Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

Пояснювальна записка
до магістерської кваліфікаційної роботи

на тему: «Модель та засіб автентифікації за клавіатурним почерком з використанням штучних імунних мереж»
08-20.МКР.013.00.000 ПЗ

Виконав: студент 2 курсу, групи 1БС-18м
Спеціальність 125 Кібербезпека
ОПП Безпека інформаційних і комунікаційних систем
_____ Рудик О.А.

Керівник: к. т. н., проф. каф. ЗІ
_____ Кондратенко Н.Р.
Рецензент
к. т. н., доц., доц. каф. ОТ
_____Крупельницький Л.В.

Вінниця – 2019 рік

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації
Освітньо-кваліфікаційний рівень магістр
Спеціальність 125 Кібербезпека
ОПП Безпека інформаційних і комунікаційних систем

ЗАТВЕРДЖУЮ
Завідувач кафедри ЗІ, д. т. н., проф.
_____ В. А. Лужецький
_____ 2019 року

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Рудику Олександру Анатолійовичу

1. Тема роботи: «Модель і засіб автентифікації за клавіатурним почерком з використанням штучних імунних мереж» керівник роботи: Кондратенко Наталія Романівна, к. т. н., проф. каф. ЗІ, затверджена наказом ректора ВНТУ №254 від 02.10.2019 року

2. Строк подання студентом роботи _____2019 р.

3. Вихідні дані до роботи:
   - Мова програмування Python
   - Операційна система Windows 10, оперативна память 2гб
   - Метод захисту – засіб для автентифікації

4. Зміст розрахунково-пояснювальної: Вступ. Аналіз відомих підходів для обробки даних при автентифікації за клавіатурним почерком. Аналіз нейромережевого підходу. Реалізація нейромережі. Висновки. Перелік використаних джерел. Додатки.

5. Перелік ілюстративного матеріалу: Результати роботи нейро мереж(плакат, А4). Схема роботи засобу автентифыкації за клавіатурним почерком (плакат, А4). Таблиця результатів роботи засобу(плакат, А4

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Кондратенко Н. Р., к. т. н., проф. каф. ЗІ		
2	Кондратенко Н. Р., к. т. н., проф. каф. ЗІ		
3	Кондратенко Н. Р., к. т. н., проф. каф. ЗІ		
4	Мацкевічус С.С., ст. викл. каф.ЕПВМ		

7. Дата видачі завдання _____2019 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз завдання. Вступ	01.09.19 – 04.09.19	
2	Аналіз літературних джерел за напрямком бакалаврської дипломної роботи	05.09.19 – 15.09.19	
3	Науково-технічне обгрунтування	16.09.19 – 21.09.19	
4	Розробка технічного завдання		
5	Розробка рішень	23.09.19 – 30.09.19	
6	Практична реалізація, моделювання, експериментування, результати	30.09.19 – 10.09.19	
7	Розробка розділу економічного обгрунтування доцільності розробки	14.10.19– 10.11.19	
8	Аналіз виконання ТЗ, висновки	11.11.19 – 17.11.19	
9	Оформлення пояснювальної записки	18.11.19 – 24.11.19	
10	Попередній захист МКР	25.11.19 – 30.11.19	
11	Перевірка магістерської роботи на наявність плагіату	28.11.19 – 01.12.19	
12	Виправлення зауважень, підготовка ілюстративного матеріалу	02.12.19 – 10.12.19	
13	Представлення МКР до захисту, рецензування	11.12.19 – 14.12.19	
14	Захист МКР	16.12.19 – 20.12.19	

Студент _____ Рудик О.А.
(підпис)

Керівник роботи _____ Кондратенко Н. Р.
(підпис)

АНОТАЦІЯ

У магістерській кваліфікаційній роботі проведено дослідження методів обробки даних при автентифікації за клавіатурним почерком. Проаналізовані методи автентифікації та виділені їх переваги і недоліки. Було обрано метод виявлення на основі штучних нейронних мереж. На основі виконаних досліджень розроблено програмний засіб для автентифікації користувачів за клавіатурним почерком на основі штучної імунної системи.

ABSTRACT

In the master degree thesis the research of methods of data processing during authentication by a keyboard handwriting was conducted. The authentication methods are analyzed and their advantages and disadvantages are highlighted. The method of detection on the basis of artificial neural networks was chosen. On the basis of the conducted researches the software for user authentication by a keyboard handwriting based on an artificial immune system.

ЗМІСТ

ВСТУП

Поява розподілених обчислювальних мереж і сучасних засобів телекомунікацій для зберігання, обробки і передачі інформації призвело до принципіально нових можливосте їх використання в самих різних сторонах людської діяльності. Зберігання конфіденційних даних в розподілених обчислювальних мережахповязано з необхідністю розмежування доступу до інформації та автентифікації користувачів.

Спосіб автентифікації за допомогою спеціалізованих пристроїв, як і спосіб парольної автентифікації, має недолік, а саме необхідність від користувача володіти певним ключем, який можна викрасти або продублювати. У випадку використання біометричних систем автентифікації ключем виступає сама людина, що суттєво підвищує стійкість таких систем до несанкціонованого доступу.

Одним з перспективних напрямів забезпечення безпеки системи є використання методу автентифікації за клавіатурним почерком з використанням штучних нейромереж, які вже довели свою ефективність у вирішенні складних задач розпізнавання, класифікації, управління і виявлення. Застосування ШНМ дозволить створити ефективну адаптивну систему автентифікації і підвищити рівень захисту комп'ютерних систем від несанкціонованого доступу.

Обєктом магістерської кваліфікаційної роботи процеси автентифікації користувачів за клавіатурним почерком.

Предмет дослідження – імунні детектори пристосовані для розвязання задачі автентифікації.

Метою роботи є підвищення точності автентифікації користувача за клавіатурним почерком на основі штучних імунних систем.

Для досягнення поставленої мети потрібно вирішити такі завдання:

- проаналізувати відомі методи обробки даних клавіатурного почерку;
- виконати моделювання архітектури нейроподібної системи;
- розробити програмний засіб;
- провести тестування розробленого засобу.

Наукова новизна. Запропоновано модель і засіб автентифікації користувачів за клавіатурим почерком з використанням штучних імуних

мереж, яка відрізняється розширеними функціональними можливостями за рахунок впровадження імунних детекторів.

