

ДВНЗ «ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»

Факультет комп'ютерно-інформаційних технологій та автоматизації

Кафедра прикладної математики та інформатики

«До захисту допущено»

Завідувач кафедри



Ольга ДМИТРИЄВА

(підпис)

(ім'я та прізвище)

«8» червня 2022 р.

Випускна кваліфікаційна робота бакалавра

(освітній ступінь)

на тему «Біометрична аутентифікація користувача при введенні логіна та пароля» із спецчастиною «Розробка модуля системи розпізнавання користувача за клавіатурним почерком»

Виконала: студентка 4 курсу, групи КІБ-18

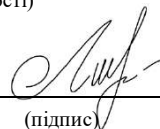
(шифр групи)

спеціальності (напрямку підготовки) 125 Кібербезпека

(шифр і назва напрямку підготовки, спеціальності)

Єжова Є. О.

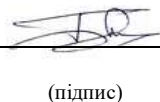
(прізвище та ініціали)



(підпис)

Керівник проф. кафедри ПМІ, д.т.н., проф. Євген БАШКОВ

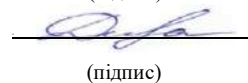
(посада, науковий ступінь, вчене звання, прізвище та ініціали)



(підпис)

Рецензент доц. кафедри КІ, к.т.н. Юлія ДІКОВА

(посада, науковий ступінь, вчене звання, прізвище та ініціали)



(підпис)

Нормоконтроль:



к.т.н., доц. Ірина НАЗАРОВА

(підпис)

30 травня 2022

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент



(підпис)

Луцьк – 2022р.

ДВНЗ «Донецький національний технічний університет»
Факультет комп'ютерно-інформаційних технологій та автоматизації
Кафедра прикладної математики та інформатики
Освітній ступінь бакалавр
Напрямок підготовки (спеціальність) 125 «Кібербезпека»

ЗАТВЕРДЖУЮ:
Завідувач кафедри ПМІ

Ольга ДМИТРИЄВА
06.04.2022 р.



ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТУ

Єжова Єлизавета Олексіївна

(прізвище, ім'я, по батькові)

1. Тема роботи «Біометрична аутентифікація користувача при введенні логіна та пароля»

Спецчастина «Розробка модуля системи розпізнавання користувача за клавіатурним почерком»

Керівник роботи Башков Є. О., д.т.н., проф., проф. каф. ПМІ

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом від “ 06 ” травня 2022 року № 180

2. Строк подання студентом роботи “17” червня 2022 року

3. Вихідні дані до роботи результати науково-дослідної роботи, матеріали переддипломної практики

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

1) аналіз предметної області та постановка задачі;

2) огляд та порівняння методів біометричної аутентифікації;

3) розробка математичної моделі користувача, що проходить аутентифікацію;

4) розробка модулю біометричної аутентифікації користувача при введенні логіну;

5) тестування розробленого програмного модуля;

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) робота містить 14 таблиць, 20 рисунків та інший матеріал, у кількості, достатній для демонстрації результатів кваліфікаційної роботи

6. Консультанти розділів роботи

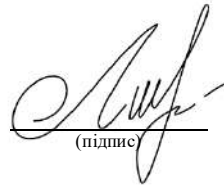
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 6 квітня 2022

КАЛЕНДАРНИЙ ПЛАН

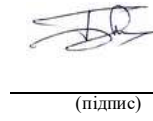
№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Прим.
1	Об'єктний аналіз предметної області і постановка завдання	6.04.2020	
2	Огляд літератури з предметної області	10.04.2020	
3	Збір інформації і методичних відомостей щодо механізмів забезпечення біометричної аутентифікації користувача	15.04.2020	
4	Розробка структури програмного модулю, вибір даних для обробки та експериментів.	18.05.2020	
5	Тестування системи	22.05.2022	
6	Оформлення пояснювальної записки	28.05.2022	
7	Нормоконтроль пояснювальної записки, її перевірка антиплагіатною системою	01.06.2022	
8	Оформлення супутніх матеріалів (розробка графічного матеріалу, тощо) та рецензування роботи	05.06.2022	
9	Написання доповіді	12.06.2022	
10	Захист роботи	24.06.2022	

Студент


(підпис)

Єжова Є.О.
(прізвище та ініціали)

Керівник роботи


(підпис)

Башков Є.О.
(прізвище та ініціали)

АНОТАЦІЯ

Єжова Єлизавета Олексіївна. Біометрична аутентифікація користувача при введенні логіна та пароля / Випускна кваліфікаційна робота на здобуття освітнього ступеня «бакалавр» за спеціальністю 125 «Кібербезпека». – ДВНЗ ДонНТУ, Покровськ, 2022.

Об'єктом дослідження є процеси захисту інформації від несанкціонованого доступу.

Предметом дослідження є методи біометричної аутентифікації користувача при введенні логіна та пароля.

Метою роботи є аналіз захищеності інформації з урахуванням аналізу клавіатурного почерку.

Виходячи з мети, завданням даної дипломної роботи є аналіз сучасних видів аутентифікації користувачів; розробка методу аутентифікації за допомогою клавіатурного почерку; розроблення програмного модуля даного методу, використовуючи мову C#.

Проведені дослідження в даній роботі базуються на сучасних методах аналізу механізмів захисту аутентифікації.

Наукова новизна полягає в розробці методу біометричної аутентифікації, шляхом використання клавіатурного почерку, а саме використання унікальних біометричних даних, що дозволило проводити аутентифікацію без необхідності втручання користувача безпосередньо в цей процес.

Практичне значення полягає в авторській розробці методу фільтрації авторського введення від неавторського та програмного модуля реалізації розробленого методу аутентифікації з використанням об'єктно-орієнтованої мови програмування C# на базі використання інтегрованого середовища розробки Visual Studio.

Дипломна робота складається з 4 розділів, включає 20 рисунків, 14 таблиць, 46 джерела інформації, загалом 120 сторінки.

На тему дипломної роботи опубліковано такі наукові роботи (наведені в додатку Д):

1. Єжова Є.О., Overview and problems of biometric authentication methods – Трускавець: Актуальні питання та перспективи проведення наукових досліджень: матеріали III Міжнародної студентської наукової конференції (Т. 3), м. Трускавець, 10 грудня, 2021 рік / ГО «Молодіжна наукова ліга». – Вінниця: ГО «Європейська наукова платформа», 2021. – 136 с. – 53–55 с. [1].

Ключові слова: аутентифікації, біометричні аутентифікації, біометричні характеристики, клавіатурний почерк, мова C#.

ABSTRACT

Yezhova Elizaveta. Biometric user authentication when entering a login and password / Graduation thesis for the degree of "bachelor" for specialty 125 "Cybersecurity". – SHEI DonNTU, Pokrovsk, 2022.

The object is the protection of information from unauthorized access.

The subject of the study is the developed method of biometric user authentication when entering a login and password.

The aim of the work is to analyze the security of information based on the analysis of keyboard handwriting.

Proceeding from the purpose, the task of this thesis is to analyze modern types of user authentication; development method of authentication using keyboard handwriting; development of software module of this method, using C# language.

The conducted research in this work is based on modern methods of analysis of authentication protection mechanisms.

Scientific novelty lies in the improvement of biometric authentication method by using keyboard handwriting, namely the use of unique biometric data, which allowed for authentication without the need for user intervention directly in the process.

Practical significance consists in author's development of method of filtering of author's input from non-author's and software module of developed authentication method realization with using of object-oriented programming language on the basis of using of integrated development environment Visual Studio.

Diploma work consists of 4 sections, includes 20 drawings, 14 tables, 46 sources of information, in total 120 pages.

The following research papers have been published on the topic of the thesis:

1. E. Ezhova, Oversight and problems of biometric authentication methods – Truskavets: Actual problems and prospects of scientific research:

Proceedings of the III International Student Scientific Conference (Vol. 3), Truskavets, December 10, 2021 / Public Association "Youth Scientific League". – Vinnytsia: NGO "European scientific platform", 2021. – 136 c. – 53–55 c. [1].

Keywords: authentication, biometric authentication, biometric characteristics, keyboard handwriting, C # language.

ЗМІСТ

Вступ	11
1 Огляд методів біометричної аутентифікації	13
1.1 Статичні методи біометричної аутентифікації	13
1.1.1 Аутентифікація за відбитком пальців	14
1.1.2 Аутентифікація за райдужкою ока	15
1.1.3 Аутентифікація за сітківкою ока	17
1.1.4 Аутентифікація за геометрією кисті	18
1.1.5 Аутентифікація за геометрією обличчя	20
1.1.6 Аутентифікація за термограмою обличчя	21
1.2 Динамічні методи	22
1.2.1 Аутентифікація за голосом	22
1.2.2 Аутентифікація за рукописним підписом	24
1.2.3 Аутентифікація за клавіатурним почерком	25
1.3 Порівняння видів біометричної аутентифікації	27
1.4 Висновки за розділом	29
2 Розроблення методики клавіатурної аутентифікації	31
2.2 Математична модель клавіатурного почерку користувача	33
2.2.1 Функція процесу набору тексту	33
2.2.2 Вектор вхідних параметрів	35
2.3 Опис процесу біометричної аутентифікації користувача	36
2.4 Висновки за розділом	41
3 Практична реалізація	42
3.1 Вибір засобів реалізації програмного модуля	42
3.2 Опис практичної реалізації	43
3.3 Висновки за розділом	48
4 Аналіз результатів дослідження	49
4.1 Результати дослідження параметрів клавіатурного почерку	49

	10
4.2 Тестування працездатності програмного модуля	57
4.3 Висновки за розділом	59
Висновки	60
Список використаних джерел	61
Додаток А Зауваження нормоконтролера	67
Додаток Б Таблиця значень критерію Ст'юдента (t-критерію)	68
Додаток В Дані для проведення тестування	69
В.1 Інтервали між натисканням клавіш автором та їх характеристики	69
В.2 Час утримання клавіш автором	70
В.3 Інтервали між натисканням клавіш та їх характеристики	72
В.4 Час утримування клавіш учасників та його характеристики	78
Додаток Г Лістинг програмного коду	85
Додаток Д Матеріали наукової роботи	106
Додаток Е Матеріали презентації	110

ВСТУП

За останні чверть століття, завдяки розвитку технологій, більшість людей придбали розумні електронні пристрої і щодня використовують їх для спілкування, пошуку інформації, торгівлі, банківських та інших операцій. Кожен користувач такого пристрою стикається з процесом аутентифікації кілька разів на день.

Традиційні методи засновані на використанні електронних ключів, карток, паролів і кодів доступу. Після отримання від користувача даних, вони порівнюються з даними в спеціальній захищеній базі даних, і якщо аутентифікація пройшла успішно, виконується авторизація, яка згодом дозволяє користувачеві працювати в системі.

Основним недоліком цього методу є неоднозначність особи, яка ідентифікується. Це пов'язано з використанням метрик ідентифікації на основі знань для визначення аутентичності особи.

Ще одним важливим недоліком традиційних методів аутентифікації є неможливість виявити заміну аутентифікованих користувачів, що дозволяє зловмисникам отримати доступ до системних ресурсів, обмежених лише привілеями ідентифікованого користувача [2].

Вищезазначені недоліки можна виправити, доповнивши систему захисту біометричними методами аутентифікації, які включають ідентифікацію особи за її унікальними фізичними або поведінковими характеристиками. Серед біометричних методів аутентифікації значну увагу приділено методам, заснованих на динамічному клавіатурному (комп'ютерному) почерку людини через їх доступність та необхідність не купувати додаткових пристроїв введення.

Отже, виходячи з вищевикладеного, завданням даної дипломної роботи є аналіз сучасних видів аутентифікації користувачів; розробка методу аутентифікації за допомогою клавіатурного почерку; розроблення

програмного модуля даного методу, проведення тестування працездатності розробленого програмного продукту.

В результаті роботи буде розроблено алгоритм, а також спроектований та реалізований модуль системи розпізнавання користувача за клавіатурним почерком.

У майбутньому планується подальша розробка та вдосконалення модуля системи з метою підвищення точності роботи.

У додатку наведено лістинг програмного модулю аналізу та фільтрації введів.

1 ОГЛЯД МЕТОДІВ БІОМЕТРИЧНОЇ АУТЕНТИФІКАЦІЇ

Біометрична аутентифікація належить до автоматизованого розпізнавання людей на основі їхніх біологічних та поведінкових ознак [3]. Аутентифікація особи користувача може бути виконана трьома способами:

- щось, що людина знає (пароль);
- щось, що у людини є (ключ, спеціальна карта);
- щось, чим людина є (відбитки пальців, слід).

По суті, це система розпізнавання образів, яка використовується для ідентифікації чи перевірки користувачів на основі їх унікальних фізичних характеристик. Біометричні системи мають низку переваг перед традиційними методами аутентифікації. Біометричну інформацію не можна отримати шляхом прямого прихованого спостереження. Її неможливо передати та важко відтворити. Вона підвищує зручність користувача, позбавляючи його необхідності запам'ятовувати довгі й випадкові паролі. Біометрія забезпечує однаковий рівень безпеки всім користувачів, на відміну від паролів, і повністю стійка до атак методом перебору.

Існує безліч методів біометричної ідентифікації, які можна розділити на дві великі групи: статистичні та динамічні [1].

1.1 Статичні методи біометричної аутентифікації

Статичні методи ґрунтуються на біологічних характеристиках людини, які існують у неї від народження. Їх неможливо втратити, вкрати чи забути. Незмінність і відкритість біометричних даних статичного методу, дозволяє підробляти біометричні ключі. Роздивимось деякі з методів, які відносяться до статистичних [1].

1.1.1 Аутентифікація за відбитком пальців

В основі цього методу лежить унікальність малюнка ліній папілярних візерунків на пальцях кожної людини. Піднесені лінії називаються хребтами, а місця западин між двома хребтами – долинами. Розпізнавання відбитків пальців – це процес пошуку відбитків пальців з однаковим малюнком. Біометрична інформація, необхідна для аутентифікації, називається ознакою, а ознаки, знайдені в відбитках пальців, називаються мініатюрами [4].

Мініатюри, як показано рисунку 1.1, поділяються на два типи кінцець та розвилка. Кінець є місцем, де закінчується хребетний потік, а розвилка – це місце, де два хребти зливаються в один. Зображення відбитка пальця має кілька кінців та розвилок [5].

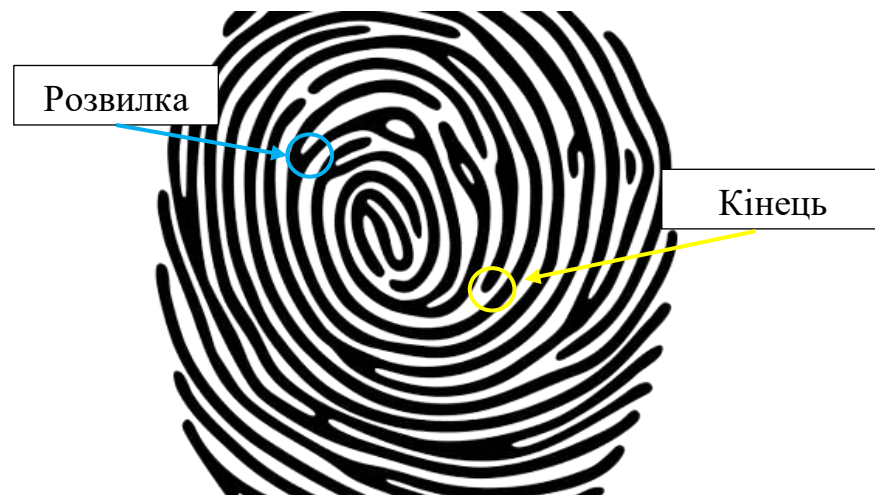


Рисунок 1.1 – Види мініатюр

Джерело: авторська розробка беручи до уваги вищевикладене

Відбиток пальця – це власна інформація користувача, але вона повинна мати властивість змінюватись, оскільки після витоку інформації всі системи аутентифікації, що використовують цю біометричну інформацію, стануть ненадійними.

Переваги аутентифікації за відбитком пальців:

- відбиток пальця не схожий на пароль тому, що його неможливо вгадати;

- вона проста у використанні;
- відбиток пальця не можна втратити як карту доступу;
- вона надзвичайно точна;
- обладнання є економічно ефективним;
- біометричний шаблон вимагає мало місця для зберігання, що зменшує розмір пам'яті бази даних;
- фізичні характеристики набагато складніше підробити, ніж ідентифікаційні карти.

Недоліки аутентифікації за відбитком пальців:

- більш нав'язлива ідентифікація, ніж ідентифікація за ім'ям користувача або пароллю;
- на точність та роботу системи впливає стан шкіри людини.

Отже, у порівнянні з іншими методами аутентифікації користувачів, цей метод біометричної аутентифікації з використанням фізичних характеристик користувача має високий рівень безпеки. Однак, цей метод має й серйозний недолік безпеки, оскільки ключ аутентифікації користувача не можна змінити [6].

1.1.2 Аутентифікація за райдужкою ока

Під час розпізнавання райдужної оболонки ока використовуються її відмінні структури [7].

Типова система розпізнавання райдужки включає чотири основні модулі, що зазначені на рисунку 1.2.

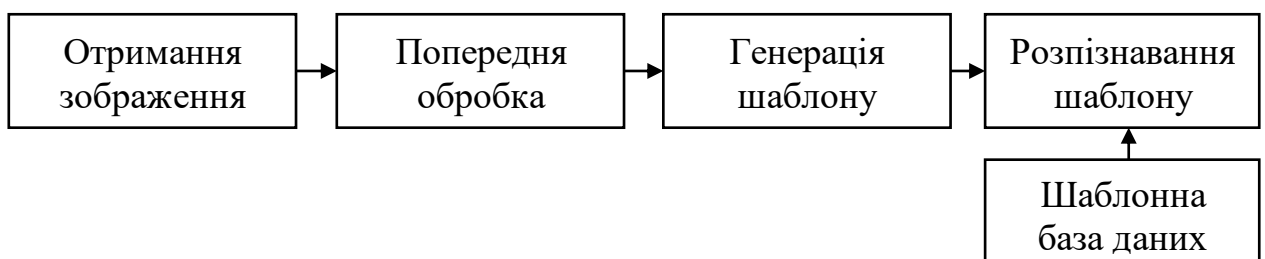


Рисунок 1.2 – Типова система розпізнавання райдужної оболонки

Джерело: [8]

На етапі отримання зображення камери райдужки ока є монохроматичними та використовують ближній інфрачервоний діапазон освітлення (область від 0,78 до 3,0 мкм [9]).

Етап попередньої обробки містить методи виявлення та виключення повік, вій та відбиття. Він дозволяє виділити та витягти із зображення малюнок райдужки ока. Під час генерації шаблону з витягнутих шаблонів райдужки створюється шаблон її коду.

На етапі розпізнавання шаблону шаблон оболонки порівнюється з іншими, щоб визначити, чи знайдено збіг. Програма зіставлення райдужної оболонки розраховує кількість бітів різниці між двома шаблонами [10].

Технології сканування райдужної оболонки ока мають величезний потенціал для застосування у сфері безпеки у всьому світі, включаючи банківську справу та особисту безпеку. Навіть високоточні автоматизовані технології, такі як розпізнавання відбитків пальців, для яких ймовірність помилкового збігу може становити 1 зі 100000, не наближаються до показників технології сканування райдужної оболонки ока [7].

Але існує ймовірність виникнення проблем, пов'язаних з розвитком очної патології у пацієнтів. Через це можуть бути труднощі з заходами безпеки, спричинені розвитком очного захворювання у будь-який час після початкової реєстрації райдужки.

З вище викладеного перевагами аутентифікації за райдужкою ока є:

- висока точність даного види аутентифікації;
- райдужка – це високозахищений внутрішній орган ока;
- імітація неможлива, а малюнок не можна змінити без ризику для ока;
- розмір шаблону невеликий;
- швидка обробка отриманого шаблону;
- при скануванні відсутній фізичний дотик;
- оскільки рогівка захищає райдужну оболонку, вона не сильно змінюється із віком;

Недоліки аутентифікації за райдужкою ока:

- оскільки оптичні зчитувачі дуже складні в експлуатації, вони вимагають підвищеної кваліфікації працівників;
- не дуже прості у використанні;
- на відстані більше кількох метрів розпізнавання райдужної оболонки ока вкрай утруднене;
- висока вартість;
- неможливо використовувати звичайну камеру, потрібен датчик джерела ІЧ-світла, оскільки видиме світло має бути зведене до мінімуму для досягнення максимальної точності, необхідної для пошуку.

1.1.3 Аутентифікація за сітківкою ока

Дана аутентифікація використовує малюнок кровоносних судин у сітківці ока, яка зображена на рисунку 1.3, як біометричний параметр: цей малюнок практично неможливо підробити підставною особою. Крім того, макроскопічна конфігурація судинної сітківки залишається стабільною протягом усього життя, а малюнок її розгалуження є унікальним для кожної людини [11].

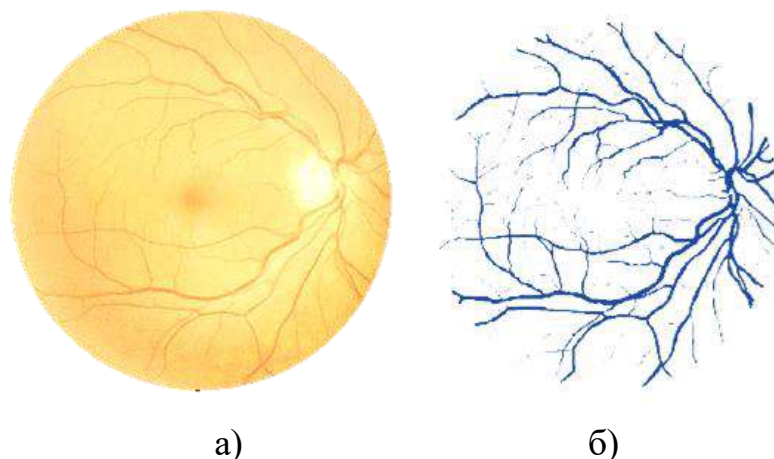


Рисунок 1.3 – Кольорове зображення сітківки (а) та сегментоване зображення судин (б)

Джерело: [12]

Деякі клінічні стани можуть викликати зміни судинної мережі, однак ці зміни обмежуються мікросудинами сітківки, і лише на останніх стадіях можуть бути залучені великі судини сітківки. Більш того, через своє внутрішнє розташування сітківка захищена від змін, викликаних впливом зовнішнього середовища (як у випадку з відбитками пальців, долонь тощо).

Ці факти, а також надійність методу реєстрації, який добре працює, навіть якщо деякі судини судинного дерева сітківки зникають через рух очей, роблять цей метод аутентифікації одним з найкращих для середовищ з високим рівнем безпеки.

Слід зазначити, що отримання шаблону полягає лише в отриманні цифрової фотографії очного дна.

1.1.4 Аутентифікація за геометрією кисті

Геометрія руки – це методика, що пов'язана з геометричною структурою руки, що містить довжину пальців, ширину пальців у різних місцях, товщину долоні, діаметр долоні тощо [13].

Типова система геометрії руки використовує камеру або пристрій на основі сканера для отримання зображень. Зображення, названі шаблонами, попередньо обробляються для отримання характеристик, в яких реалізується серія вимірювань, потім вони поміщаються в базу даних для кожного з користувачів. Виконується процес аутентифікації, у якому вхідний шаблон зіставляється з усіма записами у базі для перевірки особистості людини. У результаті людина може бути авторизована чи ні [14, 15].

Система аутентифікації складається з таких основних модулів: як сенсор, генератор характеристик, пристрій зіставлення, база даних рішень і шаблонів [16] та зазначена на рисунку 1.4.

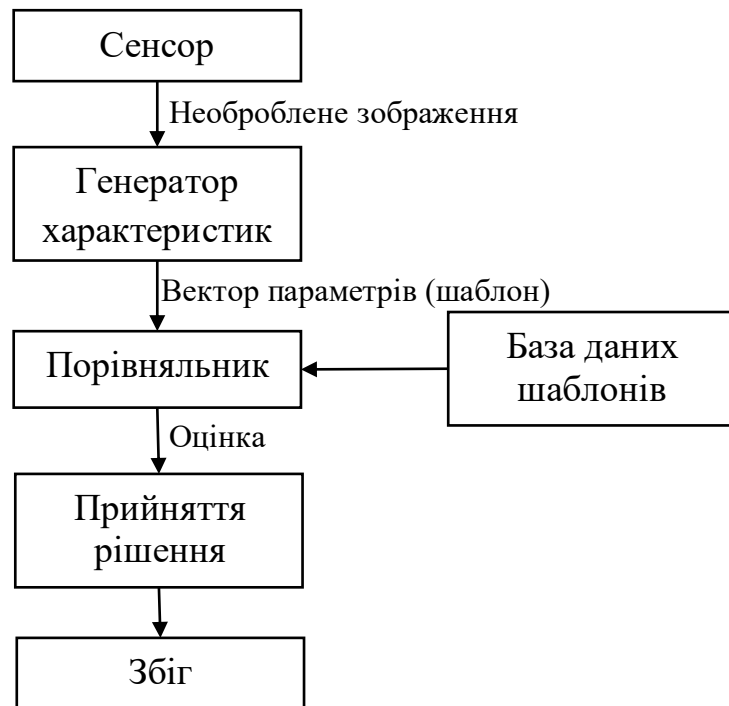


Рисунок 1.4 – Типова система аутентифікації за геометрією кисті

Джерело: авторська розробка беручи до уваги вищевикладене

Переваги аутентифікації за геометрією кисті:

- простота використання;
- основним методом підробки є зліпок або латексна модель руки, яку важко виконати, особливо, якщо застосовуються прості заходи фізичної безпеки, тому висока стійкість до підробки;
- зчитувачі довговічні та здатні обробляти великі обсяги користувачів протягом кількох років без зайвого виходу зчитувача з ладу;
- зчитувачі можуть витримувати широкий діапазон температур і працювати в несприятливих.

