не значимі.

Отже клавіатурний почерк посідає важливе місце в біометричній ідентифікації користувача, як один з найефективніших методів.

Даний метод дешевий, не потребує дорогого та спеціального програмного забезпечення та простий у використанні.

Правильно обрані методи аналізу даних (Евклідова відстань та t-критерій Стьюдента) швидко обчислюють правильність та точність критеріїв за допомогою яких ідентифікується користувач інформаційної системи.

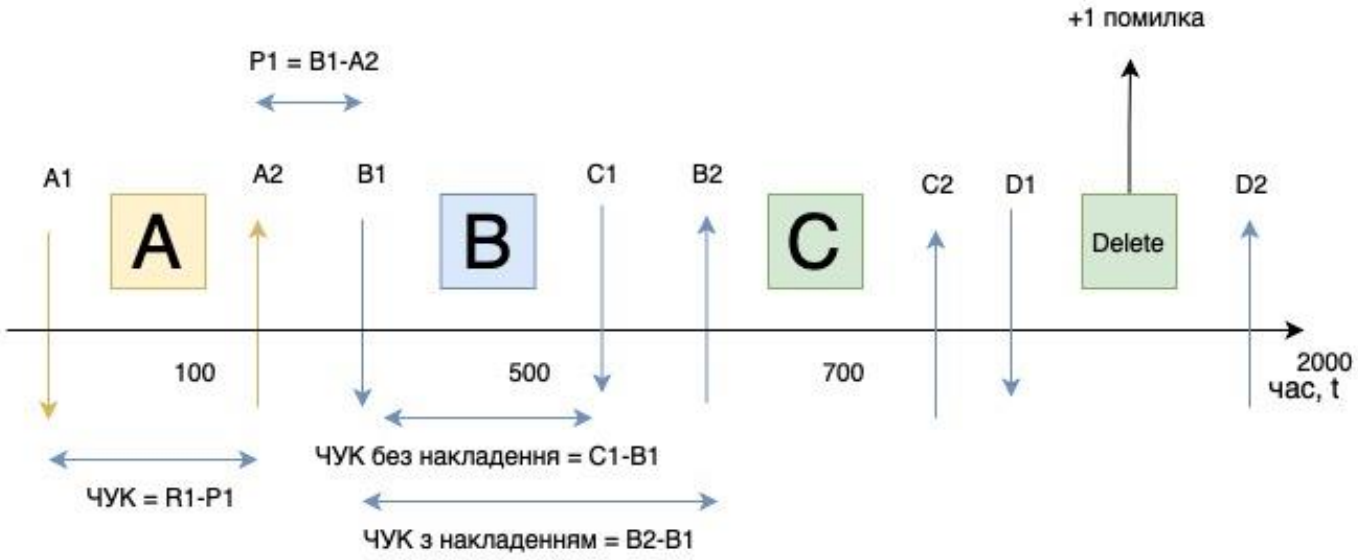
В майбутньому клавіатурний почерк може витіснити біометричну ідентифікацію за відбитком пальця та парольну.

ЛІТЕРАТУРА

1. Сидоркина И.Г., Савинов А.Н. Три алгоритма управления доступом к КСИИ на основе распознавания клавиатурного почерка оператора // Вестник Чувашского университета, 2013. – 293-301 с.
2. https://en.wikipedia.org/wiki/Student%27s_t-test
3. https://en.wikipedia.org/wiki/Euclidean_distance
4. Широчин В. П., Кулик А. В., Марченко В. В. Динамическая аутентификация на основе анализа клавиатурного почерка. - http://www.masters.donntu.edu.ua/2002/fvti/aslamov/files/bio_autentification.html