Практична цінність полягає у розробці засобу автентифікації за клавіатурним почерком з використанням штучних імунних мереж у вигляді програми для автентифікації користувачів. Відбулася апробація запропонованої моделі автентифікації за клавіатурним почерком

Окремі результати роботи доповідались на XLVII та XLVIII Науково технічних конференціях підрозділів Вінницького національно технічного університету в 2018р. та 2019р.

1 АНАЛІЗ ОСНОВНИХ МЕТОДІВ АВТЕНТИФІКАЦІЇ КОРИСТУВАЧА ЗА КЛАВІАТУРНИМ ПОЧЕРКОМ

1.1 Класифікація існуючих методів автентифікації

Автентифікація являє собою процес порівняння інформації, що надається користувачем, з еталонною. Залежно від типу інформації її можна віднести до одного з чотирьох основних факторів, або до їх комбінації:

- фактор знання(парольна автенифікація);
- матеріальний фактор(апаратна автентифікація);
- біофактор (біометрична автентифікація).

Розглянемо кожен з факторів.

На рисунку 1.1 зображена класифікація методів автентифікації

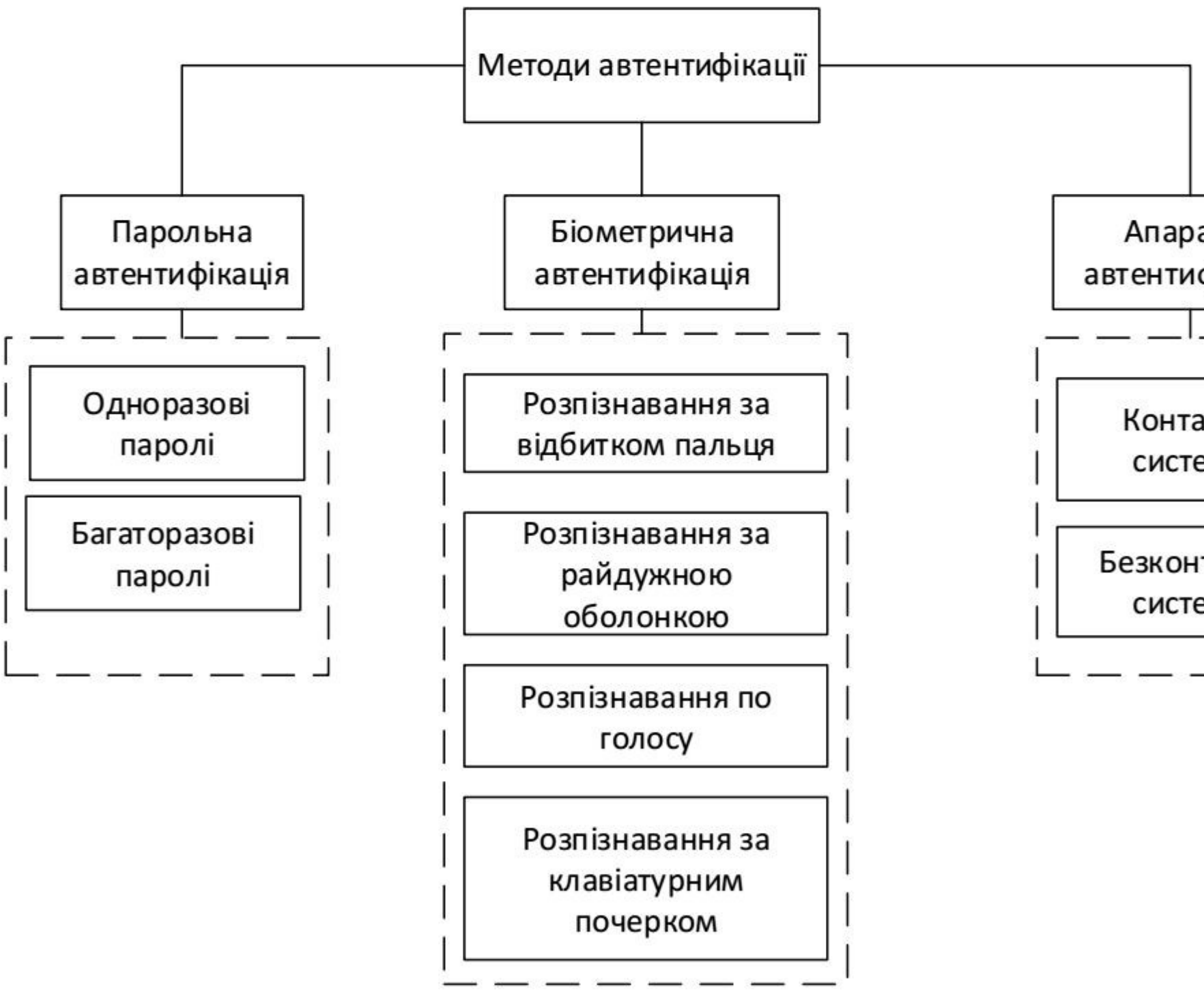


Рисунок 1.1 – Класифікація методів автентифікації

Головна перевага парольної аутентифікації – простота й звичність. Паролі давно вбудовані в операційні системи й інші сервіси. При правильному використанні, паролі можуть забезпечити прийнятний для багатьох організацій рівень безпеки. Проте, по сукупності характеристик їх варто визнати дуже слабким засобом перевірки дійсності.

Також з ростом складності пароля він все важче для запам'ятовування. Дослідження SafeNet від 2004 року виявило, що 47% респондентів забували свої паролі протягом року [1]. А з ростом кількості облікових записів від різних комп'ютерних систем, яких з кожним роком стає все більше, ситуація ще більш ускладнюється. Здатність запам’ятовувати паролі була вивчена в лабораторії Bу в 2007 році. Після першого тижня 12.5% учасників забули їх шести символьні буквеноцифрові паролі. З учасників, які повинні були пам'ятати паролі про п'ять облікових записів, 25% забули принаймні один [2].

Переносні пристрої для аутентифікації – пристрої, які використовуються для збереження аутентифікатора (інформації, що використовується для аутентифікації) на спеціальному матеріальному носії[3].