Недоліки аутентифікації за геометрією кисті:

- сканери геометрії руки відносно великі та дорогі;
- хоча основна структура кисті мало змінюється з часом, але травми, набряки або захворювання, такі як артрит, можуть загасати цю структуру та викликати проблеми з розпізнаванням;
- геометрія руки немає достатньої відмінності, щоб забезпечити пошук за принципом "один до багатьох" і зазвичай обмежується

використанням для аутентифікації 1 до 1, тому її використання обмежено перевіркою особи, а не ідентифікацією людини за базою даних.

1.1.5 Аутентифікація за геометрією обличчя

Системи розпізнавання за геометрією обличчя використовують комп'ютерні алгоритми для вибору конкретних унікальних деталей обличчя. Ці деталі, такі як відстань між очима або форма щелепи, потім перетворюються в математичне подання і порівнюються з іншими, зібраними в системі розпізнавання обличчя. У системах верифікації особи користувач робить "позитивне" утвердження щодо особи, що вимагає порівняння "один до одного" поданого зразка особи із зареєстрованим шаблоном заявленої особи [17]. Процедура показана рисунку 1.5.



Рисунок 1.5 – Типова система аутентифікації обличчя

Джерело: авторська розробка беручи до уваги вищевикладене

Перевагами аутентифікації за геометрією обличчя є те, що:

- система розпізнавання осіб проста у використанні й в будь-якому випадку може працювати так, що людина про це навіть не дізнається;
- ціна камери невелика;
- система розпізнавання осіб зручна для користувача.

Недоліки розпізнавання облич:

- система розпізнавання осіб неспроможна відрізнити однайцевих близнюків;
- впливає довкілля;
- високий діапазон помилкових збігів;
- проблема з хибним відторгненням, коли люди змінюють зачіски.

1.1.6 Аутентифікація за термограмою обличчя

Судинна система на обличчі людини створює унікальний характер зображення обличчя, коли тепло проходить крізь тканини обличчя і виділяється зі шкіри.

Такі ознаки обличчя можна зафіксувати за допомогою інфрачервоної камери, в результаті чого утвориться зображення, яке називається «термограмою обличчя». Термограма обличчя індивідуальна для кожної людини й не піддається маскуванню. Вважається, що навіть пластична хірургія, яка не перенаправляє потік крові по венах, не впливає на формування термограми обличчя [18].

Приклад термограми наведений на рисунку 1.6.

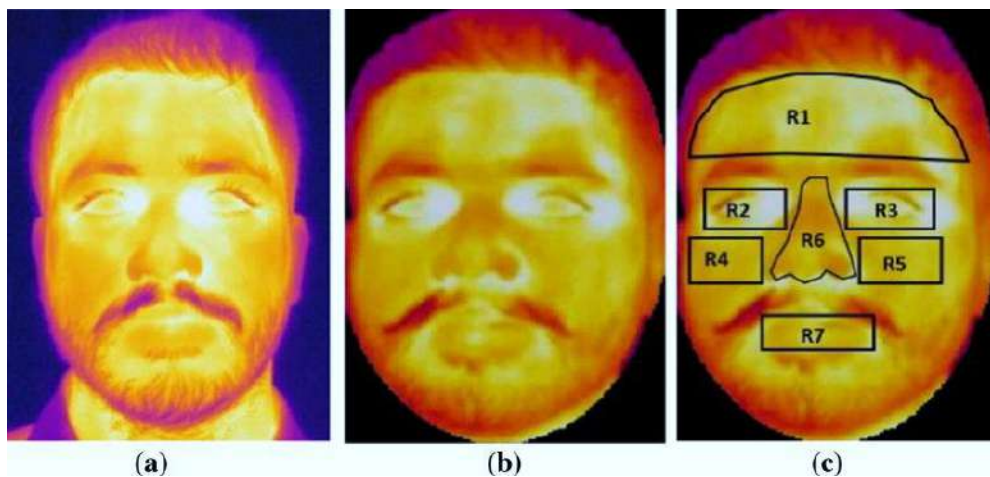


Рисунок 1.6 – Ілюстрація: а) термограми обличчя, б) витягу обличчя, с) характерних ділянок

Джерело: [19]

Термограма обличчя – це ненав’язливий біометричний метод, який дозволяє безконтактно підтвердити особу.

Заявлена перевага розпізнавання на основі термограми обличчя над візуальним розпізнаванням обличчя за допомогою звичайних камер засновано на наступних спостереженнях:

- інфрачервона камера може знімати термограму обличчя при дуже низькому освітленні або за відсутності світла взагалі;
- судинна структура може бути більш насиченою інформацією і залишається інваріантною до навмисних або ненавмисних варіацій візуального вигляду обличчя [18].

Хоча термограми обличчя і є унікальними для кожної людини, але не було доведено, що вони є достатньо легковпізнавані. Термограми обличчя можуть сильно залежати від ряду факторів, таких як емоційний стан суб’єктів або температура тіла.

Тепер перейдемо до наступного типу біометричної аутентифікації, а саме – динамічних методів.

1.2 Динамічні методи

Динамічні методи аутентифікації ґрунтуються на характеристиках поведінки людини. Вони оцінюють унікальну поведінку людини та її підсвідомі рухи у процесі відтворення будь-якої дії [20].

1.2.1 Аутентифікація за голосом

Відбиток голосу – це надійна фізіологічна та поведінкова комбінована біометрія, яка вважається такою ж біологічно унікальною для людини, як і відбитки пальців.

Є деякі переваги використання голосу як біометричної аутентифікації особистості це:

- інтуїтивно зрозуміла та природня технологія, оскільки використовує людський голос;

- може забезпечити віддалену аутентифікацію без необхідності присутності користувача;
- є недорогою технологією.

Недоліками використання голосу як біометричної аутентифікації є:

- мінливість мови через фонові шуми та часові зміни голосу
- низька безпека і низька точність, через те, що шахраї можуть підслуховувати зразки мови легітимного користувача або використовувати різні технології штучного інтелекту для створення синтетичних голосових даних, а потім провести "атаку відтворення" проти систем аутентифікації на основі голосу [21].

Схема підходу до розпізнавання голосу показана на рис. 1.7, вона показує роботу системи у двох режимах: режим навчання та режим розпізнавання.

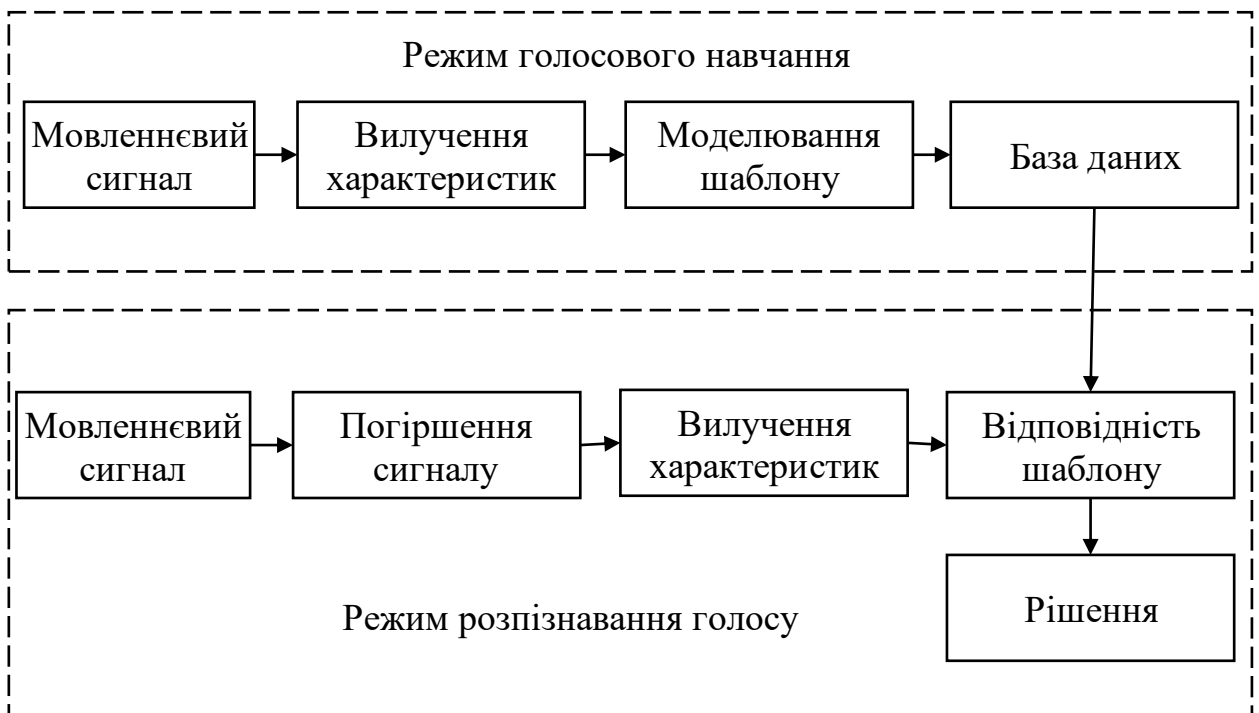


Рисунок 1.7 – Система аутентифікації за голосом

Джерело: розроблено автором за даними [22, 23]

1.2.2 Аутентифікація за рукописним підписом

Біометрична аутентифікація з використанням рукописного підпису може бути реалізована без додаткових датчиків, крім ручки та аркуша паперу. Навіть якщо форма підпису відома іншим, з проблемою можна впоратися, змінивши її форму. Аутентифікація з рукописного підпису ґрунтується на системах перевірки та ідентифікації підпису. Приналежність даного підпису конкретній особі визначається системою ідентифікації підпису, тоді як система перевірки підпису вирішує, належить цей підпис заявленій особі чи ні [24].

Аутентифікація на основі підпису може бути статичною або динамічною. У статичному режимі доступне лише цифрове зображення підпису. У динамічному режимі підписи отримують за допомогою графічного планшета або чутливого до ручки комп'ютерного дисплея.

На додаток, на відміну від більшості інших біометричних даних, у разі компрометації, доклавши певних зусиль, користувач може змінити свій підпис. З іншого боку, на нього можуть впливати фізичні та емоційні стани, і він має значну мінливість, яку необхідно враховувати в процесі аутентифікації. Перевірка підпису аналізує, як користувач підписує ім'я. Такі характеристики підпису, як швидкість, прискорення, швидкість і тиск, важливі, як і статична форма підпису [25].

Схема процесу аутентифікації за рукописним підписом наведено на рисунку 1.8.

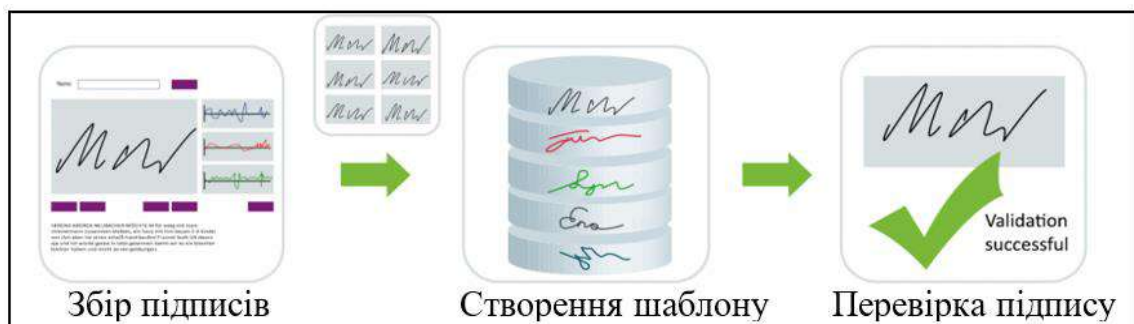


Рисунок 1.8 – Аутентифікація за рукописним підписом

Джерело: [26]

Переваги аутентифікації за рукописним підписом:

- найдешевша технологія;
- реєстрація інтуїтивно зрозуміла та швидка;
- перевірка підпису загалом має швидкий відгук та низькі вимоги до зберігання даних;
- невеликий час перевірки.

Недоліки аутентифікації за рукописним підписом:

- складність у використанні;
- для досягнення необхідної точності системи може знадобитися п'ятивимірне перо. Це робить апаратні витрати високими.
- деякі люди страждають на параліч, а в інших недостатньо тонкої моторної координації, щоб писати послідовно;
- рукописний підпис має тенденцію змінюватися з часом.

1.2.3 Аутентифікація за клавіатурним почерком

Моніторинг динаміки натискання клавіш вважається поведінковим методом аутентифікації користувачів, який використовує шаблони набору тексту людини для підтвердження його особистості. Динаміка натискання клавіш – це "не те, що ви набираєте, а те, як ви набираєте". При такому підході користувач набирає текст, як завжди, без будь-якої додаткової роботи для аутентифікації. До того ж для цього використовується лише власна клавіатура користувача, а не жодне інше зовнішнє обладнання [27].

Існує два основних етапи, через які повинен пройти користувач, щоб бути авторизованим у системах аутентифікація за клавіатурним почерком, а саме: етап реєстрації та етап входу до системи.

Перша фаза пов'язана зі збором даних про користувача, таких як ім'я користувача або пароль, а також з відстеження поведінки користувача при наборі тексту. Система збирає дані про час натискання клавіш та витягує тимчасові характеристики для створення шаблону поведінки кожного користувача під час набору тексту. Цей шаблон, також званий профілем

користувача, зберігається у базі даних відповідно до інших даних користувача.

Друга фаза відбувається щоразу, коли користувачеві необхідно використовувати систему. У цей час система збирає дані про час натискання клавіш користувачем, а потім витягує тимчасові характеристики так само як і на етапі реєстрації. Після цього система виконує зіставлення ознак із шаблоном користувача, який зберігається у базі даних. Далі, виходячи з результатів процесу зіставлення, відбувається одна з двох дій: надання доступу користувачеві, якщо два набори даних досить схожі, або відмова у доступі користувачеві в іншому випадку.

Переваги аутентифікації за клавіатурним почерком:

- час натискання клавіш можна виміряти з точністю до мілісекунд за допомогою програмного забезпечення, таким чином, відтворити свій малюнок натискання клавіш з такою високою роздільною здатністю без величезних зусиль практично неможливо;
- на відміну від інших біометричних систем, які залежать від спеціалізованих пристроїв та апаратної інфраструктури, розпізнавання динаміки натискання клавіш повністю реалізовано програмно, через це знижується вартість розгортання та створює ідеальний сценарій для впровадження у середу віддаленої аутентифікації;
- шаблони натискання клавіш складніше відтворити, ніж письмові підписи.
- біометрія динаміки натискання клавіш пропонує спосіб безперервного підтвердження легітимної особи користувача.

Недоліки аутентифікації за клавіатурним почерком:

- біометрія за динамікою натискання клавіш поступається за точністю аутентифікації через варіації ритму набору тексту, викликаних зовнішніми факторами, такими як травма, втома або відволікання уваги;
- манера набору тексту людини може поступово змінюватися внаслідок звикання до пароля, розвитку навички набору тексту, адаптації до пристроїв введення та інших факторів довкілля.

1.3 Порівняння видів біометричної аутентифікації

Виконаємо порівняння методів біометричної аутентифікації на основі наступних п'яти характеристик:

- безпека – це ефективність системи протистояти потенційним атакам, враховуючи ризик, який вони представляють, та її складність;
- точність – правильність проходження користувачем аутентифікації;
- сталість – це умова, що біометричні дані не повинні змінюватися з часом;
- зручність використання – це якість зручності та наближення до потреб і вимог користувачів;
- витрати – економічний вплив технології на загальну систему аутентифікації, наприклад витрати на впровадження, обслуговування тощо.

Через відмінності в середовищі, де збираються дані, або між зчитувачами, які використовують біометричні дані, неможливо досягти 100% точності. Таким чином, необхідно визначити певні пороги продуктивності, щоб розглянути надійну біометричну технологію. Дві звичайні показники, які використовуються для оцінки ефективності біометричних даних, – це FAR (коефіцієнт хибного сприйняття) і FRR (коефіцієнт помилкового відхилення) [28];

У наведеній нижче таблиці 1.1 показано коротке порівняння біометричних типів, поданих у цьому розділі за трьома рівнями – високим, середнім та низьким.

Таблиця 1.1 – Порівняння біометричних методів аутентифікації

Характеристика Вид аутентифікації	Безпека	Точність	Сталість	Зручність використання	Вартість
За відбитком пальця	В	В	В	В	С
За райдужкою ока	В	В	В	С	В
За сітківкою ока	В	В	С	Н	В
За геометрією кисті	В	С	С	С	В
За геометрією обличчя	С	С	С	В	Н
За термограмою обличчя	В	С	С	В	С
За голосом	С	С	Н	В	Н
За рукописним підписом	С	С	Н	С	С
За клавіатурним почерком	В	С	С	В	Н

Джерело: розроблено автором за даними [29]

Отже, провівши аналіз приведених методів біометричної аутентифікації можна дійти висновку, що узагальнена схема для всіх методів має вигляд наведений на рисунку 1.9.

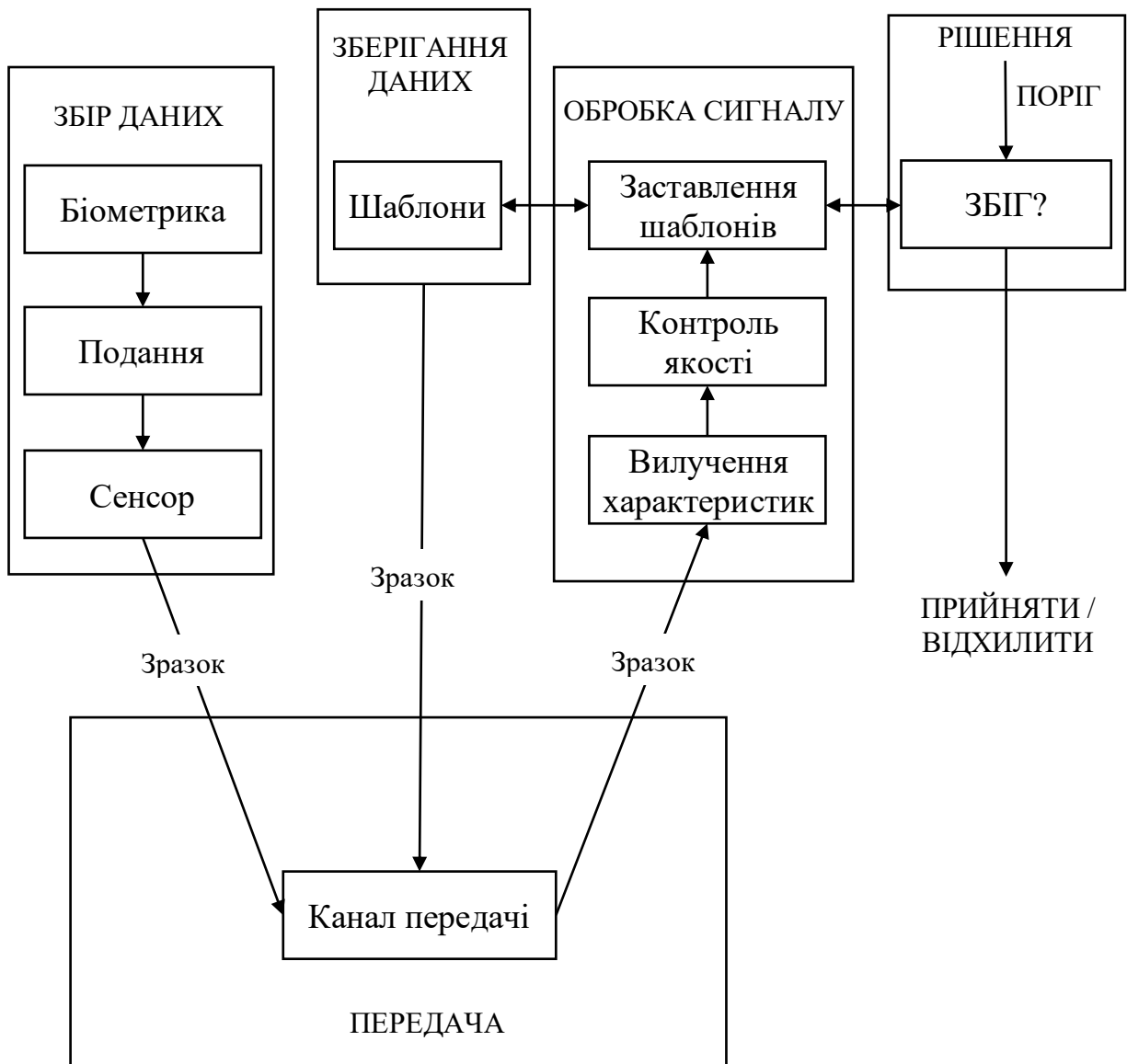


Рисунок 1.9 – Базова структура системи біометричної аутентифікації

Джерело: авторська розробка беручи до уваги вищевикладене

1.4 Висновки за розділом

В першому розділі розглянуто поняття біометричної аутентифікації. Аутентифікація безпеки на основі біометричних даних отримала значну увагу завдяки своїй точності, надійності, універсальності, сталості тощо. У розділі наведено огляд найбільш поширених методів біометричної аутентифікації та узагальнена схема для всіх методів біометричної аутентифікації. Крім того, розкриваються основні переваги та недоліки кожного методу.

Також було проведено порівняльний аналіз методів біометричної аутентифікації. Немає системи безпеки, яка була б абсолютно надійною.

Кожну систему можна зламати, витративши відповідну кількість часу та грошей. Відбитки пальців легко виявити на поверхні, до якої торкалися, і скопіювати за невеликий час, використовуючи доступні матеріали.

Детектор активності в системах розпізнавання облич також можна перемогти, використовуючи відеокліпи з особами та відтворюючи їх.

Але системи біометричної аутентифікації за клавіатурним почерком дуже важко підробити. Час натискань клавіш може бути точним до мілісекунди за допомогою програмного забезпечення, що робить майже неможливим відтворення клавіатурного почерку з високою роздільною здатністю без величезних зусиль.

Таким чином, системи з клавіатурним почерком можуть бути використані для додатків з високим рівнем безпеки та мережевої безпеки.

У наступному розділі розробимо власну методику проведення клавіатурної аутентифікації.

Перед тим, як створювати математичну модель клавіатурного почерку користувача проведемо аналіз вже розроблених систем клавіатурної аутентифікації. Загальна статистика по проведених аналізах наведено в таблиці 2.1.

Автор	В. Г. Абашин.	Р. Р. Шаріпов;	О.	Ю. А.
Назва	Автоматизація процесу визначення психофізіологічного стану оператора автоматизованого робочого місця в АСУТП [30]	Розробка полігаусового алгоритму аутентифікації користувачів в телекомунікаційних системах та мережах по клавіатурному почерку [31]	Ходашинський, М. В. Савчук, І. В. Горбунов, Р. В. Мещеряков; технологія посиленої аутентифікації користувачів інформаційних процесів [32]	Брюхоміцький; гістограмний метод розпізнавання клавіатурного почерку [33]
Час утримання клавiш	-	+	+	+
Швидкість натискання	-	+	-	-

2.2 Математична модель клавіатурного почерку користувача

2.2.1 Функція процесу набору тексту

Процес набору тексту користувачем можна описати функцією (2.1):

$$\vartheta(t) = \alpha(t) + \beta(t) + \gamma(t), \quad (2.1)$$

де $\alpha(t)$ – компонента, що характеризує підсвідомі процеси мислення під час набору тексту;

$\beta(t)$ – компонента, що характеризує свідомі процеси мислення;

$\gamma(t)$ – механічні властивості клавіатури, які впливають на процес набору тексту.

Відповідно до особливостей клавіатурного почерку основним завданням системи біометричної аутентифікації користувача є завдання виділення та подальшого виявлення компоненти $\alpha(t)$, що характеризує підсвідомий процес мислення. Це завдання визначає вихідні дані системи ідентифікації після вимірювання тимчасових характеристик клавіатурного почерку користувача.