ДОДАТОК Б

Докладна схема характеристики клавіатурного почерку



ДОДАТОК В

Приклад коду клієнтської частини – компонент додавання зразку клавіатурного почерку

```
<template>
  <div class="m-2">
    <vs-card>
      <template #title>
        <h3 v-if="mode === 'education'" class="my-4">Keyboard education</h3>
        <h3 v-else class="my-4">Verify keyboard sign</h3>
      </template>
      <template #text>
        <div
          v-if="!isAddSignMode && mode === 'education'"
          class="flex justify-center"
        >
          <vs-button warn @click="addSignTemplate">Add sign template</vs-button>
        </div>

        <div v-if="isAddSignMode || (!isAddSignMode && mode !== 'education')">
          <vs-alert color="danger">
            <template #title> Warning </template>
            Please type text below and click "Submit" button.
          </vs-alert>
          <div class="sign-example">{{ userData.keySignText }}</div>
          <div>
            <textarea
              rows="5"
              cols="35"
              v-model="keySignTemplate"
              @keyup="keyEventHandler"
              @keydown="keyEventHandler"
              @keypress="keyEventHandler"
            />
          </div>
          <div class="flex justify-center">
            <vs-
button success @click="finishAddingTemplate" :disabled="userData.keySignText !== keySignTemplat
e">Submit</vs-button>
            <vs-button danger @click="resetProgress">Reset</vs-button>
            <vs-button default @click="cancel">Cancel</vs-button>
          </div>
        </div>
      </template>
    </vs-card>
  </div>
</template>
import keyBoardService from "../services/keyBoardService";

export default {
  props: {
    userData: {
      type: Object,
    },
    mode: {
      type: String,
    },
  },
  data() {
    return {
      isAddSignMode: false,
```

```
56.       keySignTemplate: "",
57.       keyEventsData: [],
58.     };
59.   },
60.   methods: {
61.     keyEventHandler(e) {
62.       let object = {};
63.
64.       object.code = e.code;
65.       object.key = e.keyCode;
66.
67.       switch (e.type) {
68.         case "keydown":
69.           object.type = 1;
70.           break;
71.         case "keyup":
72.           object.type = 2;
73.           break;
74.         case "keypress":
75.           object.type = 3;
76.           break;
77.       }
78.
79.       object.timeStamp = new Date().toISOString();
80.
81.       this.keyEventsData.push(object);
82.     },
83.     addSignTemplate() {
84.       this.isAddSignMode = true;
85.     },
86.     async finishAddingTemplate() {
87.       var response = "";
88.
89.       if (this.mode === "education") {
90.         console.log(JSON.stringify(this.keyEventsData));
91.         response = await keyBoardService.saveTemplate(
92.           this.userData.id,
93.           this.keyEventsData
94.         );
95.
96.         if (response) {
97.           this.$vs.notification({
98.             position: "bottom-center",
99.             title: "Success",
100.                text: `Keyboard sign was added succesfully`,
101.                color: "success",
102.              });
103.            } else {
104.              this.$vs.notification({
105.                position: "bottom-center",
106.                title: "Error",
107.                text: `Keyboard sign wasn't added succesfully`,
108.                color: "danger",
109.              });
110.            }
111.          } else {
112.            response = await keyBoardService.verify(
113.              this.userData.id,
114.              this.keyEventsData
115.            );
116.
117.            if (response) {
118.              console.log(this.userData);
119.              localStorage.setItem("auth_token", this.userData.authToken);
120.
121.              this.$vs.notification({
122.                position: "bottom-center",
123.                title: "Success",
```

```
124.                   text: `Keyboard sign was succesfull`,
125.                   color: "success",
126.                 });
127.
128.
129.
130.                 this.$router.push("/dashboard");
131.               } else {
132.                 localStorage.removeItem("auth_token");
133.                 this.$vs.notification({
134.                   position: "bottom-center",
135.                   title: "Error",
136.                   text: `Keyboard sign wasn't recognized succesfully`,
137.                   color: "danger",
138.                 });
139.                 this.$emit("cancelVerify");
140.               }
141.             }
142.
143.             this.isAddSignMode = false;
144.             this.keySignTemplate = "";
145.             this.keyEventsData = [];
146.             this.counter = 0;
147.           },
148.           resetProgress() {
149.             this.keySignTemplate = "";
150.             this.keyEventsData = [];
151.             this.counter = 0;
152.           },
153.           cancel() {
154.             if (this.mode !== "education") {
155.               localStorage.removeItem("auth_token");
156.               this.$emit("cancelVerify");
157.             }
158.             this.isAddSignMode = false;
159.             this.keySignTemplate = "";
160.             this.keyEventsData = [];
161.           },
162.         },
163.       };
```