Токени максимально спрощують процедуру введення та зміни аутентифікатора. Перевагою систем аутентифікації на основі токенів є відсутність необхідності запам’ятовування аутентифікатора (окрім, можливо, нескладного pin-коду). Це дозволяє використовувати достатньо довгі ключі, а за необхідності, надає можливість реалізувати процедуру зміни ключа при кожному вході в систему, мережу тощо. Також виключається можливість підглядання інформації, яка використовується для аутентифікації.

При зазначених перевагах ця система аутентифікації має певні недоліки.

Токен можна загубити або його можуть вкрасти (від негативних наслідків в цьому випадку захищає pin-код та алгоритми блокування або стирання інформації після фіксованого числа невірних спроб введення pin-коду);

Використання токенів у більшості випадків вимагає наявності додаткового обладнання на терміналах аутентифікації інформаційно-комунікаційної системи;

1.2 Біометрична автентифікація

Біометричні системи забезпечують найбільш точну автентифікацію, оскільки перевіряють параметри, які дуже важко або неможливо змінити або підробити. Їхні переваги очевидні, оскільки традиційні системи захисту не здатні з'ясувати, наприклад, хто саме вводить код або вставляє смарт-картку.

Біометрія - це автентифікація людини по унікальних, властивих тільки йому біологічним ознакам. В теперішній час існує безліч методів біометричної автентифікації користувачів комп'ютерних систем, які можна розділити на дві великі групи: статичні і динамічні [4]. Класифікація методів біометричної автентифікації представлена на рис. 1.2.

Аналіз методів біометричної автентифікації показав, що статичні методи, грунтовані на характеристиці людини, тобто унікальній властивості, даній йому від народження і невід'ємному від нього, разом з достоїнствами (висока точність автентифікації, висока швидкість реакції та ін.) мають ряд недоліків (висока вартість устаткування, витратних матеріалів і обслуговування) [1, 3]. Методи динамічної автентифікації грунтуються на поведінковій (динамічній) характеристиці людини, т. е. побудовані на особливостях, характерних для підсвідомих рухів в процесі відтворення якої-небудь дії. Аналіз методів динамічної автентифікації показав, що, незважаючи на ряд їх недоліків (висока імовірність помилки автентифікації і неправдивих спрацьовувань системи безпеки), в силу простоти і доступності вони залишаються затребуваними в секторі невеликих установ, підприємств і організацій [1, 2].

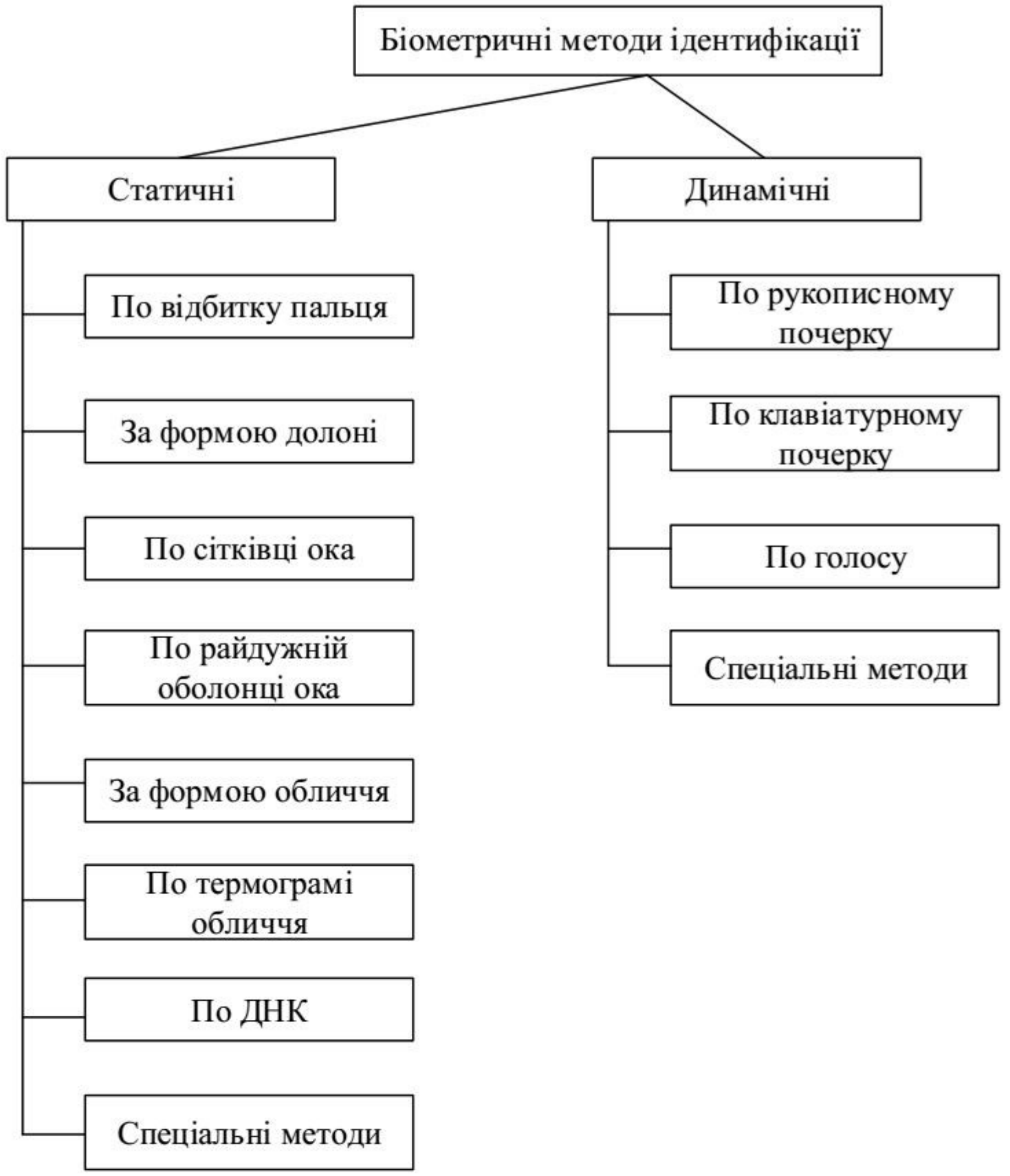


Рисунок 1.2 - Класифікація методів біометричної автентифікації користувачів

Найбільш надійним з практично реалізованих методів вважається метод сканування сітківки ока[5]. Тому він використовується в системах контролю доступу на особливо секретні об'єкти. Недоліком є висока вартість систем із використанням цього методу.