Для цього необхідно вибрати компоненти $\beta(t)$ та $\gamma(t)$ із вихідної функції $\vartheta(t)$. Оскільки немає способу змодельовати механіку руху людей, що друкують, очевидно, єдиним прийнятним рішенням є проведення збору статистичних даних клавіатурного почерку великої кількості користувачів і побудова емпіричних залежностей $\beta(t)$ та $\gamma(t)$.

Під час введення логіна або пароля ймовірність аутентифікації користувача за інтервалом часу між натисканнями клавіш є більш показовою ознакою клавіатурного почерку користувача, ніж час утримання клавіш.

Це тому, що процес, який користувачі часто використовують для введення коротких повідомлень на клавіатурі, є справжнім підсвідомим процесом мислення (м'язова пам'ять, як рукописні людські підписи). Підсвідомі рухи стабільні до тих пір, поки їх не порушує мислення вищого свідомого рівня, що призводить до ефекту «сороконіжки», який втрачається при спробі зрозуміти, як вона ходить. Продуктивність цього ефекту пояснює зниження ймовірності аутентифікації користувача при перевищенні довжини

ключової фрази певного критичного рівня. Важливо зазначити, що значення цього порогу сильно варіюється для користувачів із різним досвідом роботи з клавіатурою і може становити від 6 до 30 символів. Після цього навіть досвідчені друкарі мають ефект, який передбачає свідоме мислення та припинення друку для прийняття рішень [35].

Згідно з наведеним, можна дійти висновку, що через характер клавіатурного рукописного введення не рекомендується використовувати довгі вирази як ключові слова в системі аутентифікації користувачів, оскільки це призведе до того, що користувачі почнуть вводити текст «з розумом», що призведе до зниження якості їх аутентифікацію.

Отже, можна сказати, що час утримання клавіш найкраще передає властивості $\beta(t)$ функції $\vartheta(t)$. Вилучення компоненти $\beta(t)$ з функції $\vartheta(t)$ досягається введенням кореляційної функції $\Delta(t)$, яка отримана емпірично на основі аналізу наявної статистики. Оскільки характер кореляційної функції $\Delta(t)$ слід вибирати відповідно до досвіду роботи з клавіатурою користувача, необхідно ввести вхідний параметр L , який визначає відповідність між кореляційною функцією і складовою $\beta(t)$ [36].

Дослідження параметрів клавіатурного почерку одного користувача на різних клавіатурах дають невеликий розкид ймовірності ідентифікації, що можна пояснити випадковими причинами.

Отже, цей результат дозволяє стверджувати, що механічні властивості клавіатури практично не впливають на характеристики почерку клавіатури користувача, тому $\gamma(t)$ у виразі (2.1) можна знехтувати. Враховуючи вищевикладене, ми можемо записати функцію $\vartheta(t, L)$ так:

$$\vartheta(t, L) = \alpha(t) + \Delta(t, L). \quad (2.2)$$

Очевидно, що клавіатурний почерк користувача може змінюватися з часом. Це можна виразити так, що під час виконання i -го процесу ідентифікації ми отримуємо ймовірність ідентифікації p_i , яка на деяке значення ξ відрізняється від математичного сподівання ймовірності m_p , отриманого під час попередньої операції ідентифікації користувача.

З огляду на це, можна записати рівняння (2.2) так:

$$\vartheta(t, L) = (\alpha(t) + \Delta(t, L))T(t, L), \quad (2.3)$$

де $T(t, L)$ – функція кореляції, яка враховує залежні від часу зміни параметрів набору тексту користувачем.

2.2.2 Вектор вхідних параметрів

В даній роботі для розв'язання задачі аутентифікації користувача запропонуємо вектор біометричних параметрів:

$$V = \{t_1^{\text{пауза}}, t_2^{\text{пауза}}, \dots, t_{n-1}^{\text{пауза}}, m^{\text{пауза}}, \beta, s, d_1, d_2\}, \quad (2.4)$$

де $t_i^{\text{пауза}}$ – час між відпусканням i -ої клавіші та натисканні $(i + 1)$ -ої;

$m^{\text{пауза}}$ – нормоване математичне сподівання довжини інтервалів;

β – розкид часу між натисканням двох клавіш;

s – швидкість вводу;

y_1, y_2 – параметри зміни часових інтервалів.

Для вирахування довжини паузи між відпусканням i -ої клавіші та натисканні $(i + 1)$ -ої ($t_i^{\text{пауза}}$) та часу утримування ($t_i^{\text{утрим}}$) використовуємо виміряні $t_i^{\text{натиск}}$ та $t_i^{\text{відпуск}}$ – час натискання та відпускання i -ої клавіші відповідно.

$$t_i^{\text{утрим}} = t_i^{\text{відпуск}} - t_i^{\text{натиск}}, \quad i = 1 \dots n, \quad (2.5)$$

$$t_i^{\text{пауза}} = t_{i+1}^{\text{натиск}} - t_i^{\text{відпуск}}, \quad i = 1 \dots (n - 1), \quad (2.6)$$

де $t_i^{\text{утрим}}$ – час утримування i -ої клавіші в натиснутому стані;

$t_i^{\text{пауза}}$ – час паузи між відпусканням i -ої клавіші та натисканням $(i + 1)$ -ої;

n – довжина ключової фрази в випадку даної роботи – логіна.

Нормоване математичне сподівання $m^{\text{пауза}}$ виглядає так:

$$m^{\text{пауза}} = \frac{\sum_{i=1}^{n-1} \frac{t_i^{\text{пауза}}}{t_{\text{max}}^{\text{пауза}}}}{n - 1}. \quad (2.7)$$

Розкид часу між натисканням двох клавіш виглядає наступним чином:

$$\beta = \sqrt{\frac{\sum_{i=1}^{n-1} \left(\frac{t_i^{\text{пауза}}}{t_{\text{max}}} - m^{\text{пауза}} \right)^2}{n-2}} . \quad (2.8)$$

Швидкість набору користувачем логіна розраховується як:

$$s = \frac{\sum_{i=1}^{n-1} t_i^{\text{пауза}} + \sum_{i=1}^n t_i^{\text{утрим}}}{60} . \quad (2.9)$$

Для того, щоб простежити зміни інтервалів використовуються два параметри y_1 та y_2 . Перед початком їх знаходження потрібно вирахувати різницю між часом наступного та поточного інтервалів.

$$dt_i = t_{i+1}^{\text{пауза}} - t_i^{\text{пауза}}, i = 1 \dots n-1. \quad (2.10)$$

Відношення числа позитивних різниць до спільного числа різниць (параметр k_1) рахується наступним шляхом: нехай f – кількість різниць dt_i зі знаком плюс, k – кількість різниць. Тоді відношення має вигляд:

$$y_1 = \frac{f}{k} = \frac{f}{n-1} . \quad (2.11)$$

Відношення числа змін знака в dt_i до загального числа dt_i (параметр y_2) вираховується наступним чином: нехай h – кількість змін знака, а g – максимальне число змін знаків. Тоді відношення кількості змін знака dt_i до загальної кількості dt_i :

$$y_2 = \frac{h}{g} = \frac{h}{n-2} . \quad (2.12)$$

В результаті отримуємо вектор вхідних параметрів V , що використовується для аутентифікації користувача за клавіатурним почерком при фіксованій довжині ключової фрази.

2.3 Опис процесу біометричної аутентифікації користувача

Система клавіатурної аутентифікації функціонує в режимі настроювання та аналізу.

В режимі настроювання визначаються та зберігаються еталонні параметри клавіатурного почерку користувача, в режимі аналізу система порівнює еталонні параметри з введеними, після чого відбувається вибір з

двох гіпотез:

- користувач, котрий вводить логін є авторизованим;
- користувач розпізнаний як не є авторизованим.

В режимі настроювання користувач вводить певну L кількість раз ключову фразу, в випадку даної роботи – логін, тобто проводить етап реєстрації (навчання системи). Ці L логінів користувача відповідають L реалізаціям вектора біометричних параметрів $V = \{V_1, V_2, \dots, V_L\}$.

В режимі аналізу відбувається аутентифікація користувача, котрий ввів логін. Процедури аутентифікації користувача, що надав свої біометричні параметри в вигляді вектору V , можна роздивлятись як задачу класифікації вектору V на $(M + 1)$ класів по числу M зареєстрованих в системі користувачів («свої»), та плюс один клас для інших не зареєстрованих («чужі»).

Однак попередньо зареєстровані в системі біометричні параметри користувача можуть використовуватися як параметри класифікатора.

Це дозволяє привести задачу класифікування вектору V на $(M + 1)$ до задачі класифікувати тільки на два класи: «свої» – вектор V_C та «чужі» – вектор $V_{\text{ч}}$.

Для класифікації користувачів на два класи «свої» (вектор V_C) та «чужі» (вектор $V_{\text{ч}}$) параметричний класифікатор можна реалізувати з використанням лише однієї функції $g(V)$, знак якої визначатиме належність пред'явленого вектора V до одного з двох класів: V_C і $V_{\text{ч}}$.

Формування функції $g(V)$, розділяє області «свої» і «всі чужі» проводиться з наступних міркувань.

Нехай у загальному випадку область розподілу біометричних параметрів «свого» користувача задана безліччю зразків ω_C , що складається з векторів L $V_{Ci}, i = 1 \dots L$, нормально розподілених в N -мірному просторі ортогональної системи координат, а кожен вектор $V_{Ci}, i = 1 \dots L$ представлений своїми N компонентами:

$$V_{Ci} = \{v_1, v_2, \dots, v_j, \dots, v_N\}, j = 1 \dots N. \quad (2.14)$$

Центр розподілу векторів V_{Ci} розташовується в точці $(\xi_1, \xi_2, \dots, \xi_N)$, яка

визначається N математичними сподіваннями $m_{v_1} = \xi_1, m_{v_2} = \xi_2, m_{v_3} = \xi_3, \dots, m_{v_N} = \xi_N$.

Центральні моменти другого порядку розподілу векторів V_{Ci} утворюють квадратну матрицю моментів (коваріаційну матрицю) [37].

$$Q = \|\lambda_{jk}\| = \left\| \begin{pmatrix} \lambda_{11} & \dots & \lambda_{1N} \\ \dots & \dots & \dots \\ \lambda_{N1} & \dots & \lambda_{NN} \end{pmatrix} \right\|,$$

де

$$\lambda_{jk} = M(v_j - \xi_j) \cdot (v_k - \xi_k) = \begin{cases} \sigma_{ij}^2 & \text{при } j = k \\ cov\{v_j, v_k\} & \text{при } j \neq k \end{cases} \quad j, k = 1..N. \quad (2.16)$$

Для нормального закону розподілення N -мірних випадкових корельованих величин, що також називають розподіленням Гауса, функція щільності розподілення має вигляд:

$$f(v_1, v_2, \dots, v_N) = \frac{1}{\sqrt{(2\pi)^N \cdot \det\|\lambda_{jk}\|}} e^{\left[-\frac{1}{2} \sum_{j=1}^N \sum_{k=1}^N \lambda_{jk} \cdot (v_j - \xi_j) \cdot (v_k - \xi_k)\right]}, \quad (2.17)$$

де $\det\|\lambda_{jk}\|$ – визначник коваріаційної матриці Q , коефіцієнти λ_{jk} утворюють матрицю $\lambda = \lambda_{jk}$, зворотну коваріаційної матриці Q [38].

Для знаходження коефіцієнтів λ_{jk} використовується стандартна формула:

$$\lambda_{jk} = (-1)^{j+k} \cdot \frac{M_{jk}}{\det\|\lambda_{jk}\|}, \quad (2.18)$$

де M_{jk} – мінор визначника λ_{jk} , що отримується з нього шляхом викреслювання j -го рядка та k -го стовпця [39].

Гіпереліпсоїд розсіювання має рівну щільність розподілення N -мірних випадкових величин, тому його можна отримати з умови [39]:

$$f(v_1, v_2, \dots, v_N) = \text{const}. \quad (2.19)$$

Вираз, що фігурує в експоненті функції щільності нормального розподілення векторів V_{Ci} є додатньовизначеною квадратичною формою. Поверхня на якій ця квадратична форма постійна:

$$\frac{1}{2} \cdot \sum_{j=1}^N \sum_{k=1}^N \Lambda_{jk} \cdot (v_j - \xi_j) \cdot (v_k - \xi_j) = \text{const}, \quad (2.20)$$

є поверхнями рівних щільностей ймовірностей у N -мірному просторі і являють собою гіпереліпсоїди, які групуються навколо точки $(\xi_1, \xi_2, \dots, \xi_N)$.

Позначаючи константу у правій частині виразу (2.20) через k^2 , отримаємо:

$$\frac{1}{2} \cdot \sum_{j=1}^N \sum_{k=1}^N \Lambda_{jk} \cdot (v_j - \xi_j) \cdot (v_k - \xi_j) = k^2. \quad (2.21)$$

Константа k задає коефіцієнт пропорційності між довжинами a_j головних півосей гіпереліпсоїда та відповідними середньоквадратичними відхиленнями σ_j ;

$$a_1 = k \cdot \sigma_1, a_2 = k \cdot \sigma_2, \dots, a_N = k \cdot \sigma_N. \quad (2.22)$$

Для оптимального вирішення задачі класифікації з усіх поверхонь рівних щільностей ймовірностей доцільно вибрати ту, яка характеризує розсіювання векторів V_{Ci} щодо точки $(\xi_1, \xi_2, \dots, \xi_N)$. Ця поверхня відповідає так званому одиничному гіпереліпсоїду, у якого головні півосі рівні відповідним середньоквадратичним відхиленням $\sigma_1, \sigma_2, \dots, \sigma_N$, тобто для одиничного гіпереліпсоїда $k = 1$, та вираз (2.21) перетворюється на вид:

$$\frac{1}{2} \cdot \sum_{j=1}^N \sum_{k=1}^N \Lambda_{jk} \cdot (v_j - \xi_j) \cdot (v_k - \xi_j) = 1. \quad (2.23)$$

Через обмежену статистику біометричних зразків, що пред'являються на стадії реєстрації «своїм» користувачем, завжди залишається ймовірність того, що зразок, пред'явлений цим же користувачем при аутентифікації, вийде за межі зафіксованого в ідеалі діапазону.

Для зменшення цієї ймовірності додатково задається величина допуску між областями «свої» та «усі чужі» у вигляді коефіцієнта Стюдента $C[L_1, (1 - P_1)]$ (критичні значення, якого наведені в додатку Б для різної довірчої ймовірності P та числа ступенів свободи L), виходячи із заданої помилки першого роду (ймовірність P_1 помилкової відмови «своєму»

$$g(V) > 0, \text{ якщо } V \in V_q.$$

Таким чином, процедура аутентифікації зводиться тепер до перевірки: чи попадає пред'явлений користувачем вектор біометричних параметрів V в область, що описується, виразом (2.25).

2.4 Висновки за розділом

Таким чином, для аутентифікації користувача за клавіатурним почерком при введенні логіна або пароля необхідно провести розпізнавання за швидкістю введення та динамікою введення (часом між натисканнями клавіш і часом їх утримання). Тимчасові інтервали між натисканнями клавіш на клавіатурі та час утримання (натискання) клавіш дозволяють досить однозначно охарактеризувати почерк роботи користувача на клавіатурі, що підтверджується рядом експериментів, проведених автором під час дослідження особливостей аутентифікації користувача.

Система клавіатурної аутентифікації повинна функціонувати в режимі настроювання та аналізу.

В режимі настроювання визначаються та зберігаються еталонні параметри клавіатурного почерку користувача, в режимі аналізу система порівнює еталонні параметри з введеними.

У загальному випадку задача класифікування вектору вхідних параметрів зводить до тільки двох класів: «свої» та «чужі». В розділі 3 розробимо програмний модуль, що відображає дану задачу.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Вибір засобів реалізації програмного модуля

Для реалізації модулю системи біометричної аутентифікації за клавіатурним почерком вибрано середу розробки Visual Studio 2022, мову програмування C#, тип проєкту WinForms.

Microsoft Visual Studio – це професійний інструмент розробки, який виявився повністю інтегрованим середовищем розробки для Visual Basic, Visual C++, Visual J#, and Visual C#. У його IDE можна записати код Visual Basic, C++, J#, або C# для створення додатків Windows [42].

Це середовище є ефективним та надійним. Правила надійності підтримують безвідмовність бібліотек та програм, наприклад шляхом правильного використання пам'яті та потоків.

Для вирішення поставленого завдання необхідно використати функціональну, ефективну та зручну платформу розробки, що дозволяє застосовувати принципи об'єктно-орієнтованого програмування. Як таку платформу було обрано середовище .NET.

Вибір середовища розробки Microsoft Visual Studio 2022 був заснований на кількох критеріях:

- підтримка мови C#;
- інтуїтивно зрозумілий інтерфейс;
- функції графічного редактора, відповідних для виконання завдання.

Отже, за допомогою середовища розробки Microsoft Visual Studio 2022 виконуються задані умови.

C# – це об'єктно-орієнтована мова програмування від Microsoft, яка зосереджена на поєднанні обчислювальної потужності C++ із легкістю програмування Visual Basic. C# заснований на C++ і включає функції, подібні до Java. C# зазвичай використовується для серверних служб, додатків

Windows, розробки вебсайтів та ігор [43].

Середовище розробки Microsoft Visual Studio, яке постачається з .NET, надає інструменти, необхідні для ефективного та швидкого створення програм із графічним інтерфейсом.

Хоча C# і .NET в основному використовуються для веброзробки, вони також активно використовуються для створення програм, які повинні бути встановлені на комп'ютерах кінцевих користувачів, де буде виконуватися вся обробка даних. Розробку таких програм забезпечує бібліотека Windows Forms, яка дозволяє проєктувати графічні інтерфейси. Модуль системи, описаний в цій роботі, був розроблений за допомогою Windows Forms.

Windows Forms – це безкоштовна загальнодоступна бібліотека графічних класів від Microsoft .NET Framework, яка використовується для написання розширених клієнтських програм для ноутбуків, настільних комп'ютерів і планшетів. На відміну від пакетної програми, вона проводить більшу частину свого часу в очікуванні введення користувача, наприклад, заповнення текстового поля або натискання кнопок [44].

3.2 Опис практичної реалізації

Для проведення дослідження особливостей параметрів клавіатурного почерку при введенні логіна була обрана система динамічної аутентифікації з додаванням клавіатурного почерку шляхом використання вектору біометричних параметрів користувача, що описаний в розділі 2 даної роботи.

Етапи розробки та створення системи біометричної аутентифікації за клавіатурним почерком показаний на рисунку 3.1.



Рисунок 3.1 – Етапи розробки та створення системи біометричної аутентифікації за клавiатурним почерком

Джерело: авторська розробка

Для даного аналізу особливостей параметрів аутентифікації користувача за клавiатурним почерком розроблено програмний модуль, що дозволяє провести аналіз вхідних параметрів клавiатурного почерку окремо взятого користувача.

Для цього на головному екрані модуля виводиться вікно, в якому потрібно ввести логін.

При його введенні з'являється дві події: кнопка натиснута або відпущена.

Тому, щоб підрахувати час утримання певної кнопки, потрібно від часу, коли кнопка відпущена, відняти час, коли сталася кнопка натиснута. Для обчислення проміжків між натисканнями необхідно від часу, коли натиснута відповідна клавіша відняти час, коли попередня клавіша відпущена.

Оскільки ключова фраза вводиться в текстове поле (textbox), використовуються обробники подій `textBox_KeyUp` і `textBox_KeyDown` для подій коли кнопка відпущена або натиснута відповідно.

На рисунку 3.2 схематично зображені параметри введення логіна з 5 символів.

За d1 приймається подія, коли кнопка натиснута, d2 – кнопка відпущена.

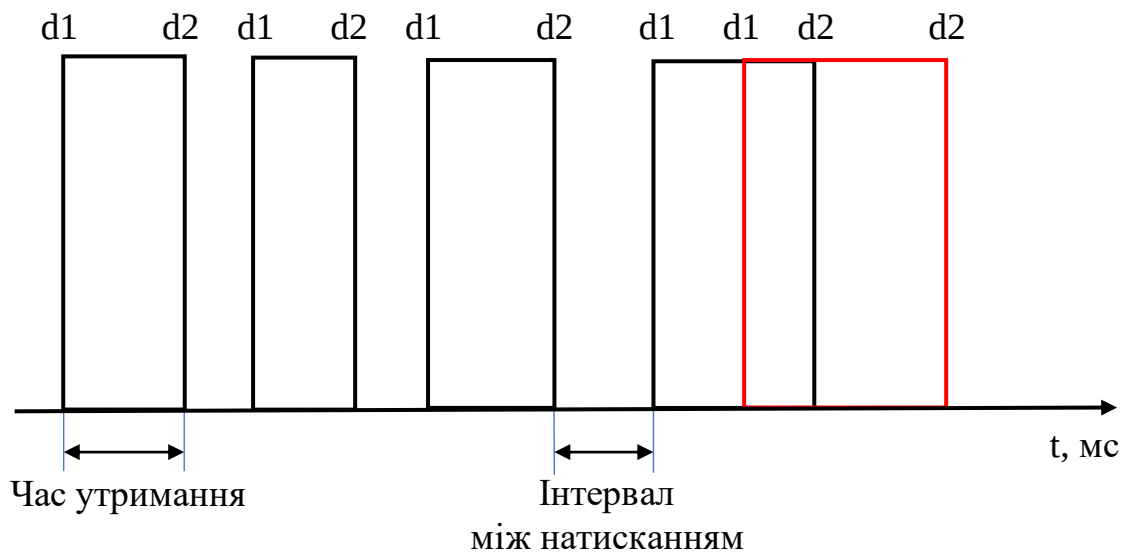


Рисунок 3.2 – Схематичне зображення 5-символьного слова

Джерело: авторська розробка

На рисунку 3.2 зображено також виняткова ситуація: 5 клавіша натиснута ще до того, як відпущена 4. В цьому випадку виникає негативне значення для проміжку. Важливо тут отримати коректне значення проміжку, оскільки значення події, коли кнопка натиснута для вже затисненої клавіші "затирає" час для попередньої.

Ця проблема вирішується використанням колекції об'єктів типу черга (Queue). У черзі реалізовано принцип FIFO (First In First Out). Тому, поміщаючи в неї дані подій, коли кнопка натиснута, можна бути впевненим, що при обчисленні часу утримання від часу, коли кнопка відпущена віднімається час, коли кнопка натиснута саме цієї клавіші, а не будь-якої іншої [45].

Для вимірювання часу використовується клас Stopwatch, що надає набір методів та засобів, що застосовуються для точного підрахунку витраченого часу [46].

Робота з обчисленням інтервалів здійснюється так:

- формується об'єкт класу Stopwatch;

- при відкритті форми запускається вимір часу (метод Start);
- під час натискання клавіші повертається точний час на цей момент, отриманий результат поміщають у чергу;
- при відпусканні кнопки з отриманого результату віднімається той час, що лежить у черзі першим, також розраховується час інтервалу між натисканнями поточної та попередньої клавіші;
- при закритті форми викликається метод Stop, що викликає зупинку виміру.

Якщо користувач натисне кнопку з написом «Подивитись», то перед ним з'являться результати останнього вводу – інтервали між сусідніми клавішами та час утримання кожної з них.

Результат подається у вигляді таблиці зі стовпцями: введений символ, час утримання поточної та попередньої клавіш. Рисунок 3.3 є прикладом для введення пароля «eliza».

Кількість інтервалів між клавішами та часу утримання кожної клавіші буде відповідати кількості введених символів у логіні. У полі інтервалу між клавішами навпроти першої клавіші завжди буде нуль.