ДОДАТОК Г

Приклад коду серверної частини, який відповідає за обробку даних зразку клавіатурного почерку, надісланого з клієнтської частини

```
1.  /// <summary>
2.          /// Receive and parse incoming event list
3.          /// </summary>
4.          /// <param name="events"></param>
5.          /// <returns></returns>
6.          private string ParseKeyBoardTemplate(List<KeyBoardEvent> events)
7.          {
8.              // mistakes = backspace + delete
9.              int mistakes = events.Count(e => e.Code == "Backspace" || e.Code == "Delete");
10.
11.             if (mistakes > 0)
12.             {
13.                 events = events.Where(e => e.Code != "Backspace" && e.Code != "Delete").ToList(
    );
14.             }
15.
16.             // order events if we have imposing
17.             Queue<KeyBoardEvent> orderedEvents = new Queue<KeyBoardEvent>();
18.
19.             for (int i = 0; i < events.Count; i+= 6)
20.             {
21.                 List<KeyBoardEvent> eventSet = events.Skip(i).Take(6).ToList();
22.
23.                 eventSet = eventSet.OrderBy(e => e.Code != eventSet[0].Code).ToList();
24.
25.                 foreach (var eventSetItem in eventSet)
26.                 {
27.                     orderedEvents.Enqueue(eventSetItem);
28.                 }
29.             }
30.
31.             Queue<KeyBoardEvent> filteredEvents = new Queue<KeyBoardEvent>();
32.
33.             for (int i = 0; i <= orderedEvents.Count - 3; i = i + 3)
34.             {
35.                 List<KeyBoardEvent> singleLetterEvent = orderedEvents.Skip(i).Take(3).ToList();

36.
37.                 KeyBoardEvent keyPressEvent = singleLetterEvent.FirstOrDefault(e => e.Type == (
    int)EventType.KeyPress);
38.
39.                 KeyBoardEvent keyDownEvent = singleLetterEvent.FirstOrDefault(e => e.Type == (i
    nt)EventType.KeyDown);
40.
41.                 KeyBoardEvent keyUpEvent = singleLetterEvent.FirstOrDefault(e => e.Type == (int
    )EventType.KeyUp);
42.
43.                 keyDownEvent.Key = keyDownEvent.Key;
44.                 keyUpEvent.Key = keyDownEvent.Key;
45.
46.                 filteredEvents.Enqueue(keyDownEvent);
47.                 filteredEvents.Enqueue(keyUpEvent);
48.             }
49.
```

```
50.             Queue<KeyBoardData> keyBoardData = new Queue<KeyBoardData>();
51.
52.             KeyBoardTemplate keyBoardTemplate = new KeyBoardTemplate();
53.
54.             for (int i = 0; i <= filteredEvents.Count - 3; i = i + 3)
55.             {
56.
57.                 List<KeyBoardEvent> singleLetterEvent = filteredEvents.Skip(i).Take(3).ToList()
    ;
58.
59.                 KeyBoardData singleKeyBoardData = new KeyBoardData();
60.
61.                 string key = singleLetterEvent[0].Code;
62.
63.                 if (singleLetterEvent[0].Code.Contains("Key"))
64.                 {
65.                     key = singleLetterEvent[0].Code.Substring(3, 1);
66.                 }
67.
68.                 singleKeyBoardData.Key = key;
69.                 singleKeyBoardData.KeyCode = singleLetterEvent[0].Key;
70.
71.                 // pause = tdown - tup
72.                 TimeSpan pauseTimeSpan = singleLetterEvent[2].TimeStamp -
     singleLetterEvent[1].TimeStamp;
73.
74.                 singleKeyBoardData.Pause = (decimal)pauseTimeSpan.TotalMilliseconds;
75.
76.                 // check if we have imposing - downTimeNextKey < upTimeCurrentKey
77.                 if (singleKeyBoardData.Pause < 0)
78.                 {
79.                     TimeSpan keyPressWithImposingSpan = singleLetterEvent[1].TimeStamp -
     singleLetterEvent[0].TimeStamp;
80.
81.                     singleKeyBoardData.KeyBoardPressTimeWithImposing = (decimal)keyPressWithImp
    osingSpan.TotalMilliseconds;
82.                 }
83.                 else
84.                 {
85.                     singleKeyBoardData.KeyBoardPressTimeWithImposing = 0;
86.                 }
87.
88.                 TimeSpan keyPressWithOutImposingSpan = singleLetterEvent[2].TimeStamp -
     singleLetterEvent[0].TimeStamp;
89.
90.                 singleKeyBoardData.KeyBoardPressTimeWithOutImposing = (decimal)keyPressWithOutI
    mposingSpan.TotalMilliseconds;
91.
92.                 keyBoardData.Enqueue(singleKeyBoardData);
93.
94.                 i = i - 1;
95.             }
96.
97.             // averagePauses = amount of pauses / pause count
98.             decimal pauseAmountTime = keyBoardData.Sum(k => k.Pause);
99.
100.                int pauseCount = keyBoardData.Count(k => k.Pause > 0);
101.
102.                keyBoardTemplate.AveragePauseTime = pauseAmountTime / pauseCount;
103.
104.                keyBoardTemplate.MistakeCount = mistakes;
105.
106.                // final key list with all average data
107.                Queue<KeyBoardSignTemplate> keyBoardDataForTemplate = new Queue<KeyBoardSign
    Template>();
108.
109.                List<string> uniqueKeys = keyBoardData.Select(k => k.Key).Distinct().ToList(
    );
```

```
110.
111.                     for (int i = 0; i < uniqueKeys.Count; i++)
112.                     {
113.                         KeyBoardSignTemplate finalKeyBoard = new KeyBoardSignTemplate();
114.
115.                         finalKeyBoard.Key = uniqueKeys[i];
116.
117.                         List<KeyBoardData> uniqueKeyList = keyBoardData.Where(k => k.Key == uniq
    ueKeys[i]).ToList();
118.
119.                         finalKeyBoard.KeyCode = uniqueKeyList[0].KeyCode;
120.
121.                         // average vuk with superimposing for each unique letter
122.
123.                         decimal keyBoardPressWithImposingSum = uniqueKeyList.Sum(k => k.KeyBoard
    PressTimeWithImposing);
124.
125.                         finalKeyBoard.AverageKeyBoardPressTimeWithImposing = keyBoardPressWithIm
    posingSum / uniqueKeys.Count;
126.
127.                         // average vuk without superimposing for each unique letter
128.
129.                         decimal keyBoardPressWithOutImposingSum = uniqueKeyList.Sum(k => k.KeyBo
    ardPressTimeWithOutImposing);
130.
131.                         finalKeyBoard.AverageKeyBoardPressTimeWithOutImposing = keyBoardPressWit
    hOutImposingSum / uniqueKeys.Count;
132.
133.                         keyBoardDataForTemplate.Enqueue(finalKeyBoard);
134.                     }
135.
136.                     keyBoardTemplate.KeyBoardList = keyBoardDataForTemplate.ToList();
137.
138.                     string keyBoardTemplateJson = Newtonsoft.Json.JsonConvert.SerializeObject(ke
    yBoardTemplate);
139.
140.                     return keyBoardTemplateJson;
141.                 }
```