Ідентифікація за формою долоні або за геометрією кисті базується на побудові трьохвимірного зображення кисті руки[6]. Для здійснення ідентифікації знімаються такі характеристики пальців чи долоні, як довжина, ширина, товщина та параметри поверхні шкіри. Недоліком методу є зміни кисті руки протягом життя, що спричиняє низьку надійність[7].

Розпізнавання за венами руки базується на використанні знімків з зовнішньої та внутрішньої сторін руки. Оскільки гемоглобін крові поглинає інфрачервоне випромінювання, ступінь відбиття променів зменшується і вени стають видимими у вигляді чорних ліній. А рисунок вен у кожної людини є індивідуальним. Сканування можна робити безконтактно. Ця технологія за надійністю є порівняною з ідентифікацією за райдужною оболонкою ока. Недоліком є вплив деяких хвороб, зокрема, артриту. А перевагою є менш дороге обладнання при високій точності[8].

Вищеперераховані методи належать до статичних[4], які використовують фізіологічні параметри людини, що не змінюються в часі. Крім них є динамічні методи, які базуються на індивідуальних поведінкових особливостях людини. До них належать голосова ідентифікація, ідентифікація за підписом, за клавіатурним почерком, за біоелектричною активністю мозку, тощо. В основі цього механізму лежить велечезна складність задачі керування рухами людини. Формально цю задачу можна представити в вигляді деякої спрощеної моделі з трьома виходами і великою кількістю входів керування мязами. Подібна умовна модель зображена на рис. 1.3. Три входи моделі X(t), Y(t), Z(t) умовно відображають траекторію руху точки (точок) організма, що нас цікавлять, в тривимірному просторі.

В першому наближенні відносна складність задачі може бути оцінена по числу входів моделі, або по числу мязів, які беруть участь у тому чи іншому виді руху. Таким чином під час письма ручкою задіюються мязи більшості пальців руки і мязи передпліччя[9]. Всього може бути задіяно більше 50 мязів, але найбільший вплив мають лише 10 мязів. Тобто при управлінні рукою під час письма людині доводиться в реальному часі вирішувати як мінімум десятивимірну задачу керування.

При роботі з клавіатурою додатково підключається ще приблизно 20 мязів плеча і плечового поясукожної руки, тобто під час друку в сліпу двома

руками потенційно можуть бути задіяні 140 мязів. Виходячи з припущення, що лише 20% з всього числа мязів мають найбільший вплив, отримаємо 28-вимірну задачу керування.

Варто звернути увагу, що в усіх розглянутих випадках задача керування, по-перше багатовимірна (як мінімум 10-вимірна), а по-друге число керуючих входів завжди суттєво перевищує число входів.

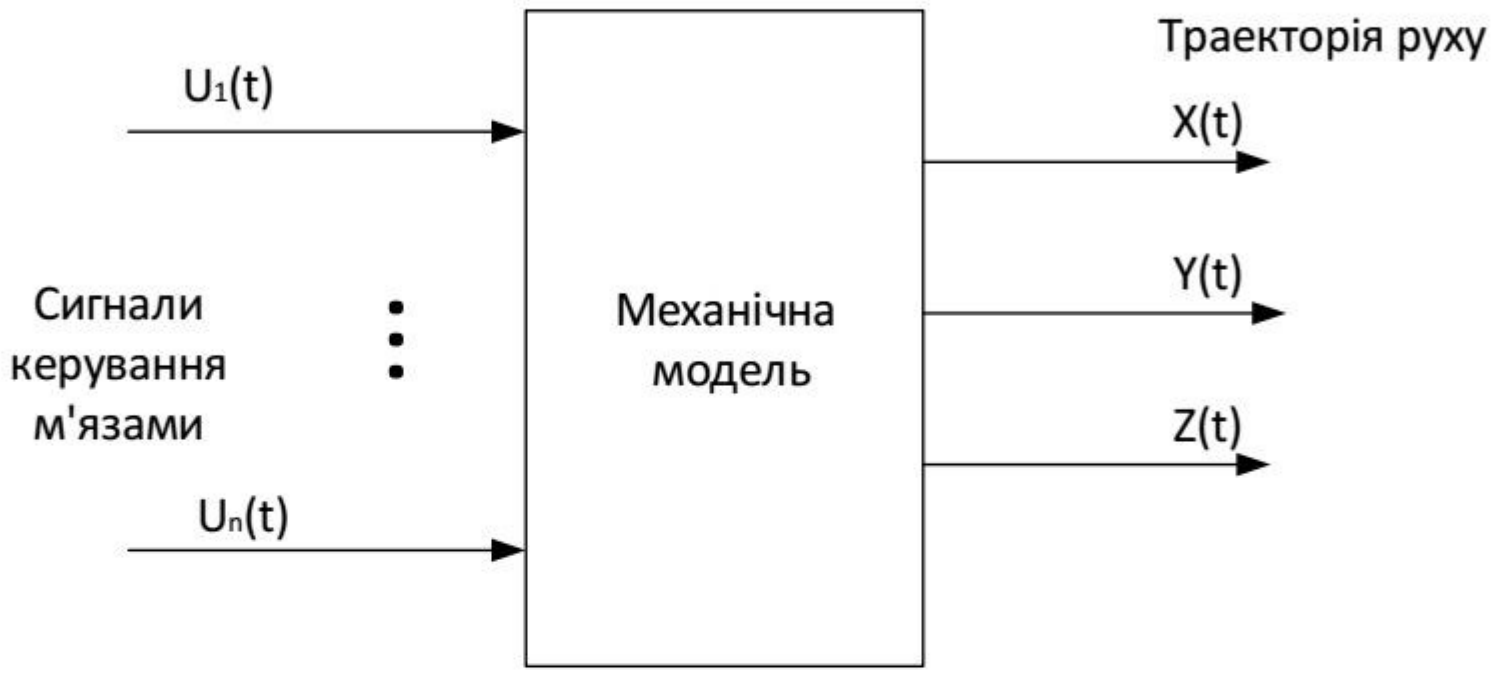


Рисунок 1.3 – Багатовимірна модель керування рухами

Одним із методів, які дозволяють розпізнавати особу на відстані та негласно, є голосова ідентифікація. Перевагами є дешевизна цього методу та відсутність психологічного дискомфорту під час ідентифікації. Під час ідентифікації за голосом аналізуються висота тону, модуляція, інтонація тощо. Але надійність і точність цього методу не є високими, оскільки голос може залежати від стану здоров'я та поведінкових факторів.