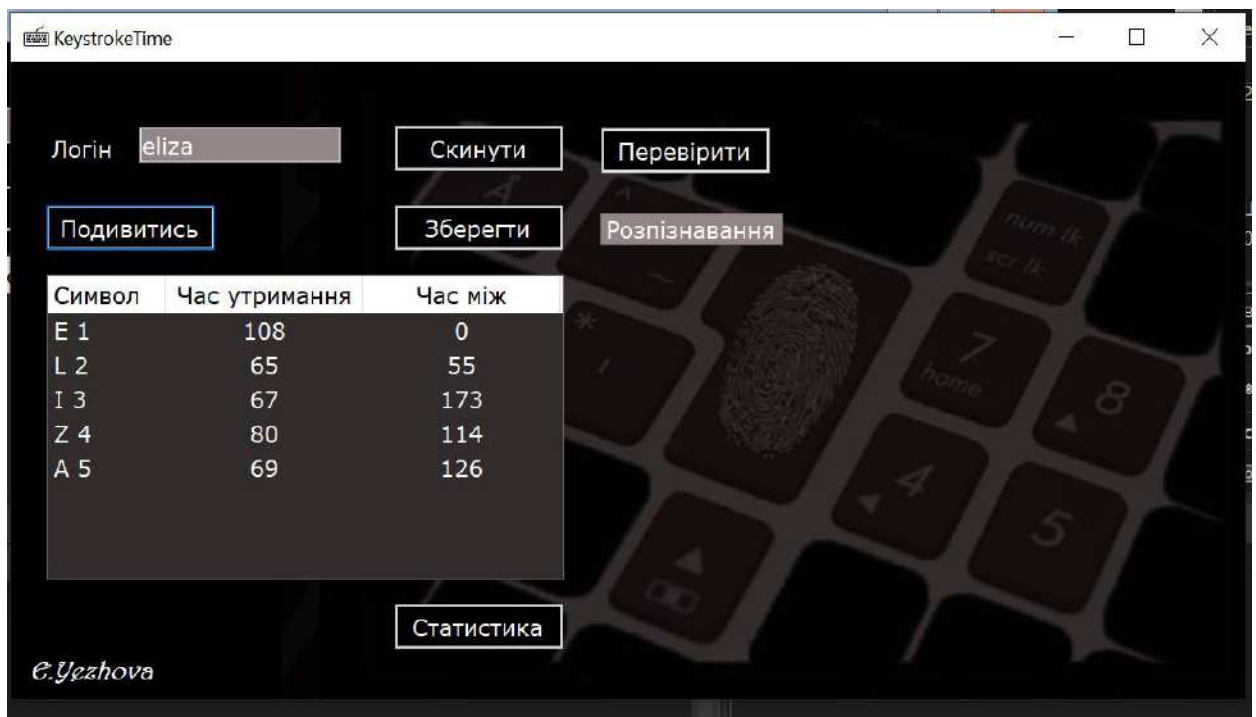


Рисунок 3.3 – Вхідні параметри вводу

Також за допомогою даного програмного модуля можна провести статистичний збір за серіями вводу. Після того як користувач вводить логін довільну кількість раз, натискаючи відповідну кнопку «Зберегти», можна переглянути зібрану статистику натискаючи кнопку «Статистика».

Дана процедура зчитує відповідні дані та далі обробляє їх. Розраховується математичне сподівання, за допомогою функції $M()$, дисперсію, за допомогою функції $D()$, середнє квадратичне відхилення, також рахується матриця кореляції та коваріації, за допомогою функцій $Correl()$ та $Covariation()$ відповідно та довірчі інтервали, за допомогою функції $Dover()$. Результати зберігаються в файли `time_between_results`, `time_press_results` відповідно.

Відповідно до попередньо отриманих даних, розроблена динамічна модель аутентифікація користувача за клавіатурним почерком. Написаний програмний модуль мовою програмування C# з використанням типу додатку WinForms на платформі .NET.

Система динамічної аутентифікації за клавіатурним почерком функціонує у двох режимах: налаштування та аналізу.

Режим налаштування – у ньому визначаються та зберігаються еталонні характеристики клавіатурного почерку авторизованого користувача.

Режим аналізу – у ньому система порівнює еталонні характеристики зі знову введеними по формулі (2.23), після чого вирішує завдання вибору двох гіпотез, після чого у вікні додатку виводиться відповідний результат «АВТОРСЬКА» чи «НЕ АВТОРСЬКА» спроба вводу.

Схема роботи даного програмного модуля наведено на рисунку 3.4.

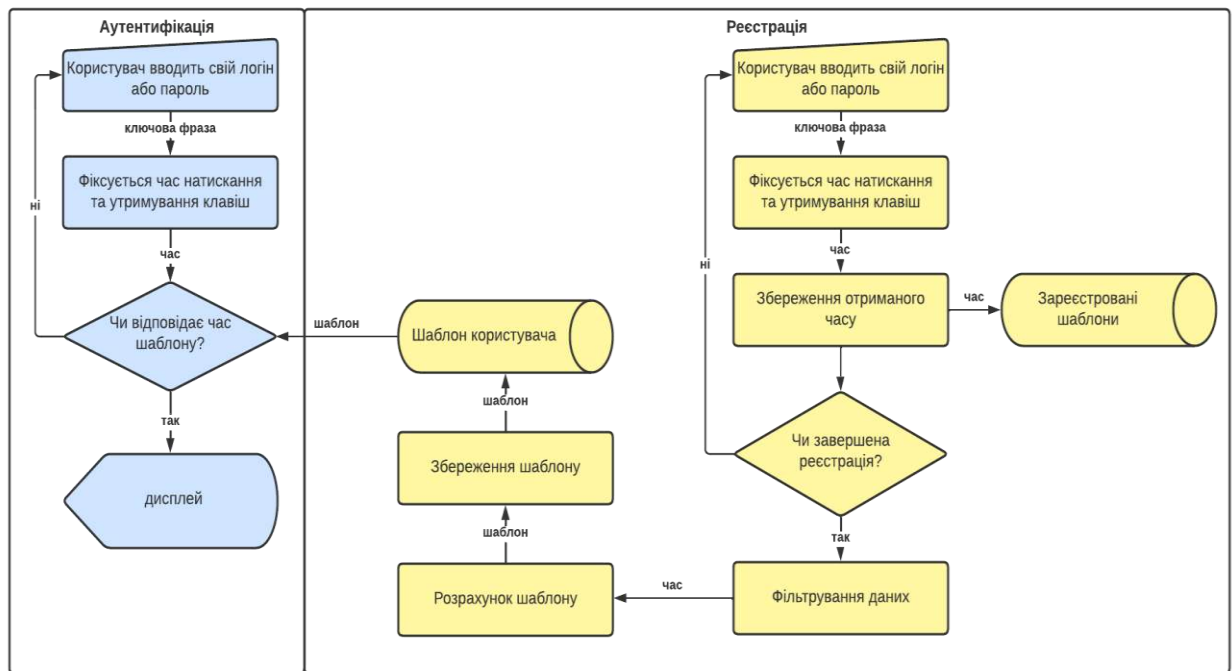


Рисунок 3.4 – Схема програмного модуля

Джерело: авторська розробка беручи до уваги вищевикладене

3.3 Висновки за розділом

На початку розділу вибрано платформу для реалізації програмного модуля, з позиції функціональності, ефективності та зручності, яка дозволяє застосовувати принципи об'єктно-орієнтованого програмування. Як таку платформу було обрано середовище .NET.

Наведено етапи розробки та створення системи біометричної аутентифікації за клавіатурним почерком.

Описано програмну реалізацію методики, що була розроблена в розділі 2, виділено режими роботи розробленого модуля.

Побудована схема роботи програмного модуля з використанням мови моделювання UML.

У четвертому розділі опишемо тестування працездатності програмного модуля та проведемо аналіз результатів дослідження параметрів клавіатурного почерку.

4 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

4.1 Результати дослідження параметрів клавіатурного почерку

Аналіз властивостей параметрів клавіатурного почерку користувача проходив у декілька етапів, далі наведено їх детальний опис.

На першому етапі аналізувався клавіатурний почерк автора даної роботи. Для цього довільну кількість раз вводився логін (100 разів), який автор використовує вже тривалий термін, зберігаючи кожну спробу вводу.

Введені дані утворили два масиви: перший – інтервали між натисканням клавіш, другий – час утримування клавіш при натиску (значення масивів наведені в додатку В пункті В.1 та В.2 відповідно). Після цього проводився аналіз зібраних параметрів.

Результати проведеного аналізу можна роздивитися на рисунку 4.1 – для інтервалів між натисканням клавіш та 4.2 – для часу утримування клавіш при натиску.

На даних рисунках різними кольорами вказані зображення 100 варіантів вводу та червоним кольором визначено математичне сподівання.

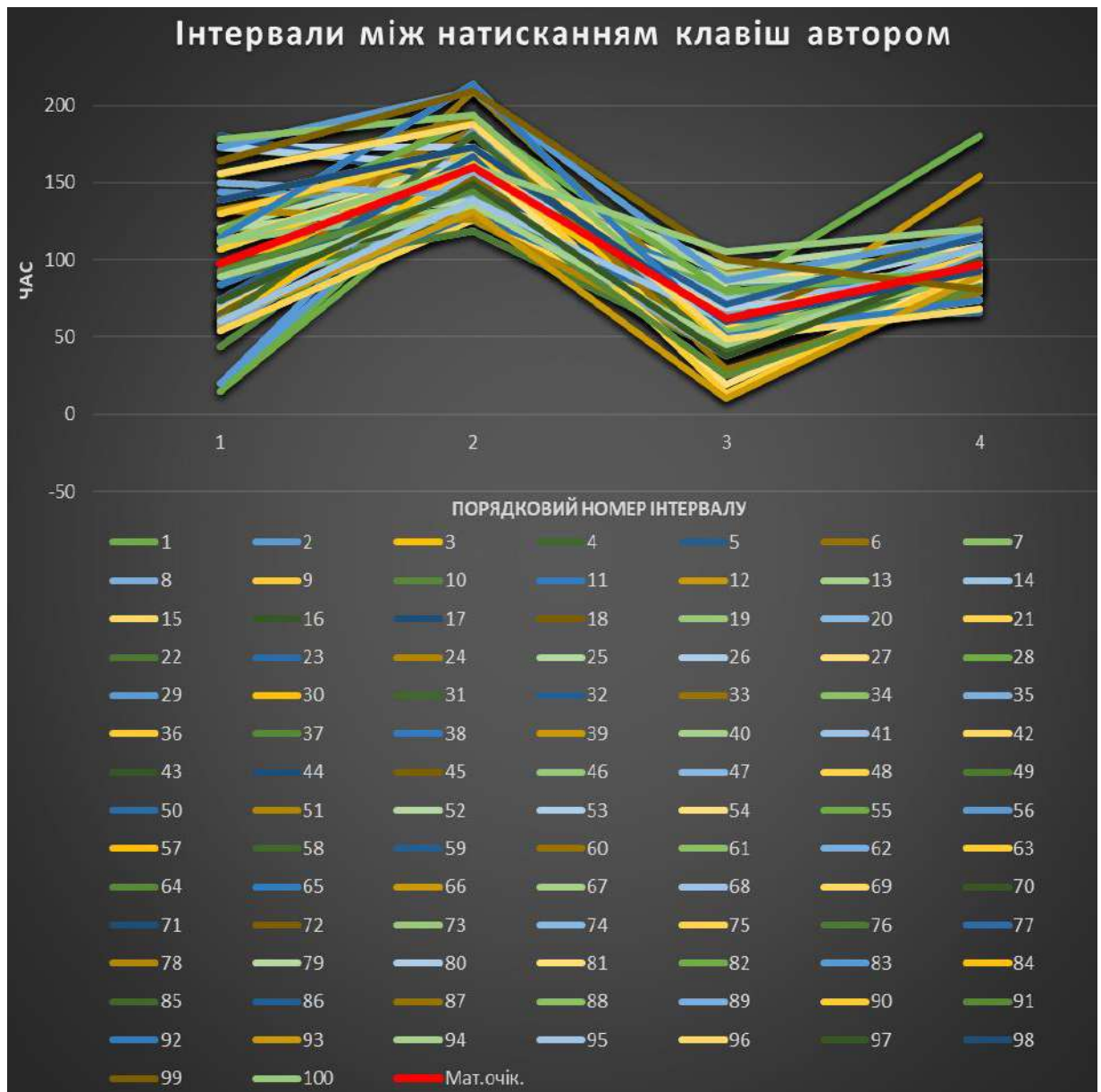


Рисунок 4.1 – Діаграма часових інтервалів між натисканням клавіш

На вертикальній осі рисунку 4.1 вказується час між натисканням клавіш автором в мілісекундах, по горизонтальній – порядковий номер інтервалу.

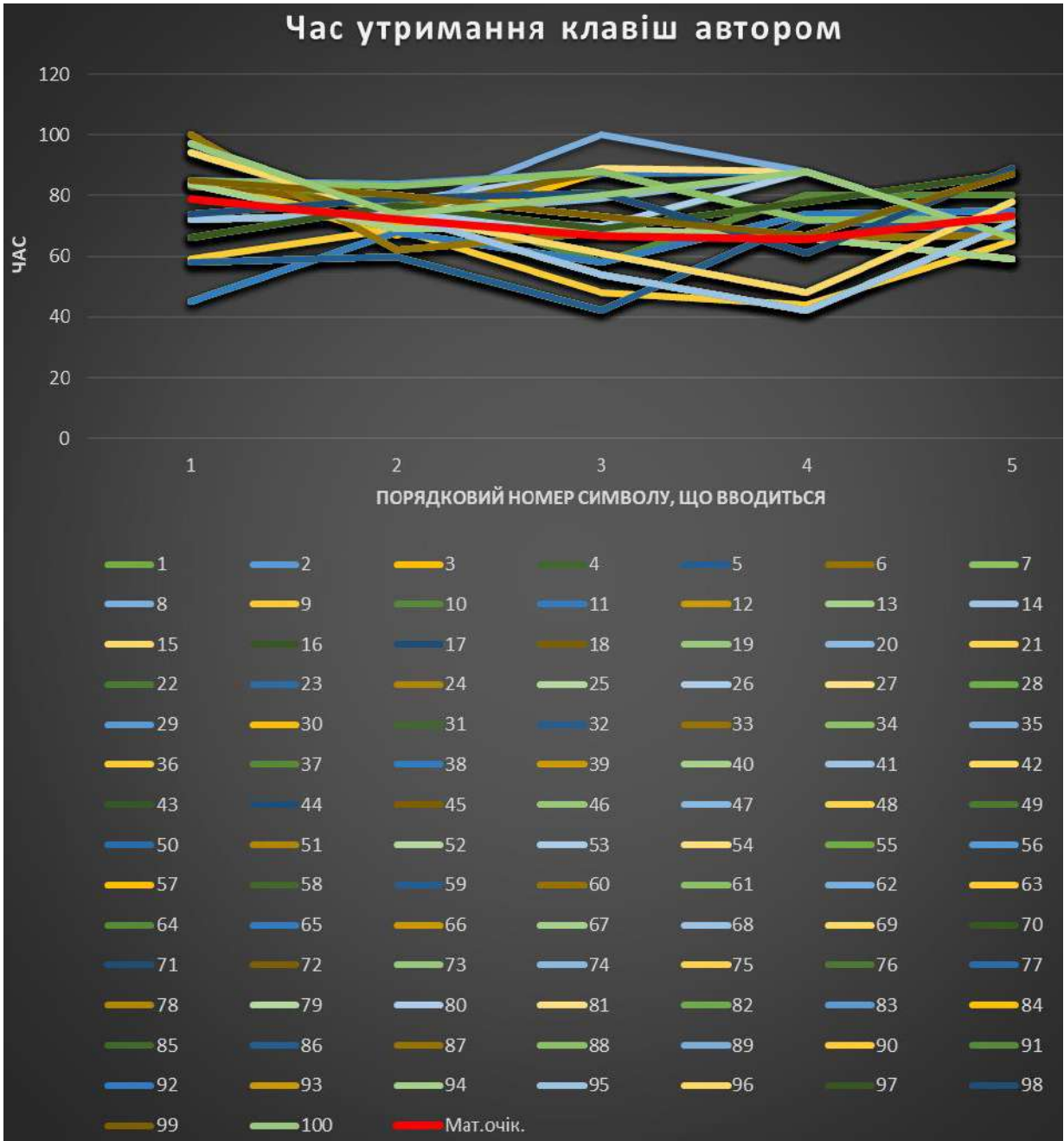


Рисунок 4.2 – Діаграма часу утримання клавiш автором

На вертикальній осі рисунку 4.2 вказується час утримування клавiш при натиску автором в мілісекундах, по горизонтальній – порядковий номер символу, що вводиться.

Проводився розрахунок середнього інтервалів часу між натисканням клавiш та часу утримування клавiш при натиску (M_x), вибірових дисперсій (D_x) та середньоквадратичних відхилень (σ), отриманні результати наведені в таблиці 4.1 та 4.2 відповідно.

Таблиця 4.1 – Середнє, вибіркова дисперсія та середньоквадратичні відхилення інтервалів часу між натисканням клавіш автором

Інтервали	1	2	3	4
Mx	97,82	160,10	61,98	96,74
Dx	1754,11	544,66	1482,12	376,28
Sigma	41,88	23,34	38,50	19,40

Таблиця 4.2 – Середнє, вибіркова дисперсія та середньоквадратичні відхилення часу утримування клавіш автором при натиску

№ символу	1	2	3	4	5
Mx	78,99	72,41	66,53	65,46	73,43
Dx	205,65	37,13	156,07	172,01	100,69
Sigma	14,34	6,09	12,49	13,12	10,03

Для аналізу чи пов'язані між собою параметри введення одного і того ж логіна, але в різний час знаходились кореляційні (табл. 4.3 та 4.4) та коваріаційні матриці (табл. 4.5 та 4.6) отриманих параметрів клавіатурного почерку для двох масивів, що зазначались вище.

Таблиця 4.3 – Кореляційна матриця інтервалів часу між натисканням клавіш автором

1	2	3	4
1,00	0,32	-0,04	-0,19
0,32	1,00	0,02	-0,02
-0,04	0,02	1,00	0,09
-0,19	-0,02	0,09	1,00

Таблиця 4.4 – Кореляційна матриця інтервалів часу утримування клавіш автором при натиску

1	2	3	4	5
1,00	0,11	0,49	0,02	-0,20
0,11	1,00	0,49	0,03	0,67
0,49	0,49	1,00	0,41	0,18
0,02	0,03	0,41	1,00	0,06
-0,20	0,67	0,18	0,06	1,00

Таблиця 4.5 – Коваріаційна матриця інтервалів часу між натисканням клавіш автором

1	2	3	4
1736,60	310,70	-63,20	-148,90
310,70	539,20	15,90	-10,50
-63,20	15,90	1467,30	63,50
-148,90	-10,50	63,50	372,50

Таблиця 4.6 – Коваріаційна матриця інтервалів часу утримування клавіш автором при натиску

1	2	3	4	5
203,60	9,40	87,50	3,50	-28,10
9,40	36,80	37,10	2,20	40,30
87,50	37,10	154,50	67	22,10
3,50	2,20	67	170,30	7,70
-28,10	40,30	22,10	7,70	99,70

Клавіатурний почерк характеризує кореляція отриманих даних. Від коефіцієнта кореляції залежать кути нахилу відрізків ламаної лінії діаграми для інтервалів часу між натисканням клавіш автором та часу утримування клавіш автором при натиску.

Оскільки при дослідженні були побудовані криві вводу для кожної вибірки (100 разів введення логіна), аналізуючи їх можна зробити висновок, що вид ламаної показує характер вводу логіна.

На другому етапі аналіз проводився на групі учасників, котрі вводили запропонований автором логін, тобто логін, який вони до цього не знали, для перевірки чи впливає на час вводу попередні навички вводу паролльної фрази.

Для кожного учасника були знайдені вибіркові середні всіх інтервалів та часу утримання кожної клавіші (дані наведені в Додатку В пункті В.3 та В.4).

Побудовані діаграми середніх значень для інтервалів між натискання клавіш (рис. 4.3) та утримання клавіш (рис. 4.4). На рисунку середні значення автора (еталонні) виділені червоним кольором.

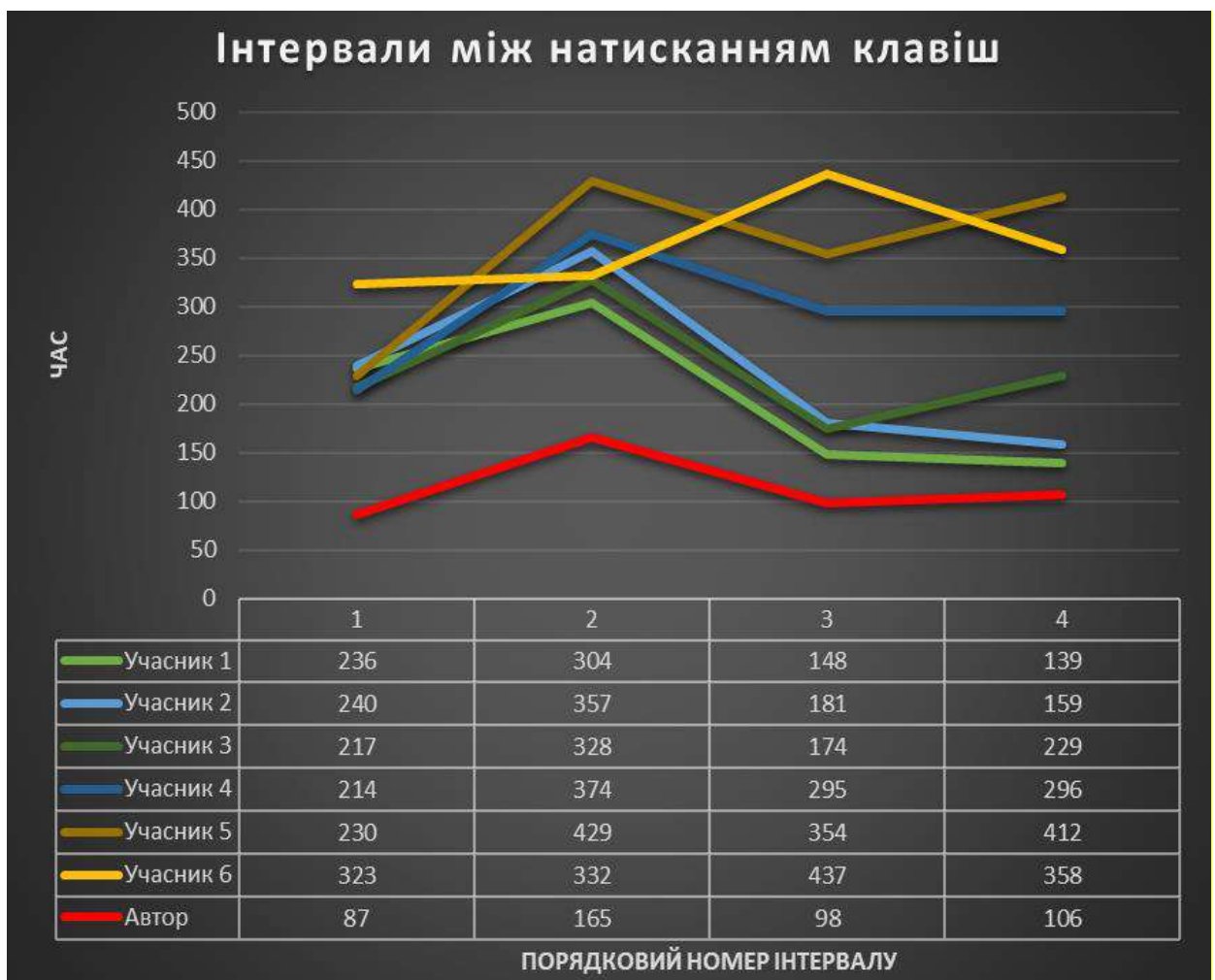


Рисунок 4.3 – Середні значення часу інтервалів між натисканням клавіш

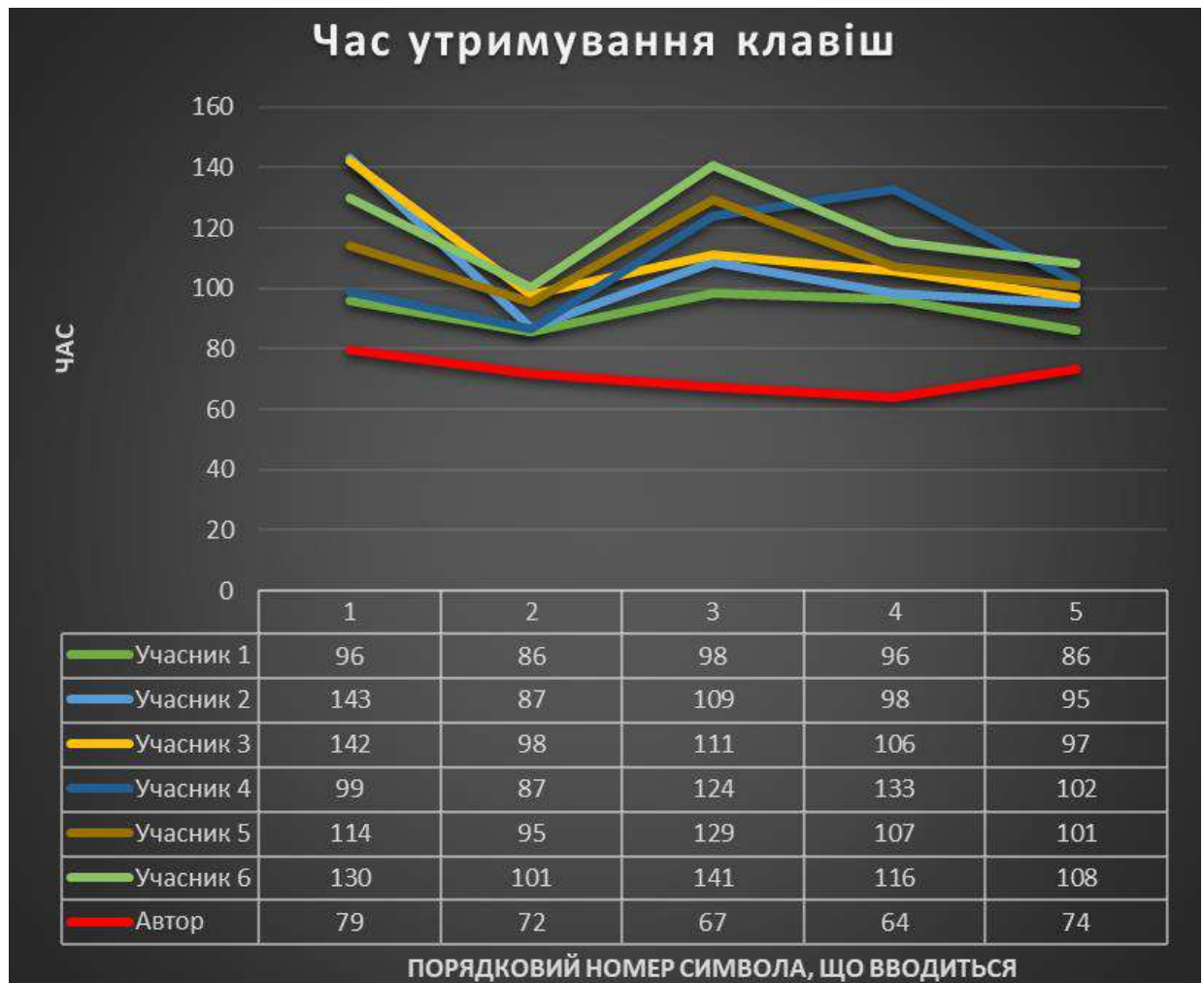


Рисунок 4.4 – Середні значення часу утримування клавiш

Як видно з рисунків 4.3 та 4.4 отримані дані під час тестування модулю різними учасниками не потрапляють в довірчі інтервали еталонних даних (табл. 4.7 та 4.8), що дає змогу зробити висновок про те, що клавіатурний почерк кожного користувача є унікальним та непідробним.