ДОДАТОК Д

Код для створення схеми бази даних проекту

```
1.  CREATE SCHEMA diploma_schema;
2.  CREATE TABLE `key_board_templates` (
3.    `id` int NOT NULL AUTO_INCREMENT,
4.    `userId` int NOT NULL,
5.    `insert_date` datetime DEFAULT NULL,
6.    `template` json NOT NULL,
7.    `hidden` tinyint(1) DEFAULT '0',
8.    PRIMARY KEY (`id`)
9.  ) ENGINE = InnoDB AUTO_INCREMENT = 70 DEFAULT CHARSET = utf8mb4 COLLATE = utf8mb4_0900_ai_ci;
10. CREATE TABLE `users` (
11.   `id` int NOT NULL AUTO_INCREMENT,
12.   `login` varchar(45) NOT NULL,
13.   `password` varchar(45) NOT NULL,
14.   `auth_token` varchar(45) DEFAULT NULL,
15.   `key_sign_acticated` int DEFAULT '0',
16.   `key_sign_text` longtext,
17.   PRIMARY KEY (`id`)
18. ) ENGINE = InnoDB AUTO_INCREMENT = 27 DEFAULT CHARSET = utf8mb4 COLLATE = utf8mb4_0900_ai_ci;
```

ДОДАТОК Е

Табличний вигляд клавіатурного почерку користувача

	1	2	3	4	5	6	7
A	15,62	17,54	16,08	15,85	17,46	17,00	30,85
L	10,62	10,38	10,62	10,15	10,54	9,54	7,77
E	71,15	48,38	30,38	31,54	30,15	35,54	32,23
X	13,77	13,77	14,38	13,62	12,38	14,54	13,31
N	16,38	18,62	16,62	14,62	17,54	13,92	17,15
D	11,38	11,23	10,46	9,77	10,08	9,23	11,00
R	22,31	12,08	10,62	9,15	11,77	32,85	11,15
Space	18,54	12,00	15,31	12,15	14,92	14,38	12,77
K	30,31	47,85	31,31	28,92	52,00	62,15	31,92
U	8,85	11,31	6,77	8,46	9,54	8,38	9,54
Z	8,08	8,23	8,38	8,92	8,08	9,77	8,92
M	9,00	11,38	10,23	8,62	14,62	8,62	9,38
O	28,62	23,23	25,85	33,08	28,46	34,46	23,77