Одним з найбільш звичних для нас методів ідентифікації особи є її підпис. Якщо підпис як графічне зображення можна підробити, то поведінку руки особи під час підпису скопіювати неможливо. Біометричний метод ідентифікації людини за підписом базується на аналізі швидкості руху руки, сили тиску та тривалості виконання етапів підпису.

Метод ідентифікації за клавіатурним почерком схожий на ідентифікацію за підписом, але тут використовується введення кодового слова на стандартній

клавіатурі комп'ютера. Основною характеристикою є динаміка набору кодового слова. Перевагою є використання звичайного комп'ютера. Такий метод наразі не є поширеним, але розробки в цій галузі здійснюються[10].

Крім того, важливим фактором надійності є те, що вона абсолютно ніяк не залежить від користувача. І дійсно, при використанні парольного захисту людина може використовувати коротке ключове слово або тримати папірець з підказкою під клавіатурою комп'ютера. При використанні апаратних ключів недобросовісний користувач буде недостатньо строго стежити за своїм токеном, в результаті чого пристрій може потрапити до рук зловмисника. У біометричних же системах від людини не залежить нічого.

Задача великої розмірності дозволяє зробити препущення про унікальність клавіатурного почерку користувача і стверджувати про доцільність використання динамічних методів автентифікації користувачів на практиці і в часності методу атентифікації користувача за клавіатурним почерком.

1.3 Метод автентифікації за клавіатурним почерком

Часові інтервали між натисканням клавіш на клавіатурі і час утримання (натискання) клавіші дозволяють достатньо однозначно охарактеризувати почерк роботи користувача на клавіатурі. При цьому часові інтервали між натисканням клавіш характеризують темп роботи, а час утримання клавіш характеризує стиль роботи з клавіатурою (різкий удар чи плавне натискання).

Автентифікація користувача за клавіатурним почерком можлива по набору ключової фрази і довільного тексту [11].

Принципова відмінність цих двох способів заключається в тому, що в першому випадку – ключова фраза задається користувачем в момент регістрації в системі (пароль), а в другому випадку використовуються ключові фрази, які генеруються системою кожен раз в момент ідентифікації

користувача. Обидва способи передбачають два режима роботи – навчання і автентифікація.

На етапі навчання користувач вводить певну кількість разів запропоновані йому текстові фрази. При цьому розраховуються і запамятовуються еталонні характеристики данного користувача. На етапі автентифікації розраховані оцінки порівнюються з еталонними, на основі чого робиться висновок про співпадіння чи неспівпадіння параметрів клавіатурного почерка.

Еталонні характеристики користувача отримані на етапі навчання системи дозволяють зробити висновок про стабільність клавіатурного почерку користувача і визначити довірчий інтервал розсіювання параметрів для подальшої автентифікації користувача. В сучасних системах автентифікації за клавіатурним почерком задля уникнення дискредитації роботи системи виконується відсіювання користувачів, клавіатурний почерк яких не є достатньо стабільним (табл. 1.1), що в свою чергу обмежує використання данного способу автентифікації для широкого загалу.

Таблиця 1.1 Оцінка стабільності клавіатурного почерку користувача.

Помилки, %	Аритмічність, %	Швидкість зн./хв.	Характеристика перекриття		Оцінка
			Число перекриття, %	Число задіяних пальців	
менше 2	менше 10	більше 200	більше 50	Всі	Відмінно
менше 4	менше 15	більше 150	більше 30	Більшість	Добре
менше 8	менше 20	більше 100	більше 10	Декілька	Задовільно
більше	більше 20	менше 100	менше 10	По одному	Незадовільно

Початковий етап обробки даних – фільтрація. На цьому етапі з потоку даних видаляється інформація про службові клавіші – клавіші керування курсором, функціоальні клавіші і тому подібне.

Після цього виділяється інформація про храктеристики користувача: кількість помилок при наборі; інтервали між натисканням клавіш; час

утримання клавіш; число перекриття між клавішами; ступінь аритмічності при наборі; швидкість набору.

Після статистичної обробки цих даних, обраховані еталонні характеристики користувача зберігаються в базі даних.

В загальному вигляді, функція μ(t), що описує процес набору тексту користувачем на клавіатурі, може бути описана наступним чином:

$$\mu(t) = \gamma(t) + \theta(t), \quad (1.1)$$

де $\gamma(t)$ – складова, що характеризує підсвідомі процеси мислення при наборі тексту, $\theta(t)$ – складова свідомих процесів мислення при наборі тексту, $\lambda(t)$ – механічна характеристика клавіатури, що вливає на процес набору тексту.

Поведені експерименти показали, що ймовірність автентифікації користувача за часом утримання клавіш в залежності від довжини ключової фрази є більш стабільною характеристикою клавіатурного почерку користувача аніж час між натисканням клавіш (паузи), який росте зі збільшенням довжини ключової фрази. Це пояснюється тим, що процес натискання клавіші на клавіатурі являється чисто підсвідомим процесом мислення. Після визначення необхідних параметрів потрібно сформувати вектор вхідних даних.