Таблиця 4.7 – Довірчі інтервали часу між натисканням клавiш автором

Інтервал	1	2	3	4
Нижня межа	59,09	153,83	41,89	90,58
Верхня межа	114,91	176,72	153,38	121,96

Таблиця 4.8 – Довірчі інтервали часу утримування клавіш автором при натиску

Символ	1	2	3	4	5
Нижня межа	70,7	67,81	58,31	55,45	67,47
Верхня межа	88,21	75,46	76,23	72,55	79,6

Порівнюючи вибірккові дисперсії, приходимо до висновку, що дисперсії практично усіх часових інтервалів для еталонних даних менше ніж дисперсії для «чужих» (табл. 4.9 та 4.10). Це пояснюється тим, що логін, який вводили учасники, був їм не знайомий, для них це довільний набір символів.

Таблиця 4.9 – Вибіркові дисперсії часу між натисканням клавіш автором

	Інтервал			
Учасник	1	2	3	4
Автор	2204,4	370,62	8789,85	696,42
1	91456,4	35027,27	25443,09	4126,67
2	77894,62	52440,22	48118,09	3111,69
3	48074,07	38904,2	37634,02	15925,22
4	37526,82	52988,27	35161,05	17704,85
5	35197,25	24078,16	18923,6	23136,27
6	38592,47	27340,07	11663,27	20996,76

Таблиця 4.10 – Вибіркові дисперсії часу утримування клавіш автором при натиску

	Інтервал				
Учасник	1	2	3	4	5
Автор	216,87	41,45	227,22	206,6	104,47
1	462,96	742,42	1067,42	690,96	517,69

Продовження таблиці 4.10

2	28992,89	597,2	857,82	717,16	476,09
3	23366,02	797,49	2563,85	944,25	1220,2
4	1410,49	1091,76	1673,69	3618,25	918,82
5	1840,89	137,45	1400,05	1946,6	777,76
6	2274,65	56,47	1571,25	1360,07	403,45

Для описаної вище вибірки вдалось відрізнити еталонний набір параметрів від «чужих».

4.2 Тестування працездатності програмного модуля

На етапі налаштування автор ввів 100 разів свій логін, як зазначалось в минулому пункті. В результаті чого був побудований еталонний вектор біометричних параметрів.

Після цього етапу модель готова проводити тестування нових введених даних.

Для тестування моделі спочатку автор вводив 250 разів свій логін, приклад одного з введів наведено на рисунку 4.5.

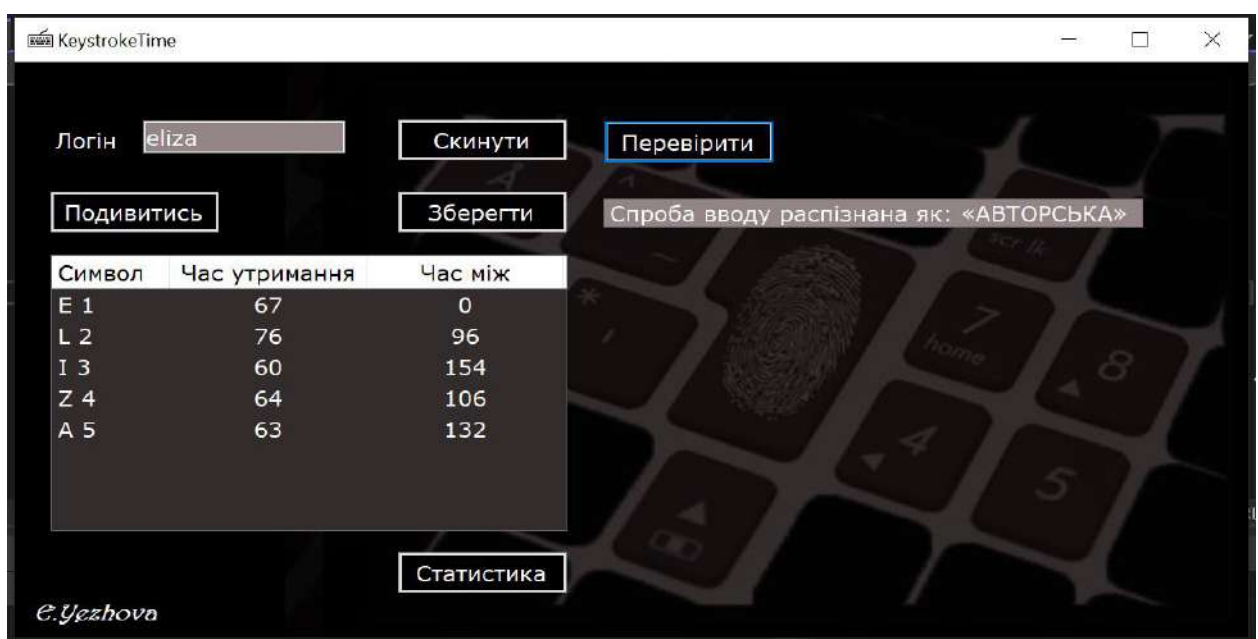


Рисунок 4.5 – Приклад вводу логіна «eliza», що розпізнаний авторським

Результат – помилка першого роду, при якій автор не зміг пройти перевірку) не більше 6%.

Далі відбулось тестування 150 неавторських введів, приклад такого вводу наведено на рисунку 4.6.

Порівняння часу утримання клавіш та часу між введенням символів логіна «eliza» автором (середніх значень, дані наведені в Додатку В пункті В.3 та В.4) та учасником наведено в таблиці 4.11 та 4.12 відповідно.

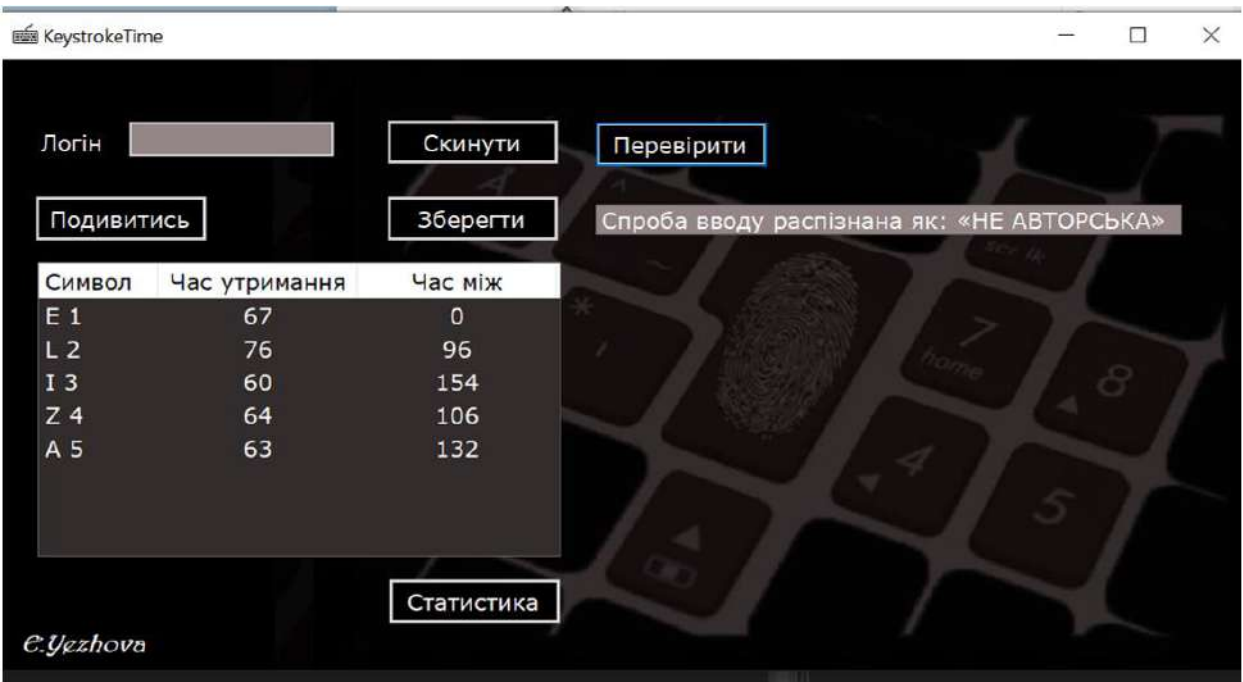


Рисунок 4.6 – Приклад вводу логіна «eliza» іншим учасником, що розпізнаний не авторським

Таблиця 4.11 – Час утримування клавіш автором та іншим учасником під час вводу логіна «eliza»

	Номер символу				
	1	2	3	4	5
Автор	79,45	71,64	67,27	64	73,55
Учасник	7751	9537	12508	15723	17855

Таблиця 4.12 – Інтервали між натискання клавiш автором та iншим учасником пiд час вводу логiна «eliza»

	Номер символу			
	1	2	3	4
Автор	87	165,27	97,64	106,27
Учасник	1784	2980	3231	2119

Результат – помилка другого роду (не автор пройшов перевiрку) склала 0%.

Очевидно, що при збiльшеннi кiлькостi тестових ввводiв помилка другого роду може виникнути, але на основi отриманих результатiв можна сказати, що функцiя досить гарно вiдрiзняє «своїх» вiд «чужих».

4.3 Висновки за роздiлом

Таким чином, за результатами проведеного аналізу можна зробити висновок, що iнтервали мiж натисканням клавiш бiльш точно показують iндивiдуальнi властивостi клавiатурного почерку користувача при введеннi логiна.

Для пiдвищення ефективностi аутентифiкацiї слiд у вектор бiометричних параметрiв внести iнтервали мiж натисканням клавiш, нормоване математичне сподiвання тривалостi цих iнтервалiв.

Також необхідно внести в вектор параметри, що показують динамiку змiн часових iнтервалiв в процесi набору.

ВИСНОВКИ

Проводячи аналіз виконаної роботи, можна дійти висновку, що вдалось розв'язати усі поставлені на початку виконання роботи задачі.

Під час даної роботи було досліджено аутентифікації користувача по клавіатурному почерку під час введення парольної фрази.

На дослідах було підтверджено, що введення коротких та добре відомих фраз відбувається на підсвідомому рівні за допомогою м'язової пам'яті.

На основі проведеного аналізу та розробки математичної функції розподілу областей «свої» та «чужі» був розроблений модуль для фільтрації авторського вводу від неавторського.

В результаті тестування якого було виявлено, що він досить добре розрізняє спроби вводу.

Використовуючи даний модуль можна створити гібридну систему керування доступом, яка буде поєднувати в собі парольні та біометричні методи аутентифікації. Таким чином, довершена система керування буде забезпечувати посилену, в порівнянні з класичної парольною аутентифікацією, процедуру аутентифікації.

Перевагою такої системи є те, що для її використання не потрібно буде закуповувати якесь додаткове обладнання, а також не потрібно значною мірою модифікувати процес аутентифікації.

Недоліком є те, що користувач не може бути під дією психотропних препаратів, алкогольного сп'яніння чи сильно хворий.

В процесі даної роботи отримано багато теоретичних та практичних знань, на практиці застосовані раніше набуті знання та вміння, отримані при навчанні на бакалавраті спеціальності 125 «Кібербезпека».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Єжова Є.О., Overview and problems of biometric authentication methods – Трускавець: Актуальні питання та перспективи проведення наукових досліджень: матеріали III Міжнародної студентської наукової конференції (Т. 3), м. Трускавець, 10 грудня, 2021 рік / ГО «Молодіжна наукова ліга». – Вінниця: ГО «Європейська наукова платформа», 2021. – 136 с. – 53–55 с.
2. Казарін М. М. Розробка та дослідження методів потайного клавіатурного моніторингу : дис. канд. техн. наук : 05.13.19 / Казарін Максим Миколайович – Таганрог, 2006. – 181 с.
3. Pato J. N. Biometric recognition: Challenges and opportunities. Washington, D.C : National Academies Press, 2010. 165 p.
4. Pankanti S., Prabhakar S., Jain A. K. On the individuality fingerprints. 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA. URL: <https://doi.org/10.1109/cvpr.2001.990563> (date of access: 04.04.2022).
5. Davida G. I., Frankel Y., Matt B. J. On enabling secure applications through off-line biometric identification. Proceedings. 1998 IEEE Symposium on Security and Privacy (Cat. No.98CB36186), Oakland, CA, USA. URL: <https://doi.org/10.1109/secpri.1998.674831> (date of access: 04.04.2022).
6. Lin C.-H., Lai Y.-Y. A flexible biometrics remote user authentication scheme. Computer Standards & Interfaces. 2004. Vol. 27, no. 1. P. 19–23. URL: <https://doi.org/10.1016/j.csi.2004.03.003> (date of access: 04.04.2022).
7. Thieme M., Nanavati S., Nanavati R. Biometrics: Identity Verification in a Networked World. Wiley, 2002. 320 p.
8. Eliza Du Y. Review of iris recognition: cameras, systems, and their applications. Sensor Review. 2006. Vol. 26, no. 1. P. 66–69. URL: <https://doi.org/10.1108/02602280610640706> (date of access: 04.04.2022).

9. Optics and photonics – Spectral bands. ISO 20473: 2007. ISO Focus № 5, 2007, 72 p. (ISO standard). URL: <https://www.iso.org/standard/39482.html> (date of access: 04.04.2022).
10. Introduction / R. M. Bolle et al. Guide to Biometrics. New York, NY, 2004. P. 3–15. URL: https://doi.org/10.1007/978-1-4757-4036-3_1 (date of access: 04.04.2022).
11. Personal authentication using digital retinal images / C. Mariño et al. Pattern Analysis and Applications. 2006. Vol. 9, no. 1. P. 21–33. URL: <https://doi.org/10.1007/s10044-005-0022-6> (date of access: 04.04.2022).
12. Retinal Vein detection using Residual Block Incorporated U-Net Architecture and Fuzzy Inference System [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/figure/Retinal-Vessel-segmentation-using-ResNet-Unet-a-Actual-image-b-Ground-truth-image-c_fig1_329164442. – Заголовок з екрану (дата звернення 04.04.2022).
13. Identity verification using shape and geometry of human hands / S. Sharma et al. Expert Systems with Applications. 2015. Vol. 42, no. 2. P. 821–832. URL: <https://doi.org/10.1016/j.eswa.2014.08.052> (date of access: 04.04.2022).
14. Jain A. K., Ross A., Prabhakar S. An Introduction to Biometric Recognition. IEEE Transactions on Circuits and Systems for Video Technology. 2004. Vol. 14, no. 1. P. 4–20. URL: <https://doi.org/10.1109/tcsvt.2003.818349> (date of access: 04.04.2022).
15. Kodituwakku S. R., Biometric Authentication: A Review, International Journal of Trend in Research and Development, Vol. 2(4), pp. 113–123, 2015.
16. NATO Advanced Research Workshop on Identity, Security and Democracy: The wider social and ethical implications of automated systems for human identification. Amsterdam : Ios Press, 2009. 117 p.
17. Face Recognition. [Електронний ресурс]. – Режим доступу: <https://www.eff.org/tr/pages/face-recognition>. – Заголовок з екрану (дата звернення 04.04.2022).

18. Jain A., Hong L., Pankanti S. Biometric identification. Communications of the ACM. 2000. Vol. 43, no. 2. P. 90–98. URL: <https://doi.org/10.1145/328236.328110> (date of access: 04.04.2022).
19. Singh M., Arora A. S. Computer Aided Face Liveness Detection with Facial Thermography. Wireless Personal Communications. 2019. Vol. 111, no. 4. P. 2465–2476. URL: <https://doi.org/10.1007/s11277-019-06996-6> (date of access: 04.04.2022).
20. Dynamic methods of biometric identity authentication. URL: http://re.mipt.ru/infsec/2006/essay/2006_Dynamic_biometric_authentication__C_herkeзов.pdf (date of access: 04.04.2022).
21. Chen W., Li H. Adversarial Attacks on Voice Recognition Based on Hyper Dimensional Computing. Journal of Signal Processing Systems. 2021. Vol. 93, no. 7. P. 709–718. URL: <https://doi.org/10.1007/s11265-020-01634-y> (date of access: 04.04.2022).
22. Inthavisas K., Lopresti D. Secure speech biometric templates for user authentication. IET Biometrics. 2012. Vol. 1, no. 1. P. 46. URL: <https://doi.org/10.1049/iet-bmt.2011.0008> (date of access: 04.04.2022).
23. Morgen B. Voice biometrics for customer authentication. Biometric Technology Today. 2012. Vol. 2012, no. 2. P. 8–11. URL: [https://doi.org/10.1016/s0969-4765\(12\)70054-1](https://doi.org/10.1016/s0969-4765(12)70054-1) (date of access: 04.04.2022).
24. DWT Domain On-Line Signature Verification / I. Nakanishi et al. Biometrics. 2011. URL: <https://doi.org/10.5772/17486> (date of access: 04.04.2022).
25. Nalwa V. S. Automatic on-line signature verification. Proceedings of the IEEE. 1997. Vol. 85, no. 2. P. 215–239. URL: <https://doi.org/10.1109/5.554220> (date of access: 04.04.2022).
26. SIGNificant Biometric Server. – 2021. – URL: <https://www.microelectronic.com.au/product/real-time-signature-verification/>. (date of access: 04.04.2022).
27. Monroe F., Rubin A. Authentication via keystroke dynamics. the 4th ACM conference, Zurich, Switzerland, 1–4 April 1997. New York, New York,

USA, 1997. URL: <https://doi.org/10.1145/266420.266434> (date of access: 04.04.2022).

28. Conrad E., Feldman J., Misenar S. Eleventh Hour CISSP: Study Guide. Syngress, 2016. 238 p.

29. Anh T. Dang. Biometric Authentication Methods / Anh T. Dang. – 2020. – URL: <https://towardsdatascience.com/biometric-authentication-methods-61c96666883a> (date of access: 04.04.2022).

30. Абашин В. Г. Автоматизація процесу визначення психофізіологічного стану оператора автоматизованого робочого місця в АСУТП : дис. канд. техн. наук : 05.13.06 / Абашин Валерій Геннадійович – Орел, 2008. – 20 с.

31. Шаріпов Р. Р. Розробка полігаусового алгоритму аутентифікації користувачів у телекомунікаційних системах та мережах за клавіатурним почерком : дис. канд. техн. наук : 05.12.13 / Шаріпов Ріфат Рашатович – Казань, 2006. – 135 с.

32. Технология поселенной аутентификации користувачів інформаційних процессов / І. А.Ходашинский, М. В. Савчук, И. В. Горбунов, Р. В. Мещеряков. // Доповіді ТУСУР. – 2011. – №2. – С. 236–248.

33. Брюхоміцький Ю. А. Гістограмний метод розпізнавання клавіатурного почерку [Електронний ресурс] / Юрій Анатолійович Брюхоміцький // Вісті ЮФУ. – 2010. – Режим доступу: <https://cyberleninka.ru/article/n/gistogrammnyy-metod-raspoznavaniya-klaviaturnogo-rocherka> (дата звернення: 26.04.2022).

34. Стрельцов О.В., Войтов В.М. Дослідження методів аутентифікації користувача за клавіатурним почерком. Наукові праці: Науково-методичний журнал. Вип. 275. Т. 287. Миколаїв : Вид-во ЧДУ ім. Петра Могили, 2016. С. 116-124.

35. Kamal M. Biometrics. What and how. [Електронний ресурс] / Moustafa Kamal // Delta Academy for Science – Режим доступу: http://www.infosecwriters.com/Papers/MKamal_Biometrics.pdf. – Заголовок з

екрану (дата звернення: 26.04.2022).

36. Методи оброблення та аналізу експериментальних даних. Головна - Таврійський державний агротехнологічний університет імені Дмитра Моторного. URL: http://www.tsatu.edu.ua/mvz/wp-content/uploads/sites/5/lekciya_10.pdf (дата звернення: 16.05.2022).

37. ННІ «Інститут геології» КНУ імені Тараса Шевченка. URL: http://www.geol.univ.kiev.ua/lib/zhukov_n_n/МС_3-4.pdf (дата звернення: 16.05.2022).

38. Слюсарчук П. В. Теорія ймовірностей та математична статистика : підруч. для студ. вищ. навч. закл. Ужгород : Карпати, 2005. 184 с.

39. Одержання біометричного еталону клавіатурного почерку. URL: <https://www.docme.su/doc/229042/laboratorna-robota-1-3> (дата звернення: 16.05.2022).

40. Власенко К. В. t – критерій Стьюдента. URL: <http://vmdbi.net.ua/t-kriteriy-styudenta/> (дата звернення: 16.05.2022).

41. Таблиця значень критерія Стьюдента (t-критерій). URL: <https://repository.tdmu.edu.ua/handle/1/10868> (дата звернення: 16.05.2022).

42. McGrath M. C# Programming in Easy Steps / Mike McGrath. – Warwickshire: In Easy Steps Limited, 2020. – 192 с. – (In Easy Steps). – (ISBN-10: 1840789069).

43. Sufyan B. Mastering C#: A Beginner's Guide (Mastering Computer Science) / Bin Uzayr Sufyan. – Бока-Ратон, Флорида: CRC Pr I Llc, 2022. – 308 с. – (1st Edition) – (ISBN-10: 1032103221).

44. Sells C. Windows Forms Programming in C# / Chris Sells. – USA: AddisonWesley Professional, 2003. – 734 с. – (Illustrated edition). – (ISBN-10: 0321116208).

45. Kenton W. First In, First Out (FIFO). Investopedia. URL: <https://www.investopedia.com/terms/f/fifo.asp> (date of access: 22.05.2022).

46. Stopwatch class (system.diagnostics). Developer tools, technical documentation and coding examples | Microsoft Docs. URL:

<https://docs.microsoft.com/en-us/dotnet/api/system.diagnostics.stopwatch?view=net-6.0> (date of access: 22.05.2022).

Додаток А

Таблиця А.1 – Зауваження нормоконтролера

[illegible]

Додаток Б

Таблиця значень критерію Стьюдента (Т-критерію)

У таблиці Б.1 критичних значень t-критерію Стьюдента є теоретичні значення критерію.

Таблиця Б.1 – Критичні значення t-критерію Стьюдента

L	P		
	p=0,05	p=0,01	p=0,001
10	2,228	3,169	4,587
50	2,009	2,678	3,496
100	1,984	2,626	3,390
110	1,982	2,621	3,381
120	1,980	2,617	3,373
130	1,978	2,614	3,367
140	1,977	2,611	3,361
150	1,976	2,609	3,357
200	1,972	2,601	3,340
250	1,969	2,596	3,330
300	1,968	2,592	3,323
350	1,967	2,590	3,319

Додаток В

Дані для проведення тестування

В.1 Інтервали між натисканням клавіш автором та їх характеристики

Таблиця В.1 – Інтервали між натисканням клавіш автора

1	15	164	91	104		51	132	164	91	104
2	20	165	96	114		52	120	165	96	114
3	90	173	75	109		53	173	173	75	109
4	84	161	84	108		54	111	161	84	108
5	132	149	80	180		55	15	149	80	180
6	120	183	50	101		56	20	183	50	101
7	173	158	50	83		57	90	158	50	83
8	111	159	74	95		58	84	159	74	95
9	84	164	40	100		59	84	164	40	100
10	54	209	59	90		60	54	209	59	90
11	74	133	375	85		61	74	133	75	85
12	57	143	65	86		62	57	143	65	86
13	107	162	59	97		63	107	162	59	97
14	44	160	51	96		64	44	160	51	96
15	84	148	86	82		65	84	148	86	82
16	59	162	45	154		66	59	162	45	154
17	71	169	90	105		67	71	169	90	105
18	70	158	74	88		68	70	158	74	88
19	67	151	69	87		69	67	151	69	87
20	97	125	50	82		70	97	125	50	82
21	180	150	55	90		71	180	150	55	90
22	65	153	60	125		72	65	153	60	125
23	112	134	45	83		73	112	134	45	83
24	61	157	55	109		74	61	157	55	109
25	54	126	55	103		75	54	126	55	103
26	90	119	40	93		76	90	119	40	93
27	62	129	61	66		77	62	129	61	66
28	120	126	40	96		78	132	126	40	96
29	144	139	86	91		79	120	139	86	91
30	68	160	70	79		80	173	160	70	79
31	120	163	19	95		81	111	163	19	95
32	119	190	79	94		82	119	190	79	94
33	173	209	87	117		83	173	209	87	117
34	61	162	48	74		84	61	162	48	74
35	60	181	40	94		85	60	181	40	94
36	74	167	71	115		86	74	167	71	115
37	155	191	29	79		87	155	191	29	79

Продовження таблиці В.1

38	178	194	53	90
39	150	142	38	102
40	130	173	13	95
41	93	145	25	83
42	114	214	50	74
43	61	131	10	90
44	89	136	45	97
45	60	139	60	95
46	156	188	49	68
47	73	149	38	100
48	139	173	60	93
49	164	209	100	81
50	111	160	105	120

88	178	194	53	90
89	150	142	38	102
90	130	173	13	95
91	93	145	25	83
92	114	214	50	74
93	61	131	10	90
94	89	136	45	97
95	60	139	60	95
96	156	188	49	68
97	73	149	38	100
98	139	173	60	93
99	164	209	100	81
100	111	160	105	120

Таблиця В.2 – Характеристики інтервалів між натисканням клавіш автора

Mx	97,82	160,1	61,98	96,74
Dx	1754,11	544,66	1482,12	376,28
sigma	41,88	23,34	38,5	19,4
Кореляція:				
	1	0,321	-0,04	-0,185
	0,321	1	0,018	-0,024
	-0,04	0,018	1	0,086
	-0,185	-0,024	0,086	1
Коваріація:				
	1736,6	310,7	-63,2	-148,9
	310,7	539,2	15,9	-10,5
	-63,2	15,9	1467,3	63,5
	-148,9	-10,5	63,5	372,5
Довірчі інтервали:				
	89,56	155,5	54,39	92,91
	106,08	164,7	69,57	100,57

В.2 Час утримання клавіш автором

Таблиця В.3 – Час утримування клавіш автором

1	58	60	42	72	68		51	58	60	42	72	68
2	100	62	68	66	67		52	100	62	68	66	67
3	85	70	88	72	72		53	85	78	88	72	72
4	84	69	69	66	59		54	84	69	69	66	59

Продовження таблиця В.3

5	72	74	54	42	71		55	58	60	42	72	68
6	94	74	61	48	78		56	100	62	68	66	67
7	66	77	69	78	87		57	85	67	88	72	72
8	74	79	81	61	89		58	84	69	69	66	59
9	85	80	73	67	87		59	66	77	69	78	87
10	97	74	87	88	66		60	74	79	81	61	89
11	59	69	48	44	65		61	85	80	73	67	87
12	83	79	58	80	80		62	97	74	100	88	66
13	45	68	58	74	75		63	59	69	48	44	65
14	84	69	69	66	59		64	83	79	58	80	80
15	72	74	54	42	71		65	45	68	58	74	75
16	94	74	61	48	78		66	84	69	69	66	59
17	66	77	69	78	87		67	72	74	54	42	71
18	74	79	81	61	89		68	94	74	61	48	78
19	85	80	73	67	87		69	66	77	69	78	87
20	97	74	79	88	66		70	58	60	42	72	68
21	84	69	69	66	59		71	100	62	68	66	67
22	72	74	54	42	71		72	85	75	88	72	72
23	94	74	61	48	78		73	84	69	69	66	59
24	66	77	69	78	87		74	84	69	69	66	59
25	74	79	81	61	89		75	72	74	54	42	71
26	85	80	73	67	87		76	94	74	61	48	78
27	97	74	89	88	66		77	66	77	69	78	87
28	59	69	48	44	65		78	74	79	81	61	89
29	83	79	58	80	80		79	85	80	73	67	87
30	45	68	58	74	75		80	97	74	70	88	66
31	84	69	69	66	59		81	58	60	42	72	68
32	72	74	54	42	71		82	100	62	68	66	67
33	94	74	61	48	78		83	85	84	88	72	72
34	66	77	69	78	87		84	84	69	69	66	59
35	58	60	42	72	68		85	84	69	69	66	59
36	100	62	68	66	67		86	58	60	42	72	68
37	85	78	88	72	72		87	100	62	68	66	67
38	84	69	69	66	59		88	85	83	88	72	72
39	94	74	61	48	78		89	84	69	69	66	59
40	66	77	69	78	87		90	84	69	69	66	59
41	74	79	81	61	89		91	72	74	54	42	71
42	85	80	73	67	87		92	94	74	61	48	78
43	97	74	69	88	66		93	66	77	69	78	87
44	59	69	48	44	65		94	84	69	69	66	59
45	83	79	58	80	80		95	72	74	54	42	71
46	45	68	58	74	75		96	94	74	61	48	78
47	84	69	69	66	59		97	66	77	69	78	87
48	72	74	54	42	71		98	74	79	81	61	89
49	94	74	61	48	78		99	85	80	73	67	87
50	66	77	69	78	87		100	97	74	80	88	66

Таблиця В.4 – Характеристики часу утримування клавіш автором

Mx	78,99	72,41	66,53	65,46	73,43
Dx	205,65	37,13	156,07	172,01	100,69
sigma	14,34	6,09	12,49	13,12	10,03
Кореляція:					
	1	0,108	0,493	0,019	-0,197
	0,108	1	0,493	0,028	0,665
	0,493	0,493	1	0,413	0,178
	0,019	0,028	0,413	1	0,059
	-0,197	0,665	0,178	0,059	1
Коваріація:					
	203,6	9,4	87,5	3,5	-28,1
	9,4	36,8	37,1	2,2	40,3
	87,5	37,1	154,5	67	22,1
	3,5	2,2	67	170,3	7,7
	-28,1	40,3	22,1	7,7	99,7
Довірчі інтервали:					
	76,16	71,21	64,07	62,87	71,45
	81,82	73,61	68,99	68,05	75,41

В.3 Інтервали між натисканням клавіш та їх характеристики

Таблиця В.5 – Інтервали між натисканням клавіш учасника – автор

Автор					
1	58	60	42	72	68
2	100	62	68	66	67
3	85	70	88	72	72
4	84	69	69	66	59
5	72	74	54	42	71
6	94	74	61	48	78
7	66	77	69	78	87
8	74	79	81	61	89
9	85	80	73	67	87
10	97	74	87	88	66
11	59	69	48	44	65

Таблиця В.6 – Характеристики інтервалів між натисканням клавіш учасника – автор

Mx	87	165,27	97,64	106,27
Dx	2204,4	370,62	8789,85	696,42

Продовження таблиці Б.6

sigma	46,95	19,25	93,75	26,39
Кореляція:				
	1	-0,199	-0,196	0,109
	-0,199	1	-0,595	-0,208
	-0,196	-0,595	1	-0,189
	0,109	-0,208	-0,189	1
Коваріація:				
	2004	-163,4	-784,3	123
	-163,4	336,9	-976,2	-96
	-784,3	-976,2	7990,8	-424,3
	123	-96	-424,3	633,1
Довірчі інтервали:				
	59,09	153,83	41,89	90,58
	114,91	176,72	153,38	121,96

Таблиця В.7 – Інтервали між натисканням клавіш учасника 1

Учасник 1				
1	122	140	256	120
2	60	193	97	114
3	60	653	77	114
4	1053	696	597	329
5	77	206	112	130
6	119	234	99	134
7	214	251	53	120
8	535	198	74	114
9	94	236	49	111
10	154	260	74	98
11	108	282	141	140

Таблиця В.8 – Характеристики інтервалів між натисканням клавіш учасника 1

Mx	236	304,45	148,09	138,55
Dx	91456,4	35027,27	25443,09	4126,67
sigma	302,42	187,16	159,51	64,24
Кореляція:				
	1	0,541	0,805	0,867
	0,541	1	0,567	0,667
	0,805	0,567	1	0,94
	0,867	0,667	0,94	1

Продовження таблиці В.8

Коваріація:				
	83142,2	27854,2	35312,4	15305,9
	27854,2	31843	15377,9	7293
	35312,4	15377,9	23130,1	8757,4
	15305,9	7293	8757,4	3751,5
Довірчі інтервали:				
	56,2	193,18	53,25	100,35
	415,8	415,73	242,93	176,74

Таблиця В.9 – Інтервали між натисканням клавіш учасника 2

Учасник 2				
1	134	145	355	125
2	70	124	235	123
3	76	678	75	143
4	999	578	780	300
5	154	234	54	145
6	100	533	100	234
7	214	123	53	134
8	500	145	74	134
9	89	236	65	143
10	180	678	78	134
11	121	456	123	135

Таблиця В.10 – Характеристики інтервалів між натисканням клавіш учасника 2

Mx	239,73	357,27	181,09	159,09
Dx	77894,62	52440,22	48118,09	3111,69
sigma	279,1	229	219,36	55,78
Кореляція:				
	1	0,159	0,772	0,713
	0,159	1	0,156	0,464
	0,772	0,156	1	0,7
	0,713	0,464	0,7	1
Коваріація:				
	70813,3	9210	42979	10090,3
	9210	47672,9	7101,3	5384,3
	42979	7101,3	43743,7	7790,5
	10090,3	5384,3	7790,5	2828,8

Продовження таблиці В.10

Довірчі інтервали:				
	73,79	221,12	50,67	125,93
	405,66	493,42	311,51	192,26

Таблиця В.11 – Інтервали між натисканням клавіші учасника 3

Учасник 3				
1	145	134	345	134
2	89	100	200	143
3	68	400	80	154
4	800	600	700	321
5	165	250	60	154
6	121	543	121	254
7	200	120	45	234
8	450	165	78	124
9	90	240	78	332
10	134	600	89	134
11	130	456	121	532

Таблиця В.12 – Характеристики інтервалів між натисканням клавіші учасника 3

Mx	217,45	328	174,27	228,73
Dx	48074,07	38904,2	37634,02	15925,22
sigma	219,26	197,24	193,99	126,2
Кореляція:				
	1	0,272	0,752	0,101
	0,272	1	0,302	0,357
	0,752	0,302	1	0,146
	0,101	0,357	0,146	1
Коваріація:				
	43703,7	10676,1	29065,6	2550,3
	10676,1	35367,5	10520,3	8076,5
	29065,6	10520,3	34212,7	3250
	2550,3	8076,5	3250	14477,5
Довірчі інтервали:				
	87,09	210,73	58,93	153,7
	347,81	445,27	289,61	303,76

Таблиця В.13 – Інтервали між натисканням клавіш учасника 4

Учасник 4				
1	134	154	234	345
2	98	130	543	545
3	78	450	334	111
4	700	654	354	321
5	145	255	324	321
6	132	567	365	322
7	222	145	34	443
8	467	175	543	124
9	95	245	54	332
10	145	666	32	154
11	135	678	432	234

Таблиця В.14 – Характеристики інтервалів між натисканням клавіш учасника 4

Mx	213,73	374,45	295,36	295,64
Dx	37526,82	52988,27	35161,05	17704,85
Sigma	193,72	230,19	187,51	133,06
Кореляція:				
	1	0,186	0,243	-0,121
	0,186	1	-0,026	-0,46
	0,243	-0,026	1	-0,021
	-0,121	-0,46	-0,021	1
Коваріація:				
	34115,3	7549,3	8036,5	-2843,6
	7549,3	48171,2	-1017,5	-12812,9
	8036,5	-1017,5	31964,6	-479,8
	-2843,6	-12812,9	-479,8	16095,3
Довірчі інтервали:				
	98,55	237,59	183,88	216,53
	328,9	511,32	406,85	374,75

Таблиця В.15 – Інтервали між натисканням клавіш учасника 5

Учасник 5				
1	341	567	435	453
2	98	433	342	433
3	54	450	43	543
4	566	654	234	434
5	456	255	233	333
6	213	342	433	322

Продовження таблиці В.15

7	43	453	532	443
8	43	234	433	124
9	432	234	433	564
10	145	666	433	654
11	135	433	343	234

Таблиця В.16 – Характеристики інтервалів між натисканням клавіш учасника 5

Мх	229,64	429,18	354	412,45
Дх	35197,25	24078,16	18923,6	23136,27
sigma	187,61	155,17	137,56	152,11
Кореляція:				
	1	0,072	-0,161	0,157
	0,072	1	-0,075	0,504
	-0,161	-0,075	1	-0,105
	0,157	0,504	-0,105	1
Коваріація:				
	31997,5	1901,1	-3769,9	4081,4
	1901,1	21889,2	-1461,7	10811,3
	-3769,9	-1461,7	17203,3	-2001,2
	4081,4	10811,3	-2001,2	21033
Довірчі інтервали:				
	118,09	336,92	272,21	322,02
	341,18	521,44	435,79	502,89

Таблиця В.17 – Інтервали між натисканням клавіш учасника 6

Учасник 6				
1	345	443	543	123
2	434	43	534	322
3	54	433	564	322
4	543	564	544	543
5	433	433	454	322
6	213	44	431	321
7	43	333	323	322
8	54	234	233	456
9	453	433	322	321
10	543	344	432	654
11	433	353	422	234

Таблиця В.18 – Характеристики інтервалів між натисканням клавіш
учасника 6

Mx	322,55	332,45	436,55	358,18
Dx	38592,47	27340,07	11663,27	20996,76
sigma	196,45	165,35	108	144,9
Кореляція:				
	1	0,25	0,349	0,252
	0,25	1	0,183	0,076
	0,349	0,183	1	-0,159
	0,252	0,076	-0,159	1
Коваріація:				
	35084,1	7380,8	6736,3	6519,7
	7380,8	24854,6	2974,5	1653,9
	6736,3	2974,5	10603	-2256,5
	6519,7	1653,9	-2256,5	19088
Довірчі інтервали:				
	205,75	234,15	372,34	272,03
	439,34	430,76	500,75	444,33

В.4 Час утримування клавіш учасників та його характеристики

Таблиця В.19 – Час утримування клавіш учасника – автор

Автор					
1	58	60	42	72	68
2	100	62	68	66	67
3	85	70	88	72	72
4	84	69	69	66	59
5	72	74	54	42	71
6	94	74	61	48	78
7	66	77	69	78	87
8	74	79	81	61	89
9	85	80	73	67	87
10	97	74	87	88	66
11	59	69	48	44	65

Таблиця В.20 – Характеристики часу утримування клавіш учасником –
автор

Mx	79,45	71,64	67,27	64	73,55
Dx	216,87	41,45	227,22	206,6	104,47

Продовження таблиці В.20

Sigma	14,73	6,44	15,07	14,37	10,22
Кореляція:					
	1	0,081	0,613	0,25	-0,103
	0,081	1	0,472	-0,036	0,724
	0,613	0,472	1	0,563	0,268
	0,25	-0,036	0,563	1	0,039
	-0,103	0,724	0,268	0,039	1
Коваріація:					
	197,2	7	123,8	48,1	-14,1
	7	37,7	41,6	-3	43,3
	123,8	41,6	206,6	110,9	37,5
	48,1	-3	110,9	187,8	5,3
	-14,1	43,3	37,5	5,3	95
Довірчі інтервали:					
	70,7	67,81	58,31	55,45	67,47
	88,21	75,46	76,23	72,55	79,62

Таблиця В.21 – Час утримування клавіш учасника 1

Учасник 1					
1	86	85	64	101	73
2	76	66	58	67	75
3	115	47	66	63	64
4	141	40	143	140	128
5	117	122	142	142	61
6	80	85	92	87	108
7	105	110	123	82	76
8	91	119	88	108	98
9	93	78	119	101	116
10	70	105	122	92	81
11	80	86	64	75	67

Таблиця В.22 – Характеристики часу утримування клавіш учасником 1

Mx	95,82	85,73	98,27	96,18	86,09
Dx	462,96	742,42	1067,42	690,96	517,69
sigma	21,52	27,25	32,67	26,29	22,75
Кореляція:					
	1	-0,349	0,528	0,572	0,233
	-0,349	1	0,243	0,202	-0,313
	0,528	0,243	1	0,74	0,408
	0,572	0,202	0,74	1	0,39

Продовження таблиці В.22

	0,233	-0,313	0,408	0,39	1
Коваріація:					
	420,9	-186,2	337,2	293,9	103,6
	-186,2	674,9	196,4	131,6	-176,5
	337,2	196,4	970,4	577,9	275,4
	293,9	131,6	577,9	628,1	212,1
	103,6	-176,5	275,4	212,1	470,6
Довірчі інтервали:					
	83,03	69,53	78,85	80,55	72,56
	108,61	101,93	117,7	111,81	99,62

Таблиця В.23 – Час утримування клавіші учасника 2

Учасник 2					
1	89	85	78	121	123
2	65	60	98	68	65
3	654	65	77	65	78
4	122	45	140	144	132
5	117	120	155	141	87
6	87	87	100	88	98
7	100	111	132	87	79
8	90	120	78	100	100
9	95	79	123	90	121
10	75	100	136	98	85
11	80	85	79	78	78

Таблиця В.24 – Характеристики часу утримування клавіш учасником 2

Mx	143,09	87	108,73	98,18	95,09
Dx	28992,89	597,2	857,82	717,16	476,09
sigma	170,27	24,44	29,29	26,78	21,82
Кореляція:					
	1	-0,292	-0,303	-0,335	-0,206
	-0,292	1	0,155	0,135	-0,227
	-0,303	0,155	1	0,553	0,131
	-0,335	0,135	0,553	1	0,642
	-0,206	-0,227	0,131	0,642	1
Коваріація:					
	26357,2	-1105,4	-1375,2	-1388,9	-697,3
	-1105,4	542,9	100,6	80,3	-110,3
	-1375,2	100,6	779,8	394,6	76,3

Продовження таблиці В.24

	-1388,9	80,3	394,6	652	341
	-697,3	-110,3	76,3	341	432,8
Довірчі інтервали:					
	41,86	72,47	91,31	82,26	82,12
	244,33	101,53	126,14	114,1	108,06

Таблиця В.25 – Час утримування клавіш учасника 3

Учасник 3					
1	98	87	45	132	132
2	67	65	232	69	67
3	600	56	77	76	98
4	132	70	78	167	144
5	100	136	140	132	88
6	89	98	89	87	99
7	120	120	121	89	23
8	98	132	89	132	132
9	97	88	132	98	121
10	79	130	143	100	85
11	85	95	79	80	78

Таблиця В.26 – Характеристики часу утримування клавіш учасником 3

Mx	142,27	97,91	111,36	105,64	97
Dx	23366,02	797,49	2563,85	944,25	1220,2
sigma	152,86	28,24	50,63	30,73	34,93
Кореляція:					
	1	-0,486	-0,282	-0,242	0,034
	-0,486	1	0,04	0,225	-0,213
	-0,282	0,04	1	-0,404	-0,487
	-0,242	0,225	-0,404	1	0,664
	0,034	-0,213	-0,487	0,664	1
Коваріація:					
	21241,8	-1906,2	-1986,8	-1032,1	165,2
	-1906,2	725	51,8	177,3	-190,9
	-1986,8	51,8	2330,8	-571,1	-782,9
	-1032,1	177,3	-571,1	858,4	647,9
	165,2	-190,9	-782,9	647,9	1109,3
Довірчі інтервали:					
	51,39	81,12	81,26	87,37	76,23
	233,16	114,7	141,47	123,91	117,77

Таблиця В.27 – Час утримування клавіш учасника 4

Учасник 4					
1	100	87	100	132	143
2	45	60	232	69	45
3	200	67	121	234	98
4	100	72	90	167	144
5	100	130	140	234	99
6	89	78	89	53	94
7	111	100	121	90	82
8	90	143	95	145	132
9	91	23	132	123	121
10	79	100	143	123	89
11	85	95	100	89	78

Таблиця В.28 – Характеристики часу утримування клавіш учасником 4

Mx	99,09	86,82	123,91	132,64	102,27
Dx	1410,49	1091,76	1673,69	3618,25	918,82
sigma	37,56	33,04	40,91	60,15	30,31
Кореляція:					
	1	-0,053	-0,372	0,658	0,243
	-0,053	1	-0,256	0,228	0,077
	-0,372	-0,256	1	-0,137	-0,698
	0,658	0,228	-0,137	1	0,394
	0,243	0,077	-0,698	0,394	1
Коваріація:					
	1282,3	-59,3	-519,1	1351,2	251,6
	-59,3	992,5	-314,8	411,9	70,5
	-519,1	-314,8	1521,5	-305,9	-786,6
	1351,2	411,9	-305,9	3289,3	653,2
	251,6	70,5	-786,6	653,2	835,3
Довірчі інтервали:					
	76,76	67,17	99,59	96,87	84,25
	121,42	106,46	148,23	168,4	120,29

Таблиця В.29 – Час утримування клавіш учасника 5

Учасник 5					
1	100	100	123	67	134
2	89	90	232	69	57
3	234	92	131	58	98
4	121	97	98	123	143
5	132	98	134	200	77
6	99	78	98	67	98

Продовження таблиці В.29

7	121	100	123	93	82
8	98	121	99	145	132
9	94	78	134	123	121
10	80	100	140	143	89
11	87	95	111	89	78

Таблиця В.30 – Характеристики часу утримування клавіш учасником 5

Мх	114,09	95,36	129,36	107	100,82
Дх	1840,89	137,45	1400,05	1946,6	777,76
sigma	42,91	11,72	37,42	44,12	27,89
Кореляція:					
	1	-0,038	-0,091	-0,198	0,015
	-0,038	1	-0,208	0,378	0,235
	-0,091	-0,208	1	-0,171	-0,623
	-0,198	0,378	-0,171	1	0,065
	0,015	0,235	-0,623	0,065	1
Коваріація:					
	1673,5	-17,6	-133	-340,7	16,8
	-17,6	125	-83	177,6	69,9
	-133	-83	1272,8	-256,3	-590,8
	-340,7	177,6	-256,3	1769,6	73,2
	16,8	69,9	-590,8	73,2	707,1
Довірчі інтервали:					
	88,58	88,39	107,12	80,77	84,24
	139,6	102,33	151,61	133,23	117,4

Таблиця В.31 – Час утримування клавіш учасника 6

Учасник 6					
1	120	100	123	67	134
2	98	90	232	90	90
3	234	100	131	98	98
4	121	97	200	123	143
5	132	98	134	200	98
6	99	99	120	93	98
7	121	100	123	98	90
8	98	121	99	145	132
9	94	98	134	123	121
10	210	100	140	143	89
11	99	103	111	91	99

Таблиця В.32 – Характеристики часу утримування клавіш учасником 6

Mx	129,64	100,55	140,64	115,55	108,36
Dx	2274,65	56,47	1571,25	1360,07	403,45
sigma	47,69	7,51	39,64	36,88	20,09
Кореляція:					
	1	-0,088	-0,056	0,124	-0,32
	-0,088	1	-0,674	0,223	0,367
	-0,056	-0,674	1	-0,083	-0,018
	0,124	0,223	-0,083	1	-0,022
	-0,32	0,367	-0,018	-0,022	1
Коваріація:					
	2067,9	-28,8	-96,3	198,5	-278,4
	-28,8	51,3	-182,5	56,1	50,3
	-96,3	-182,5	1428,4	-110,7	-13,2
	198,5	56,1	-110,7	1236,4	-15,1
	-278,4	50,3	-13,2	-15,1	366,8
Довірчі інтервали:					
	101,28	96,08	117,07	93,62	96,42
	157,99	105,01	164,2	137,47	120,31

Додаток Г

Лістинг програмного коду

Лістинг Form1.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Text.RegularExpressions;

namespace LoginTimer1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            public struct timers // структура для часу
            {
                public double w1, w2;
            }

            timers ftime = new timers();
            double d1, d2; // час dow, час up
            public double workUp = -300; // час попереднього натиску
            клавіші

            static System.Diagnostics.Stopwatch watch = new
            System.Diagnostics.Stopwatch();
```