Нехай n – довжина ключової фрази, що використовується для автентифікації користувача, t_i^{down}, t_i^{up} – час натискання і час відпускання i-ї клавіші, c_i – ASCII код i-ї клавіші ($i = \overline{1,n}$), тоді

$$t_i^{press} = t_i^{up} - t_i^{down},\ i = \overline{1,n} \quad (1.2)$$

$$t_i^{pause} = t_{i+1}^{down} - t_i^{up},\ i = \overline{1,n-1}$$

де t_i^{press} – час утримання i-ї клавіші в натиснутому положенні, t_i^{pause} – довжина паузи між відпусканням i-ї клавіші і натисканням (i+1)-ї клавіші. Так як входи нейромережі мають діапазон значень від нуля до одиниці, необхідно нормувати получені величини до відповідних меж.

$$t_i^{press} = \frac{t_i^{up} - t_i^{down}}{t_{max}^{press}},\ i = \overline{1,n}; \qquad (1.3)$$

$$t_i^{pause} = \omega^{pause} + (\frac{t_{i+1}^{down} - t_i^{up}}{t_{max}^{pause}}),\quad i = \overline{1,n-1};$$

де t_{max}^{press} і t_{max}^{pause} – максимальні значення відповідних параметрів. На праці варто використовувати $t_{max}^{press} = 1.25\,c.$ так як данна величина є часом ввімкнення клавіатурою компютера режиму автоповтору клавіші, і час утримання клавіші користувачем впринципі не може перевищувати дане значення. Мінімально допустима швидкість друку s_{min} , необхідна автентифікації користувача по клавіатурному почерку складає 100 символів за хвилину (табл.1), відповідно очікуване середнє значення пауз m^{pause} повинно приблизно складати:

$$m^{pause} = \frac{s_{min}}{60} = \frac{100}{60} \approx 1{,}6\ \text{с}; \qquad (1.4)$$

Враховуючи, що середній час утримання клавіші в натиснутому положенні складає 240 мс, в якості значення t_{max}^{pause} достатньо прийняти $t_{max}^{pause} = 1{,}5\text{с}$. Теоретично можливе виникнення пауз в процесі набору текста довших ніж t_{max}^{pause}, тому необхідно ввести функцію ω^{pause}, яка буде контролювати діапазон величин t_i^{pause}:

$$\omega^{pause}(t) = \begin{cases} t, & t \le 1; \\ 1, & t > 1; \end{cases} \qquad (1.5)$$

Аритмічність швидкості набору α і нормалізоване математичне очікування $\bar{m}^{pause}$ обраховуються наступним чином:

$$\bar{m}^{pause} = \frac{\sum_{i=1}^{n-1} \omega^{pause}(\frac{t_i^{pause}}{t_{max}^{pause}})}{(n-1)};\quad \alpha = \sqrt{\frac{\sum_{i=1}^{n-1}(\omega^{pause}\left(\frac{t_i^{pause}}{t_{max}^{pause}}\right) - \bar{m}^{pause})}{n-2}}; \qquad (1.6)$$

Аритмічність часу утримання клавіш β і нормалізоване математичне очікування $\bar{m}^{press}$ обраховуються наступним чином:

$$\bar{m}^{press} = \frac{\sum_{i=1}^{n} t_i^{press}}{n t_{max}^{press}};\beta = \sqrt{\frac{\sum_{i=1}^{n}\left(\frac{t_i^{press}}{t_{max}^{press}}\right) - \bar{m}^{press})}{n-1}}; \qquad (1.7)$$

Нормована швидкість набору $\bar{s}$:

$$\bar{s} = \frac{n}{\frac{(t_n^{up} - t_1^{down})}{60} s_{max}}; \quad (1.8)$$

де s_{max} – максимальна швидкість набору тексту, що відповідає 900 знаків за хвилину[12].

Наявність факту перекриття між і-ю і (і+1)-ю клавішами визначається при виконанні умови $t_i^{down} \leq t_{i+1}^{up}$ і $t_{i+1}^{down} \leq t_i^{up}$. В протилежному випадку факт перекриття клавіш відсутній. Для клавіш в яких відслідковується факт перекриття ітераційно виконується наступий процес:

$$n_c = n_c + 1; \quad (1.9)$$

$$t_c = t_c + t_i^{up} + t_{i+1}^{down};$$

Необхідно підрахувати число клавіш з перекриттям і сумарний час перекриття. Далі розраховується нормований середній час перекриття $\bar{t}_c$ і середньоквадратичне відхилення d_c:

$$\bar{t}_c = \frac{t_c}{n_c t_{max}^{press}}; \quad d_c = \sqrt{\frac{\sum_i (\frac{t_i^{up} - t_{i+1}^{down}}{t_{max}^{press}} -)^2}{n_c - 1}} \quad (1.10)$$

В результаті отримуємо вектор вхідних параметрів ν, необхідний для автентифікації користувача за клавіатурним почерком при фіксованій довжині ключової фрази і його довжину l:

$$\nu = \{t_1^{press}, \dots, t_n^{press}, t_1^{pause}, \dots, t_n^{pause}, \bar{c}_1, \dots, \bar{c}_n, \bar{m}^{pause}, \alpha, \bar{m}^{press}, \beta, \bar{s}, \bar{t}_c, d_c\};$$

$\bar{c}_i$ – нормоване значення ASCII кодів клавіш:

$$\bar{c}_i = \frac{c_i - 32}{223}; \quad (1.11)$$

У відповідності до (1) необхідно розбити сформований вектор вхідних параметрів на ν_γ і ν_θ, для характеристики підсвідомих і свідомих процесів відповідно:

$$\nu_\gamma = \{t_1^{press}, \dots, t_n^{press}, \bar{m}^{press}, \beta, \bar{t}_c, d_c\};$$

$$\nu_\theta = \{t_1^{pause}, \ldots, t_{n-1}^{pause}, \bar{m}^{pause}, \alpha, \bar{s}, \}; \quad (1.12)$$

$$\nu = \nu_\gamma + \nu_\theta + \{\overline{c_1}, \ldots, \overline{c_n}\};$$

В сформований вектор вхідних параметрів включається не тільки величини, що відображають характеристики клавіатурного почерка користувача, але і сама ключова фраза, а саме ASCII коди натиснутих клавіш, що дозволяє підвищити надійність роботи системи в режимі ідентифікації користувача.

Порівняння характеристик клавіатурного почерку може відбуватися з використанням імовірнісно-статистичних методів, гістаграмного методу і за допомогою нейронних мереж.