```

        Dictionary<string, timers> ColDic = new Dictionary<string,
timers>(); // колекція
        Queue<int> qd1 = new Queue<int>(); // черга d1

        public static string work, t_bet = "", t_pre = "", t_bet1 =
"", t_pre1 = ""; // змінна для створення ключа + порядковий номер
        int i = 0, k = 0; // порядковий номер

        private void Form1_Shown(object sender, EventArgs e)
        {
            watch.Start();
            label2.Text = "Розпізнавання";
            label3.Text = "E.Yezhova";
        }

        private void Form1_FormClosing(object sender,
FormClosingEventArgs e)
        {
            watch.Stop();
        }

        private void button1_Click(object sender, EventArgs e) // друк
результатів
        {
            foreach (KeyValuePair<string, timers> f in ColDic)
            {
                ListViewItem str = new ListViewItem(f.Key);
                str.SubItems.Add("" + f.Value.w1);
                str.SubItems.Add("" + f.Value.w2);
                listView1.Items.Add(str);
            }
        }

```

```

        private void button2_Click(object sender, EventArgs e) //
кнопка "скинути"
        {
            ColDic.Clear();
            d1 = d2 = 0;
            i = 0;
            work = "";
            workUp = -300;
            t_pre1 = t_bet1 = "";
            textBox1.Clear();
            listView1.Items.Clear();
            ColDic = new Dictionary<string, timers>();
            textBox1.Focus();
        }

        private void button3_Click(object sender, EventArgs e) //
кнопка "зберегти"
        {
            t_bet += t_bet1 + Environment.NewLine;
            File.WriteAllText("time_between.txt", t_bet);
            t_pre += t_pre1 + Environment.NewLine;
            File.WriteAllText("time_press.txt", t_pre);
            k += 1;
            label2.Text = "Кількість збережень: " + k;
            button2_Click(sender, e);
        }

        private void button4_Click(object sender, EventArgs e)
//кнопка "результати"
        {
            Read();
        }

```

```

        private void button5_Click(object sender, EventArgs e)
//кнопка "Перевірити"
        {
            Predict();
        }

        private void textBox1_KeyDown(object sender, KeyEventArgs e)
        {
            d1 = watch.ElapsedMilliseconds;
            qd1.Enqueue((int)d1);
        }

        private void textBox1_KeyUp(object sender, KeyEventArgs e)
        {
            i++; // збільшення ітератора для унікальності ключа
            d2 = watch.ElapsedMilliseconds;
            work = "" + e.KeyData + " " + i; // створення унікального
ключа
            ftime.w1 = (d2 - qd1.Dequeue()); // час (Up-Down)
            if (workUp == -300) { ftime.w2 = 0; } // якщо перший ввід,
то час між = 0;
            else { ftime.w2 = d1 - workUp; t_bet1 += ftime.w2 + "\t";
        }

            t_pre1 += ftime.w1 + "\t";
            ColDic.Add(work, ftime);
            workUp = d2; // оновлення workUp
        }

        public static double[,] Vector(double[,] between, double[,]
press)
        {
            int z = 0;
            double[] arrMint = Proverka.Mint(between); //
            double[] arrAlf = Proverka.Aritm(between, arrMint); //

```

```

double[] si = Proverka.Speed(between, press); //
double[,] diff = Proverka.Difference(between);
double[,] V = new double[between.GetLength(0),
between.GetLength(1) +
5]; //+5
for (int t = 0; t < V.GetLength(0); t++) //
{
    for (z = 0; z < V.GetLength(1) - 5; z++) //-5
    {
        V[t, z] = between[t, z];
    }
    V[t, z] = arrMint[t];
    V[t, z + 1] = arrAlf[t];
    V[t, z + 2] = si[t];
    V[t, z + 3] = diff[t, 1];
    V[t, z + 4] = diff[t, 2];
}
return V;
}

private void Read()// читання масиву з файлу та обробка
{
    string[] strArr1 = File.ReadAllLines("time_between.txt",
Encoding.Default);
    string[] strArr2 = File.ReadAllLines("time_press.txt",
Encoding.Default);
    int z = 0;
    foreach (string s in strArr2)
    {
        strArr2[z] = Regex.Replace(strArr2[z], " +", "
").Trim();
        z++;
    }
    z = 0;

```

```

foreach (string s in strArr1)
{
    strArr1[z] = Regex.Replace(strArr1[z], " +", "
").Trim();

    z++;
}
int m, k; m = strArr1.Length; k = strArr2.Length;
int n, l; l = Regex.Split(strArr2[0], "\t").Count();
n = Regex.Split(strArr1[0], "\t").Count();
int[,] res_bet = new int[m, n];
int[,] res_pres = new int[k, l];
int j = 0;
for (int i = 0; i < m; i++)
{
    foreach (string s in Regex.Split(strArr1[i], "\t"))
    {
        res_bet[i, j] = Convert.ToInt32(s);
        j++;
    }
    j = 0;
}
for (int i = 0; i < k; i++)
{
    foreach (string s in Regex.Split(strArr2[i], "\t"))
    {
        res_pres[i, j] = Convert.ToInt32(s);
        j++;
    }
    j = 0;
}
double[] arrM = ZbirStatic.M(0.0, res_bet);
double[] arrD = ZbirStatic.D(res_bet, arrM, 0.0);
double[,] arrKor = ZbirStatic.Correl(arrM, res_bet, 0);

```

```

        double[,] arrCov = ZbirStatic.Covariation(arrM, res_bet,
0);

        double[,] arrDov = ZbirStatic.Dover(arrM, arrD,
res_bet.GetLength(0));

        ZbirStatic.Print(res_bet, arrM, arrD, arrKor, arrCov,
arrDov, "time_betweenResults.txt");

        arrM = ZbirStatic.M(0.0, res_pres);
        arrD = ZbirStatic.D(res_pres, arrM, 0.0);
        arrKor = ZbirStatic.Correl(arrM, res_pres, 1);
        arrCov = ZbirStatic.Covariation(arrM, res_pres, 1);
        arrDov = ZbirStatic.Dover(arrM, arrD,
res_pres.GetLength(0));

        ZbirStatic.Print(res_pres, arrM, arrD, arrKor, arrCov,
arrDov, "time_pressResults.txt");

        arrM = ZbirStatic.M(0.0, res_pres);
    }

    public void Predict()
    {
        double[,] res_bet = ZbirStatic.Mass("time_between1.txt");
// Читання даних з файлів
        double[,] res_pres = ZbirStatic.Mass("time_press1.txt");
        double[,] V = Vector(res_bet, res_pres); // створення
еталонного вектору
        double[] arrM = ZbirStatic.M(0.0, V); // створення
мат.сподівання по вектору
        double[] arrD = ZbirStatic.D(V, arrM, 0.0); // створення
дисперсій по вектору
        double[,] arrCov = new double[V.GetLength(1),
V.GetLength(1)];
        alglib.covm(V, out arrCov);
        int info, n1 = 0, a1 = 0;
        alglib.matinvreport rep;

```

```

        alglib.rmatrixinverse(ref arrCov, out info, out rep); //
підрахунок зворотної матриці
        double[,] res_bet_test =
ZbirStatic.Mass("time_between.txt"); // читання тестових даних з
файлів
        double[,] res_pres_test =
ZbirStatic.Mass("time_press.txt");
        for (int i = 0; i < res_bet_test.GetLength(0); i++)
//запуск тестових значень
        {
            double[,] bet_test = new double[1,
res_bet_test.GetLength(1)];
            double[,] pres_test = new double[1,
res_pres_test.GetLength(1)];
            for (int j = 0; j < res_bet_test.GetLength(1); j++)
            {
                bet_test[0, j] = res_bet_test[i, j];
            }
            for (int j = 0; j < res_pres_test.GetLength(1); j++)
            {
                pres_test[0, j] = res_pres_test[i, j];
            }
            double[,] V_test = Vector(bet_test, pres_test);
//утворення вектору вводу
            double r = Form1.G(arrCov, arrM, V_test); // запуск
функція розподілення гіперплощини
            if (r < 0)
            {
                a1++; label2.Text = "Спроба вводу розпізнана як:
«АВТОРСЬКА» ";
            } // друк результатів
            else
            {

```

```

        n1++; label2.Text = "Спроба вводу розпізнана як:
«НЕ АВТОРСЬКА» ";
    }
}
//label3.Text = "Authors = " + a1 + " Nonauthors " + n1;
}
public static double G(double[,] arrCov, double[] arrM,
double[,] U) //
{
    double work = 0.0;
    for (int j = 0; j < arrCov.GetLength(0); j++) //
    {
        for (int k = 0; k < arrCov.GetLength(0); k++)
        {
            work = work + arrCov[j, k] * ((U[0, j] - arrM[j])
* (U[0, k] -
            arrM[k]));
        }
    }
    return (0.5 * work - 3.3735 * 3.3735); //
}

private void listView1_SelectedIndexChanged(object sender,
EventArgs e)
{
}

public static double Max(double[,] arr, int h) // максимальний
елемент в рядку
{
    double work = arr[h, 0];
    for (int i = 1; i < arr.GetLength(1); i++)
    {

```

```

        if (arr[h, i] > work) work = arr[h, i];
    }
    return work;
}

private void Form1_Load(object sender, EventArgs e)
{

}
}
}

```

Лістинг ZbirStatic.cs:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Text.RegularExpressions;

namespace LoginTimer1
{

    internal class ZbirStatic
    {
        public static double[] M(double work, double[,] arr) //
        розрахунок мат.сподівання
        {
            double[] arrM = new double[arr.GetLength(1)];
            for (int h = 0; h < arr.GetLength(1); h++)
            {
                for (int j = 0; j < arr.GetLength(0); j++)
                {

```

```

        work = work + arr[j, h]; // середнє
    }
    arrM[h] = work / arr.GetLength(0);
    work = 0;
}
return arrM;
}

```

розрахунок мат.сподівання

```

{
    double[] arrM = new double[arr.GetLength(1)];
    for (int h = 0; h < arr.GetLength(1); h++)
    {
        for (int j = 0; j < arr.GetLength(0); j++)
        {
            work = work + arr[j, h]; // середнє
        }
        arrM[h] = work / arr.GetLength(0);
        work = 0;
    }
    return arrM;
}

```

розрахунок дисперсій

```

{
    double[] arrD = new double[arr.GetLength(1)];
    for (int h = 0; h < arr.GetLength(1); h++)
    {
        for (int j = 0; j < arr.GetLength(0); j++)
        {
            work = work + Math.Pow(arr[j, h] - arrM[h], 2);
        }
    }
}

```

```

        arrD[h] = work / (arr.GetLength(0) - 1);
        work = 0;
    }
    return arrD;
}

public static double[] D(int[,] arr, double[] arrM, double
work) // розрахунок дисперсій
{
    double[] arrD = new double[arr.GetLength(1)];
    for (int h = 0; h < arr.GetLength(1); h++)
    {
        for (int j = 0; j < arr.GetLength(0); j++)
        {
            work = work + Math.Pow(arr[j, h] - arrM[h], 2);
        }
        arrD[h] = work / (arr.GetLength(0) - 1);
        work = 0;
    }
    return arrD;
}

public static double[,] Correl(double[] arrM, int[,] arr, int
n) // розрахунок матриці кореляцій
{
    double[,] arrKor = new double[arr.GetLength(1),
arr.GetLength(1)];
    for (int j = 0; j < arr.GetLength(1); j++)
    {
        for (int h = 0; h < arr.GetLength(1); h++)
        {
            //if (h == j + 1) arrKor[j, h - 1] = 1;
            arrKor[j, h] = UpCor(j, h, arrM, arr) / DownCor(j,
h, arrM,
arr);

```

```

        }
    }
    return arrKor;
}

public static double UpCor(int j, int h, double[] arrM, int[,]
arr) // розрахунок числівника в кореляції
{
    double res = 0;
    for (int i = 0; i < arr.GetLength(0); i++)
    {
        res = res + (arr[i, j] - arrM[j]) * (arr[i, h] -
arrM[h]);
    }
    return res;
}

public static double DownCor(int j, int h, double[] arrM,
int[,] arr) // розрахунок знаменника в кореляції
{
    double res = 0, res1 = 0, res2 = 0;
    for (int i = 0; i < arr.GetLength(0); i++)
    {
        res1 = res1 + Math.Pow((arr[i, j] - arrM[j]), 2);
        res2 = res2 + Math.Pow((arr[i, h] - arrM[h]), 2);
    }
    res = Math.Sqrt(res1) * Math.Sqrt(res2);
    return res;
}

public static double[,] Covariation(double[] arrM, int[,] arr,
int n) //розрахунок матриці коваріацій
{
    double[,] arrCov = new double[arr.GetLength(1),
arr.GetLength(1)];
    for (int j = 0; j < arr.GetLength(1); j++)

```

```

        {
            for (int h = 0; h < arr.GetLength(1); h++)
            {
                arrCov[j, h] = UpCor(j, h, arrM, arr) /
(arr.GetLength(0));
            }
        }
        return arrCov;
    }

    public static double[,] Dover(double[] arrM, double[] arrD,
int n) // розрахунок довірчих інтервалів
    {
        double[,] arrDov = new double[2, arrM.GetLength(0)];
        for (int i = 0; i < 2; i++)
        {
            for (int j = 0; j < arrM.GetLength(0); j++)
            {
                if (i == 0) arrDov[i, j] = arrM[j] -
                    1.9719 * Math.Sqrt(arrD[j]) / Math.Sqrt(n);
                else arrDov[i, j] = arrM[j] + 1.9719 *
Math.Sqrt(arrD[j]) /
                    Math.Sqrt(n);
            }
        }
        return arrDov;
    }

    public static double[,] Mass(string file) // текст в массив
    {
        string[] strArr = File.ReadAllLines(file,
Encoding.Default);
        int z = 0;
        foreach (string s in strArr)
        {

```

```

        strArr[z] = Regex.Replace(strArr[z], " +", "
").Trim();
        z++;
    }
    int m = strArr.Length;
    int n = Regex.Split(strArr[0], "\t").Count();
    double[,] arr = new double[m, n];
    int j = 0;
    for (int i = 0; i < m; i++)
    {
        foreach (string s in Regex.Split(strArr[i], "\t"))
        {
            arr[i, j] = Convert.ToDouble(s);
            j++;
        }
        j = 0;
    }
    return arr;
}

```

```

    public static void Print(int[,] arr, double[] arm, double[]
ard, double[,] arrKor, double[,] arrCov, double[,] arrDov, string
file) //запис в файл підрахунку
    {
        string str = "Матриця", work = "";
        for (int h = 0; h < arr.GetLength(0); h++)
        {
            str += Environment.NewLine + "\t";
            for (int j = 0; j < arr.GetLength(1); j++)
            {
                str += arr[h, j] + "\t";
            }
        }
    }
}

```

```

str += Environment.NewLine + Environment.NewLine + "Mx" +
"\t";

for (int i = 0; i < arm.Length; i++)
{ str = str + Math.Round(arm[i], 2) + "\t"; }
str += Environment.NewLine + Environment.NewLine + "Dx" +
"\t";

for (int i = 0; i < ard.Length; i++)
{ str = str + Math.Round(ard[i], 2) + "\t"; }
str += Environment.NewLine + Environment.NewLine + "sigma"
+ "\t";

for (int i = 0; i < ard.Length; i++)
{ str = str + Math.Round(Math.Sqrt(ard[i]), 2) + "\t"; }
str += Environment.NewLine + Environment.NewLine +
"Кореляція:" + Environment.NewLine;
for (int i = 0; i < arr.GetLength(1); i++)
{
    str += Environment.NewLine + "\t";
    for (int j = 0; j < arr.GetLength(1); j++)
    {
        str += Math.Round(arrKor[i, j], 3) + "\t";
    }
}
str += Environment.NewLine + Environment.NewLine +
"Коваріація:" + Environment.NewLine;
for (int i = 0; i < arr.GetLength(1); i++)
{
    str += Environment.NewLine + "\t";
    for (int j = 0; j < arr.GetLength(1); j++)
    {
        str += Math.Round(arrCov[i, j], 1) + "\t";
    }
}
str += Environment.NewLine + Environment.NewLine +
"Довірчі інтервали: " + Environment.NewLine;

```

```

        for (int i = 0; i < 2; i++)
        {
            str += Environment.NewLine + "\t";
            for (int j = 0; j < arm.GetLength(0); j++)
            {
                str += Math.Round(arrDov[i, j], 2) + "\t";
            }
        }
        File.WriteAllText(file, str);
    }
}

```

Лістинг Proverka.cs:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Text.RegularExpressions;

namespace LoginTimer1
{
    internal class Proverka
    {
        public static double[] Mint(double[,] arr) //розрахунок
нормованого мат.сподівання
        {
            double work = 0;
            double[] arrM = new double[arr.GetLength(0)];
            for (int h = 0; h < arr.GetLength(0); h++)
            {

```

```

        for (int j = 0; j < arr.GetLength(1); j++)
        {
            work = work + arr[h, j] / Form1.Max(arr, h);
            // Средние mint=  $\sum X[i,k]/\max(X[i,k])$ 
        }
        arrM[h] = work / arr.GetLength(1); // mint=1/n( $\sum$ )
        work = 0;
    }
    return arrM;
}

public static double[] Aritm(double[,] arr, double[] arrM)
//аритмічність набору
{
    double work = 0;
    double[] alf = new double[arr.GetLength(0)];
    for (int h = 0; h < arr.GetLength(0); h++)
    {
        for (int j = 0; j < arr.GetLength(1); j++)
        {
            work = work + Math.Pow(arr[h, j] / Form1.Max(arr,
h) - arrM[h], 2);
            //  $\sum (X[i,k]/\max() - \text{mint}[i])^2$ 
        }
        alf[h] = Math.Sqrt(work / (arr.GetLength(1) - 1));
        //√(1/(n-1))
        work = 0;
    }
    return alf;
}

public static double[] Speed(double[,] arrB, double[,] arrP)
//швидкість вводу
{

```

```

double work = 0;
double[] arrS = new double[arrB.GetLength(0)];
for (int h = 0; h < arrB.GetLength(0); h++)
{
    for (int j = 0; j < (arrP.GetLength(1)); j++)
    {
        work = work + arrP[h, j]; //  $\Sigma X[i,k]$ 
    }
    for (int j = 0; j < (arrB.GetLength(1)); j++)
    {
        work = work + arrB[h, j]; //  $+\Sigma Y[k,i]$ 
    }
    arrS[h] = work * 0.015; // (900/60)*1/1000 (измиллисек
    в сек)
    work = 0;
}
return arrS;
}

```

```

public static double[,] Difference(double[,] between)
{
    double[,] arrD = new double[between.GetLength(0), 3];
    double[] arr = new double[between.GetLength(1) - 1];
    double work = 0.0, plus = 0.0, exch = 0.0;
    for (int i = 0; i < between.GetLength(0); i++)
    {
        for (int j = 0; j < arr.GetLength(0); j++)
        {
            arr[j] = between[i, j + 1] - between[i, j];
        }
        if (arr[0] >= 0) plus += 1;
        for (int j = 1; j < arr.GetLength(0); j++)
        {
            if (arr[j] * arr[j - 1] <= 0) exch += 1;
        }
    }
}

```

```

        if (arr[j] >= 0) plus += 1;
    }
    arrD[i, 1] = plus / (arr.GetLength(0));
    arrD[i, 2] = exch / (arr.GetLength(0) - 1);
    work = plus = exch = 0;
}
return arrD;
}

public static double ns(double[,] res_bet, int h) // Число
перекриттів
{
    double ns = 0;
    for (int i = 0; i < res_bet.GetLength(1); i++)
    {
        if (res_bet[h, i] < 0) ns += 1;
    }
    return ns;
}
}
}

```

Лістинг Program.cs:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace LoginTimer1
{
    static class Program
    {
        /// <summary>

```

```
/// The main entry point for the application.
/// </summary>
[STAThread]
static void Main()
{
    Application.SetHighDpiMode(HighDpiMode.SystemAware);
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Application.Run(new Form1());
}
}
```

ДОДАТОК Д МАТЕРІАЛИ НАУКОВОЇ РОБОТИ



УДК 001(08)
А 43



Голова оргкомітету: Каренюк І.О.
Верстка: Зрада С.І.
Дизайн: Бондаренко І.В.



Конференцію зареєстровано Державною науковою установою «УкрІНТЕІ» в базі даних науково-технічних заходів України та інформаційному бюлетені «План проведення наукових, науково-технічних заходів в Україні» (Посвідчення №798 від 28.09.2021).

Матеріали конференції знаходяться у відкритому доступі на умовах ліцензії CC BY-NC 4.0 International.

А 43 **Актуальні питання та перспективи проведення наукових досліджень:** матеріали III Міжнародної студентської наукової конференції (Т. 3), м. Трускавець, 10 грудня, 2021 рік / ГО «Молодіжна наукова ліга». — Вінниця: ГО «Європейська наукова платформа», 2021. — 136 с.

ISBN 978-617-8037-43-7
DOI 10.38074/liga-inter-10.12.2021

ISBN 978-617-8037-46-8 (ТОМ 3)

Викладено матеріали учасників III Міжнародної мультидисциплінарної студентської наукової конференції «Актуальні питання та перспективи проведення наукових досліджень», яка відбулася 10 грудня 2021 року у місті Трускавець, Україна.

УДК 001 (08)

© Колектив учасників конференції, 2021
ISBN 978-617-8037-46-8 (ТОМ 3) © ГО «Молодіжна наукова ліга», 2021
ISBN 978-617-8037-43-7 © ГО «Європейська наукова платформа», 2021

ПОРІВНЯЛЬНИЙ АНАЛІЗ СПОСОБІВ ІНТЕГРАЦІЇ СИСТЕМ Молчанов К.Д., Науковий керівник: Хацько Н.С.	34
СПОСОБИ КЕРУВАННЯ НАВАНТАЖЕННЯМ ПРИ ТЕСТУВАННІ ПРОДУКТИВНОСТІ Газієва Н.П.	37

СЕКЦІЯ 21. СИСТЕМНИЙ АНАЛІЗ, МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЯ

ДОСЛІДЖЕННЯ ВІДКРИТИХ СИСТЕМ ДЛЯ РОЗВИТКУ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ Могиліук М.Л., Науковий керівник: Кульбовський І.І.	40
МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ІНФЕКЦІЙНИХ ЗАХВОРЮВАНЬ Алексейко В.О., Науковий керівник: Скрипник Т.К.	42
РЕТРОСПЕКТИВА РОЗВИТКУ МЕТОДІВ ПРОЄКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ Король І.В., Науковий керівник: Кульбовський І.І.	44
СИСТЕМНІ ТЕХНОЛОГІЇ ПРОЄКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ Яччук М.М., Науковий керівник: Кульбовський І.І.	46

СЕКЦІЯ 22. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

FEATURES OF SATELLITE NAVIGATION SYSTEM FOR AUTOMOTIVE TRANSPORT Trofimova Y.V., Scientific adviser: Fedik L.Y.	49
MODEL OF COMPETENCE LEVELS OF THE LOCAL COMMUNITY IN PROJECTS TO IMPROVE MUNICIPAL ENERGY EFFICIENCY Sorochan V., Scientific adviser: Sachenko O.	51
OVERVIEW AND PROBLEMS OF BIOMETRIC AUTHENTICATION METHODS Yezhova Y.A.	53
АВТОМАТИЗАЦІЯ ВЕНТИЛЯЦІЙНОЇ СИСТЕМИ Чайка Д.А., Науковий керівник: Шибко О.М.	56
ВИКОРИСТАННЯ 3D ТЕХНОЛОГІЙ У WEB РОЗРОБЦІ Гуменюк Д.В.	59
ВИКОРИСТАННЯ СИСТЕМИ АВТОМАТИЗОВАНОГО КЕРУВАННЯ В ПРОЄКТУВАННІ СИСТЕМ АВТОМАТИЗАЦІЇ Троянчук Б.В., Науковий керівник: Федік Л.Ю.	61
ВРАХУВАННЯ ГНУЧКОСТІ ТА АДАПТИВНОСТІ WEB-ДОДАТКУ ПРИ ВИКОРИСТАННІ У БУДЬ-ЯКІЙ СФЕРІ ДІЯЛЬНОСТІ Риґаль І.Т.	63
МЕТОД ВИЯВЛЕННЯ АНОМАЛІЙ В БІОСИГНАЛАХ НА ОСНОВІ МАШІННОГО НАВЧАННЯ Дека В.	65
ПЕРЕВАГИ ТА НЕДОЛІКИ АВТОМАТИЗАЦІЇ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ Сузілий Д.Р.	67

Yezhova Yelizaveta Alekseevna, an applicant for higher education at the faculty of CITA
Donetsk National Technical University, Ukraine

OVERVIEW AND PROBLEMS OF BIOMETRIC AUTHENTICATION METHODS

The paper discusses biometric authentication methods. Describes static and dynamic biometric authentication methods based on the use of biometric data such as fingerprints, iris, retina, hand geometry, face geometry, thermal imaging of the face, voice, and handwriting. The advantages of a dynamic biometric authentication method are disclosed.