1.4 Імовірнісно-статистичний метод порівняння характеристик клавіатурного почерку

Один з підходів до автентифікації по клавіатурного почерку, заснований на аналізі функції щільності розподілу ймовірностей випадкових змінних[13]. Цей метод поділяється на чотири етапи: створення моделі, групування спостережень, оцінка критеріїв і прийняття рішень. Модель включає в себе тимчасові характеристики НН, НО і ОН, що визначаються по набору на клавіатурі поточного слова. Групування спостережень здійснюється за допомогою класифікації по зростаючій ієрархії, що використовує в якості відстані між двома профілями евклідова відстань. Профіль характеризується вектором математичних сподівань μ і відхилень σ. Вектор Vi, що визначає репрезентативність і -ї характеристики, задається як:

$$V_i = m / n_i$$

де ni - загальна кількість даних, зареєстрованих для і-ї характеристики; mi - кількість даних, відібраних для подальшого обчислення і-й характеристики.

Потім розглядається гіпотеза, відповідно до якої дані спостережень розподілені за законом Коші:

$$F_{\mu,\sigma}(x) = \frac{\sigma^3}{(\sigma^3 + |x-\mu|^3)}$$

Виходячи з цього формується функціонал:

$$\Psi_i = \alpha_0 * F_{\mu,\sigma}(x) * V_i,$$

де α0 - коефіцієнт, який використовується для нормалізації ваги.

Потім обчислюється сума значень таких функціоналів, що становлять глобальний функціонал:

$$\Psi = \sum \Psi_i .$$

Значення глобальних функціоналів також передбачаються розподіленими за законом Коші з математичними очікуваннями і дисперсіями, обчисленими для всіх користувачів, що дозволяє визначити наступну сумарну оцінку:

$$I = \alpha V_1 + \beta V_2 + \gamma V_3 + \delta V_4 ,$$

де $V_k = F_{\mu\sigma}(S)$, S_k - оцінки, отримані для вектора характеристик НН, НО, ОН та ОО; α, β, γ, δ - деякі параметри.

Далі сумарна оцінка порівнюється з допустимим значенням для відбору або видалення відповідного спостереження.

1.5 Гістаграмний метод порівняння характеристик клавіатурного почерку

Пропонується метод розпізнавання клавіатурного почерку особистості, який полягає в розбитті області розподілу клавіатурних ознак на багатовимірні прямокутні області та отриманні оцінки щільності розподілу як зваженої суми розподілу ознак по областям[14].

Виділимо в r-вимірному просторі R^r обмежену область I^r, яка містить всі зразки навчальної вибірки, і приведемо цю область до початку координат.

Визначимо діапазон зміни кожного компонента для всіх зразків навчальної вибірки

$$\vartheta_{j\,min} = \min \vartheta_j\{V_i\}, i = \overline{1, L;}$$

$$\vartheta_{j\,max} = \max \vartheta_j\{V_i\}, i = \overline{1, L;}$$

Приведені до початку координат значення всіх компонент векторів утворені як

$$\widehat{\vartheta_j} = \vartheta_{j\,max} - \vartheta_{j\,min}$$

визначають координати наведеної r-мірної області розподілу векторів.

Розіб'ємо область I^r на l однакових непересічних r-мірних прямокутних підобластей і підрахуємо число зразків m_k навчальної вибірки, що потрапили в кожну підобласть.

Тепер можна зробити оцінку щільності розподілу векторів:

$$\widetilde{W}(V) = \frac{1}{L}\sum_{k=1}^{l}\frac{m_k\{V_i\}}{O_k}$$

де O_k – міра області I_k, що обчислюється за формулою

$$O_i = \prod_{j=1}^{r}\widehat{V_{iJ}}$$

У відповідності з властивостями функції щільності ймовірності для отриманої оцінки щільності розподілу векторів справедливе співвідношення:

$$\int \widetilde{W}(V)dV = 1$$

Основна перевага гістограмного методу оцінки щільності розподілу - його простота і ясний фізичний зміст. Крім того, не потрібна постійна апріорна інформація про поведінку щільності розподілу ознак, крім її позитивності і відсутності стрибків у всій області свого визначення.

1.6 Нейромережевий підхід порівняння характеристик клавіатурного почерку

Сучасний підхід до задачі автентифікації пов'язаний з використанням нейромереж. Нейронні мережі – це узагальнена назва декількох груп алгоритмів, які повязані однією властивістю – вони вміють навчатися на прикладах, аналізуючи приховані закономірності з даних. Якщо між вхідними і вихідними даними існує якийсь звязок, навіть якщо його не можливо виявити традиційними кореляційними методами, нейронна мережа здатна автоматично налаштуватись на цей звязок з заданою степенню точності. Особливість штучних нейронних мереж заключається в тому, що моделюючи інформаційні процеси своїх біологічних прообразів, вони демонструють якості притаманні живим організмам, а саме: здатність до навчання, аптацію до зовнішніх впливів, що змінюються, елементи самоорганізації.

Використання нейромережевого підходу в системах автентифікації користувача дозволяє однаково успішно виконувати як процес автентифікації так і процес ідентифікації користувача за однаковий час і дозволяє вирішити ряд проблем, що виникають при використанні стандартних методів статистичної обробки вхідного потоку данних – відмова від аналізу типу статистичного розподілення, і враховуючи здатність нейромереж до фільтрації випадкових завад присутніх у вхідних даних, відмовитися від алгоритмів сгладжування екперментальних даних, необхідних при використанні апарату, побудованого на методах статистичного аналізу.

"Нейронні мережі" використовують алгоритм, який встановлює відповідність унікальних параметрів особи що перевіряється і параметрів шаблону, що знаходиться в базі даних, при цьому застосовується максимально можливе число параметрів.

У міру порівняння визначаються невідповідності між особою що перевіряється і шаблону з бази даних, потім запускається механізм, який за

допомогою відповідних вагових коефіцієнтів визначає ступінь відповідності особи, що перевіряється шаблоном з бази даних. Цей метод збільшує якість автентифікації особи у складних умовах. Мінімальною реалізацією нейронної мережі є двошарова нейронна мережа, що складається з вхідного (розподільчого), проміжного (прихованого)і вихідного шару[15].

Нейронні мережі не програмуються в звичайному розумінні цього слова, вони навчаються. Можливість навчання — одна з головних переваг нейронних мереж перед традиційними алгоритмами[16].

Технічно навчання полягає в знаходженні коефіцієнтів зв'язків між нейронами. У процесі навчання нейронна мережа здатна виявляти складні залежності між вхідними даними і вихідними, а також виконувати узагальнення [17].