The prospect of researching the development of this biometric authentication method is the development of methods and technologies for mathematical tools, algorithms, information and software in this subject area.

Keywords: authentication, biometric authentication, biometric characteristics, static methods, dynamic methods.

Introduction

Currently, the use of information systems (IS) has become an integral part of all spheres of human activity. A large amount of information with limited access is transmitted, stored and processed in an IS, which creates the need to ensure its information security [1–2].

Information protection in the IS is ensured by the creation of an integrated security system, one of its main components is the method of preventing unauthorized access.

The basis of software and hardware means of protection against tampering are the procedures of identification (the user calls himself) and authentication (authentication) of users [3].

The user authentication process is possible upon presentation:

- knowledge of something (password, personal number, etc.);
- possession of something (card, key, etc.);
- parts of the user himself (biometric characteristics) [4].

Biometric Authentication and Biometric Features

Biometric authentication has the following benefits:

- convenient to use due to the absence of the need for contact;
- it is impossible to lose or forget since biometric characteristics are part of the identifiable user;
- transmission and forgery of biometric data is seriously difficult.

Biometric authentication is a promising area for solving many of the problems that arise when using traditional authentication procedures.

Currently, there are two classes of biometric authentication methods (Table 1) [5].

Table 1

Biometric Authentication Methods	
Static	Dynamic
by fingerprint; along with the iris of the eye; on the retina of the eye; on the geometry of the hand; on the geometry of the face.	by voice; by handwriting.

Static Biometric Authentication Methods

The static method is based on the biological characteristics of a person who exists from birth. These cannot be lost, stolen or forgotten.

The invariability and openness of biometric data used in the static method allow biometric keys to be forged. Thus, there are more advantages to using dynamic biometric authentication methods.

Dynamic biometric authentication methods

Dynamic methods are based on the characteristics of human behaviour [6].

Voice authentication is widely used today. The main advantages of this method are ease of use and implementation, general availability and availability of many methods for building templates. However, due to various factors (health, age, mood), there are many problems with voice variability. The speech processing process itself is divided into the following stages: preliminary signal processing, criterion selection and recognition [7, 8].

Handwriting authentication is based on the details of hand movements when signing a document. There are two ways to process signature data: analysing the signature itself and analysing the dynamic characteristics of the record. User authentication based on handwriting keyboard input is also being studied and undoubtedly has prospects for further development.

The main difficulties and challenges in the field of biometric authentication

Among the disadvantages of biometric authentication is the following [9]:

- lack of well-trained professional staff;
- high cost of some biometric authentication devices;
- the complexity of setting up such systems;
- opposition from employees, since the management gets the opportunity to control all their movements and actually control the working time;
- cannot be used to identify people with some physical disabilities;
- the ability to use a dummy (for example, a fingerprint);
- unusual height (less than 1.55 m or more than 2.1 m), which can make it difficult to scan a face.

Based on the analysis of the shortcomings of biometric authentication, we will formulate the main problematic tasks that need to be resolved:

- training of highly qualified personnel;
- minimizing the cost of biometric devices, creating a budget line of products;
- simplification of setting up biometric systems;
- work with personnel to popularize this area;
- adaptation of biometric systems to identify people with certain physical disabilities;
- improvement of biometric identification systems to exclude the possibility of using a dummy.

Conclusion

Biometric authentication technology has broad development prospects. By using a system based on biometric methods, access to programs becomes faster, safer and easier. However, despite its many advantages, biometric technology still presents many challenges and challenges.

The introduction of biometric authentication is one of the priorities in the development of international relations. The use of biometric technology in IS can significantly simplify the identification process, reduce the likelihood of subjective errors

by controllers, and speed up the border control process.

Based on the analysis carried out, it can be concluded that the most accurate and best method of biometric authentication is authentication based on dynamic methods.

References:

1. Nyrkov A. P. Multiservice network of the transport industry / A. P. Nyrkov, S. S. Sokolov, A. S. Belousov // Computer and information technologies. 2014. No. 4 (118). S. 33–38.
2. Nyrkov A. P. Ensuring the safe functioning of the multiservice network of the transport industry / A. P. Nyrkov, S. S. Sokolov, A. S. Belousov // Report of the Tomsk State University of Control Systems and Radioelectronics. 2014. No. 2 (32). S. 143–149.
3. Methods and means of protection against unauthorized access // URL: <http://www.panashenko.ru/Articles/77/77.html> (date of access: 31.03.2016).
4. Identification and authentication, access control // URL: <http://citforum.ru/security/articles/galatenko> (date of access: 03/27/2016).
5. Biometric authentication systems // URL: https://ru.wikipedia.org/wiki/Biometric_authentication_systems#D0.A1.D1.82.D0.B0.D1.82.D0.B8.D1.87.D0.B5.D1.81.D0.BA.D0.B8.D0.B5.D0.BC.D0.B5.D1.82.D0.BE.D0.B4.D1.8B (date of access: 28.03.2016).
6. Dynamic methods of biometric identity authentication // URL: http://re.mipt.ru/infsec/2006/essay/2006_Dynamic_biometric_authentication_Cherkezo.pdf (date of access: 28.03.2016).
7. Shchemelinin V. L. Study of the stability of voice verification to attacks using the synthesis system / V. L. Shchemelinin, K. K. Simonchik // Higher educational institution instrument making. 2014. Vol. 57, No. 2. P. 84–88.
8. Tassov K. L. Method of human identification by voice / K. L. Tassov, R. A. Dyatlov // Engineering journal: science and innovations. 2013. Issue. 6 // URL: <http://engjournal.ru/catalog/it/biometric/1103.html>.
9. Bezmal V. Password protection: past, present, future / V. Bezmal // Computer Press 9. - Russia, 2008.

НАУКОВЕ ВИДАННЯ

МАТЕРІАЛИ ІІІ МІЖНАРОДНОЇ
СТУДЕНТСЬКОЇ НАУКОВОЇ КОНФЕРЕНЦІЇ

**«АКТУАЛЬНІ ПИТАННЯ ТА ПЕРСПЕКТИВИ
ПРОВЕДЕННЯ НАУКОВИХ ДОСЛІДЖЕНЬ»**

10 грудня 2021 рік • м. Трускавець, Україна

ТОМ 3

Українською, російською, англійською,
узбецькою і таджицькою мовами

*Всі матеріали пройшли перевірку на плагіат та експертизу за формальними ознаками
(форматування, стиль мови, оформлення цитувань та списку використаних джерел).
За точність викладеного матеріалу відповідальність несуть автори та їх наукові керівники.
Організаційний комітет не завжди поділяє позицію авторів.*

Підписано до друку 10.12.2021.
Папір офсетний. Цифровий друк. Формат 60×84/16.
Гарнітура Times New Roman, Poiret One та Arial.
Умовно-друк. арк. 7,9. Замовлення № 244.
Тираж: 50 екземплярів. Віддруковано з готового оригінал-макету.

Контактна інформація організаційного комітету:
Громадська організація «Молодіжна наукова ліга»
21037, Україна, м. Вінниця, вул. Зодчих, 40, офіс 103
Телефони: +38 098 1948380; +38 098 1956755
E-mail: info@liga.science

Видавець: ГО «Європейська наукова платформа».
21037, Україна, м. Вінниця, вул. Зодчих, 18, офіс 81. E-mail: info@ukrlogos.in.ua
Свідоцтво суб'єкта видавничої справи: ДК № 7172 від 21.10.2020.

Додаток Е

Матеріали презентації



Донецький національний технічний університет
Факультет комп'ютерно-інтегрованих технологій та автоматизації
Кафедра прикладної математики та інформатики

Біометрична аутентифікація користувача при введенні логіна та пароля

Розробка модуля системи розпізнавання користувача за клавіатурним почерком

Виконала: ст. гр. КІБ-18
Єжова Є. О.
Керівник: проф. кафедри ПМІ, д.т.н., проф.
Башков Є.О.

Луцьк - 2022

2

Об'єкт дослідження

- процеси захисту інформації від несанкціонованого доступу.

Предмет дослідження

- методи біометричної аутентифікації користувача при введенні логіна та пароля.

Мега роботи

- аналіз захищеності інформації з урахуванням аналізу клавіатурного почерку.

Завдання роботи

- аналіз сучасних видів аутентифікації користувачів;
- розробка методу аутентифікації за допомогою клавіатурного почерку;
- розроблення програмного модуля даного методу, використовуючи мову C#.

Наукова новизна

- полягає в розробці методу біометричної аутентифікації, шляхом використання клавіатурного почерку, а саме використання унікальних біометричних даних, що дозволило проводити аутентифікацію без необхідності втручання користувача безпосередньо в цей процес.

Практичне значення

- полягає в авторській розробці методу фільтрації авторського введення від неавторського та програмного модуля реалізації розробленого методу аутентифікації з використанням об'єктно-орієнтованої мови програмування C# на базі використанням інтегрованого середовища розробки Visual Studio.

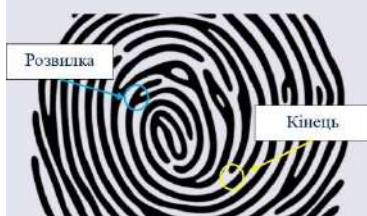
Огляд методів біометричної аутентифікації. Статистичні методи**За відбитком пальців**

Рисунок 1 – Види мініатюр

За райдушкою ока

Рисунок 2 – Типова система розпізнавання райдушкої оболонки

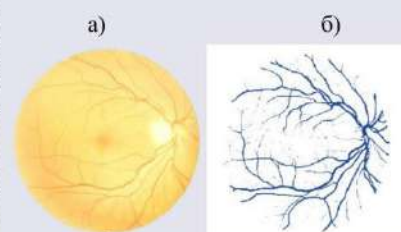
За сітківкою ока

Рисунок 3 – Кольорове зображення сітківки (а) та сегментоване зображення судин (б)

Огляд методів біометричної аутентифікації. Статистичні методи

За геометрією кисті



Рисунок 4 – Типова система аутентифікації за геометрією кисті

За геометрією обличчя



Рисунок 5 – Типова система аутентифікації обличчя

За термограмою обличчя

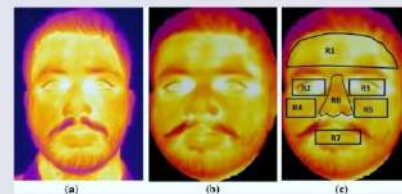


Рисунок 6 – Ілюстрація: а) термограми обличчя, б) витягу обличчя, с) характерних ділянок

Огляд методів біометричної аутентифікації. Динамічні методи

За голосом



Рисунок 7 – Система аутентифікації за голосом

За рукописним підписом



Рисунок 8 – Аутентифікація за рукописним підписом

Огляд методів біометричної аутентифікації. Порівняння видів біометричної аутентифікації



Рисунок 9 – Базова структура системи біометричної аутентифікації

Математична модель клавіатурного почерку користувача. Функція процесу набору тексту

Процес набору тексту користувачем можна описати функцією:

$$\vartheta(t) = \alpha(t) + \beta(t) + \gamma(t) \quad (1)$$

де $\alpha(t)$ – компонента, що характеризує підсвідомі процеси мислення під час набору тексту;

$\beta(t)$ – компонента, що характеризує свідомі процеси мислення;

$\gamma(t)$ – механічні властивості клавіатури, які впливають на процес набору тексту.

Враховуючи вищевикладене, ми можемо записати функцію $\vartheta(t, L)$ так:

$$\vartheta(t, L) = \alpha(t) + \Delta(t, L) \quad (2)$$

$$\vartheta(t, L) = (\alpha(t) + \Delta(t, L))T(t, L) \quad (3)$$

де $T(t, L)$ – функція кореляції, яка враховує залежні від часу зміни параметрів набору тексту користувачем.

Математична модель клавіатурного почерку користувача. Вектор вхідних параметрів

$$V = \{t_1^{\text{пауза}}, t_2^{\text{пауза}}, \dots, t_{n-1}^{\text{пауза}}, m^{\text{пауза}}, \beta, s, d_1, d_2\} \quad (4)$$

де $t_i^{\text{пауза}}$ – час між відпусканням i -ої клавіші та натисканні $(i + 1)$ -ої;

$m^{\text{пауза}}$ – нормоване математичне очікування довжини інтервалів;

β – розкид часу між натисканням двох клавіш;

s – швидкість вводу;

y_1, y_2 – параметри зміни часових інтервалів.

$$t_i^{\text{утрим}} = t_i^{\text{відпуск}} - t_i^{\text{натиск}}, \quad i = 1 \dots n \quad (5)$$

$$t_i^{\text{пауза}} = t_{i+1}^{\text{натиск}} - t_i^{\text{відпуск}}, \quad i = 1 \dots (n - 1) \quad (6)$$

де $t_i^{\text{утрим}}$ – час утримування i -ої клавіші в натиснутому стані;

$t_i^{\text{пауза}}$ – час паузи між відпусканням i -ої клавіші та натисканням $(i + 1)$ -ої;

n – довжина ключової фрази в випадку даної роботи – логіна.

Математична модель клавіатурного почерку користувача. Вектор вхідних параметрів

Розкид часу між натисканням двох клавіш виглядає наступним чином:

$$\beta = \sqrt{\frac{\sum_{i=1}^{n-1} (t_i^{\text{пауза}} - m^{\text{пауза}})^2}{n - 2}} \quad (7)$$

Швидкість набору користувачем логіна розраховується як:

$$s = \frac{\sum_{i=1}^{n-1} t_i^{\text{пауза}} + \sum_{i=1}^n t_i^{\text{утрим}}}{60} \quad (8)$$

$$dt_i = t_{i+1}^{\text{пауза}} - t_i^{\text{пауза}}, i = 1 \dots n - 1 \quad (9)$$

$$y_1 = \frac{f}{k} = \frac{f}{n - 1} \quad (10) \quad y_2 = \frac{h}{g} = \frac{h}{n - 2} \quad (11)$$

Опис процесу біометричної аутентифікації користувача

$$V_{Ci} = \{v_1, v_2, \dots, v_j, \dots, v_N\}, j = 1 \dots N \quad (12)$$

Центр розподілу векторів V_{Ci} розташовується в точці $(\xi_1, \xi_2, \dots, \xi_N)$, (яка визначається N математичними очікуваннями $m_{v_1} = \xi_1, m_{v_2} = \xi_2, m_{v_3} = \xi_3, \dots, m_{v_N} = \xi_N$).

$$Q = \|\lambda_{jk}\| = \left\| \begin{pmatrix} \lambda_{11} & \dots & \lambda_{1N} \\ \dots & \dots & \dots \\ \lambda_{N1} & \dots & \lambda_{NN} \end{pmatrix} \right\|$$

де

$$\lambda_{jk} = M(v_j - \xi_j) \cdot (v_k - \xi_j) = \begin{cases} \sigma_{ij}^2 & \text{при } j = k \\ cov\{v_j, v_k\} & \text{при } j \neq k \end{cases} j, k = 1..N \quad (13)$$

Опис процесу біометричної аутентифікації користувача

$$\frac{1}{2} \cdot \sum_{j=1}^N \sum_{k=1}^N \Lambda_{jk} \cdot (v_j - \xi_j) \cdot (v_k - \xi_j) = \{C[L, (1 - P_1)]\}^2 \quad (14)$$

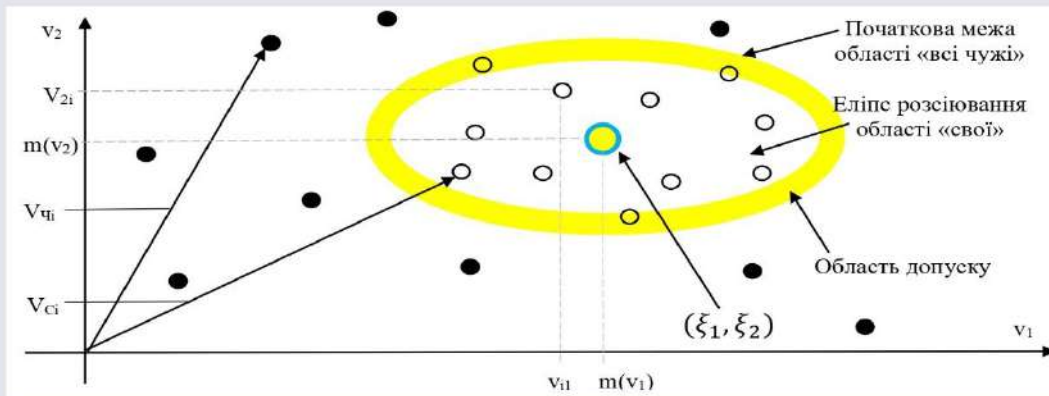


Рисунок 10 – Ілюстрація поділу областей «свої» від області «всі чужі» у разі 2-мірного простору

Опис практичної реалізації



Рисунок 11 – Етапи розробки та створення системи біометричної аутентифікації за клавіатурним почерком

Опис практичної реалізації

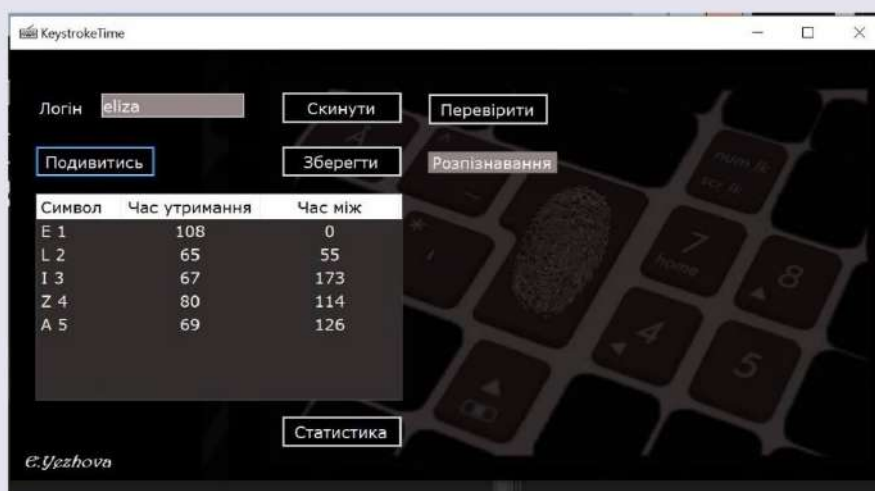


Рисунок 12 – Вхідні параметри вводу

Опис практичної реалізації

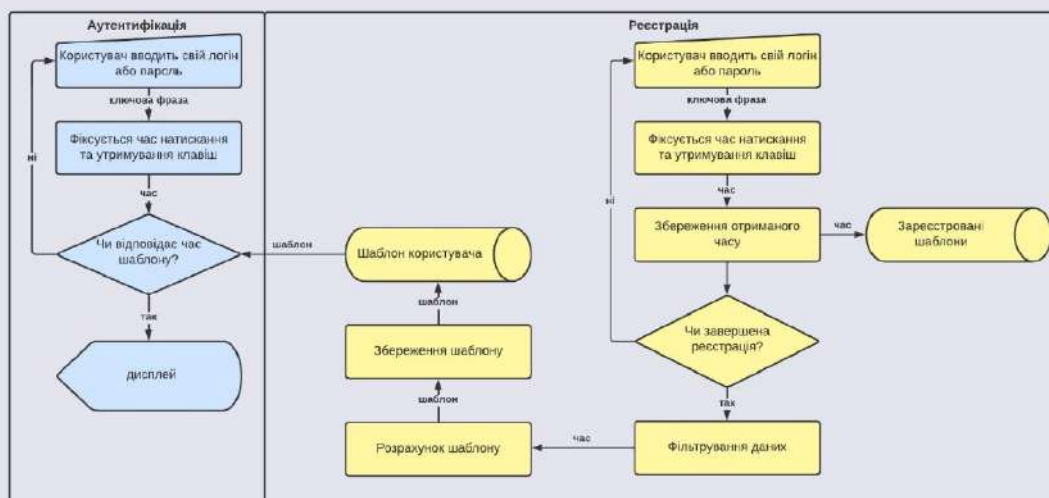
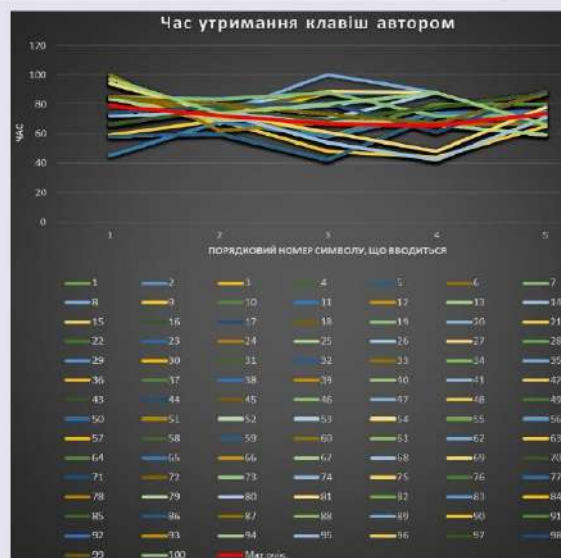
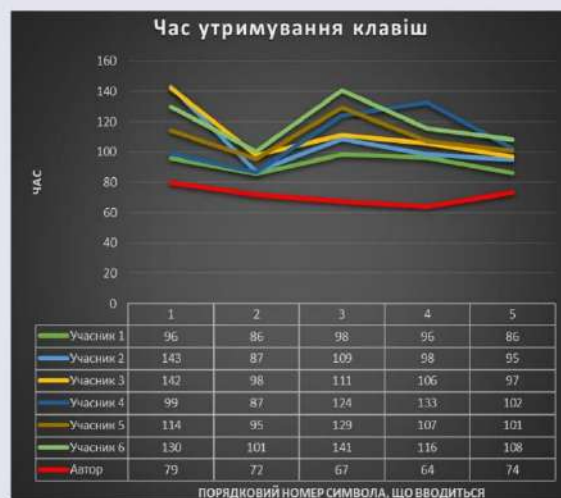


Рисунок 13 – Схема програмного модуля

Результати дослідження параметрів клавіатурного почерку



Результати дослідження параметрів клавіатурного почерку



Тестування працездатності програмного модуля

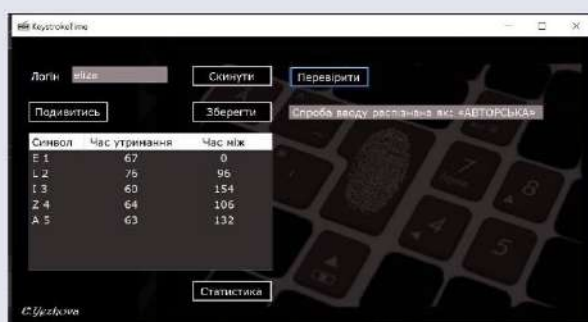


Рисунок 14 – Приклад вводу логіна «eliza», що розпізнаний авторським

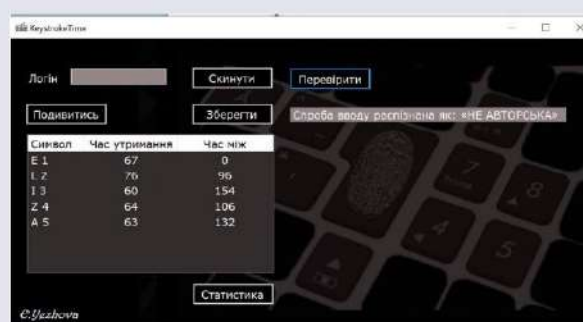


Рисунок 15 – Приклад вводу логіна «eliza» іншим учасником, що розпізнаний не авторським

Висновки

Проводячи аналіз виконаної роботи, можна дійти висновку, що вдалось розв'язати усі поставлені на початку виконання роботи задачі.

Під час даної роботи було досліджено аутентифікації користувача по клавіатурному почерку під час введення паролльної фрази.

На основі проведеного аналізу та розробки математичної функції розподілу областей «свої» та «чужі» був розроблений модуль для фільтрації авторського вводу від неавторського.

В результаті тестування якого було виявлено, що він досить добре розрізняє спроби вводу.

Використовуючи даний модуль можна створити гібридну систему керування доступом, яка буде поєднувати в собі паролльні та біометричні методи аутентифікації. Таким чином, довершена система керування буде забезпечувати посилену, в порівнянні з класичної паролльною аутентифікацією, процедуру аутентифікації.

На тему дипломної роботи опубліковано:

- Єжова Є.О., Overview and problems of biometric authentication methods – Трускавець: Актуальні питання та перспективи проведення наукових досліджень: матеріали III Міжнародної студентської наукової конференції (Т. 3), м. Трускавець, 10 грудня, 2021 рік / ГО «Молодіжна наукова ліга». – Вінниця: ГО «Європейська наукова платформа», 2021. – 136 с. – 53–55 с.

Дякую за увагу!