Вище була описана модель клавіатурного почерку користувача яка була представлена функцією (1), зазначалось, що складова $\theta(t)$, яка відповідає за свідомі процеси мислення під час друку не достатньо стабільна і нею досить часто нехтують, замінюючи на кореляційну функцію, яку отримують емпірично на основі аналізу відповідних статистичих даних.

Функція ймовірності автентифікації від пауз між натисканням клавіш має максимум свого значення при довжині ключової фрази 8-10 символів. Це пояснюється тим, що ключові фрази не великої довжини, які складаються з одного, максимум двох слів, користувач набирає підсвідомо. Підсвідомі рухи стабільні до тих пір, поки в них не втручається вищий рівень свідомого мислення, що призводить до появи ефекту «сороконіжки». Поява даного ефекту пояснює зменшення ймовірності автентифікації користувача при перевищенні довжини ключової фрази критичного рівня. Значення цього рівня може сильно варіюватися для користувачів з різним досвідом роботи з клавіатурою і може приймати діапазон значень від 6 до 30 символів. Після цієї межі, навіть у самих досвідчених операторів копютерного набору

спостерігається ефект підключення свідомого мислення і зупинки в наборі тексту для прийняття рішення.

Однак, така характеристика як паузи між натисканням клавіш є дуже важливою для систем автентифікації за клавіатурним почерком, так як з цієї характеристики виводиться багато додаткових даних, які можна використати в системі як еталонні, а чим більше еталонних характериистик буде на етапі навчання і автентифікації, тим точніше буде працювати система.

В загальному випадку для аналізу вектора вхідних даних використовується нейромережа персептронного типу, щоб уникнути збоїв при роботі мережі, характеристику, що відповідає за свідомий вплив під час друку замінюють на кореляційну функцію. Для нейромережі імунного типу характерне покращене виявлення звязку між вхідними і етлонними даними, що дозволить використати характеристику, яка відповідає за паузи між натисканням клавіш, в повній мірі.

1.6.1 Математична модель штучної імунної системи

Штучні імунні системи є новою областю інформаційних технологій. Моделі і алгоритми штучних імунних систем є спробою реалізації ідей, закладених в природній імунній системі вищих ссавців.

Складна структура імунної системи і широка функціональність дозволяють знайти їй застосування в багатьох областях, причому всі штучні імунні системи використовують спрощення і тільки ті характерні риси та структурні елементи, які потрібні для вирішення конкретного завдання. В нашому випадку для вирішення задачі ідентифікації буде використано принцип афінності.

Аффінність - це характеристика, яка оцінює ступінь близькості (схожості) генетичних наборів антигену і антитіла. Для штучних імунних систем, заснованих на бінарному коді, аффінність оцінюється за допомогою

відстані Хеммінга. Для систем, заснованих на матеріальному коді - за допомогою евклідової відстані.

На сьогоднішній день в апараті ШІС існують три провідні теорії: теорія клонального відбору [21], теорія ідеоптичних мереж Ерне [22] і теорія небезпеки [23].

Результатом дослідження в теорії клонального відбору являється алгоритм CLONALG, який можна коротко описати наступним чином:

1. Відкрити базу даних зразків клавіатурних почерків і сформувати початкову популяцію (MP – main population).
2. Получити дані невідомого клавіатурного почерку (антигену) і розрахувати енергію звязку (афінність) з елементами із MP (антитілами).
3. Антитіла з кращою афінністю клонувати, а клони мутувати.
4. Сформувати проміжну популяцію з клонів (TP – temporary population0).
5. Вибрати з TP найкращі клони і записати їх в базу даних. Видалити з бази даних найгірші антитіла і згенерувати нові.
6. Виконувати пункти 2-5 поки не виконається критерій алгоритму.

На теорії небезпеки базується алгоритм DCA, який являється бінарним класифікатором. Принцип роботи алгоритму заснований на механізмі роботи біологічної дендридної клітини. Згідно теоріїї небезпеки такі клітині здатні запускати імунну відповідь, збираючи і аналізуючи сигнали, що вказують на аномальну смерть клітин в організмі. В DCA модель дендридної клітини виконує аналіз даних невідомого клавіатурного почерку, як сигналів, з подальшим їх віднесенням до одного з ідентифікуючих класів.

Кожен з перерахованих алгоритмів здатен вирішувати спеціалізований клас задач, проте для рішення задачі автентифікації користувача за клавіатурним почерком необхідне обєднання цих алгоритмів в один алгоритм мультиклональної селекції для використання всіх переваг зазначених алгоритмів для рішення задачі.

Для задачі автентифікації введемо наступні поняття. Математичне визначення антитіла: At = <Mas, inf>, де Mas – масив ознак (генів). Кожна

ознака представляє собою час утримання клавіш та час пауз між натисканнями в мілісекундах. inf – значення антитіла. Mp = {At} – основна популяція антитіл. Ag = <Mas> - антиген. Ступінь подібності або аффінність Ag-At можна обрахувати за допомогою манхетенської метрики:

$$D = \sum_{i=1}^{l} |at_i - ag_i|$$

де l - кількість елементів масиву генів, at_i - і-й ген антитіла At, ag_i - і-й ген антигену Ag. Для зміни або мутації генів використана формула отримана експериментальним шляхом.

$$Pm(at_i) = var * D_{ati} * Km/D$$

де var – число, що випадково приймає значення -1 і 1 і визначає напрям мутації, Km - емпірично встановлений коефіцієнт (принятий за 17000),D_{ati} - афінність між at_i і і-им геном ag_i антигену Ag.

Для розрахунку розмірів проміжної популяції F(D) запропоновано використовувати формулу 5, також отриману емпірично. Вона дозволяє отримувати клони в інтервалі від 0 до n+1, пропорційно їх афінності. Антитіла в проміжній популяції з меншою аффінністю виробляють найменшу кількість нащадків, а антитіла з більшою аффінносгью найбільше, навіть якщо мінімальна і максимальна аффінність в проміжній популяції приймає значення в межах 93 – 95% схожості.

$$F(D) = n * \frac{D * (D - D_S)}{100 * (D_{max} - D_S)} + 1$$

де, D – афінність між антитілом At та антигеном Ag. n – ціле число (прийнято за 6), що визначає максимальну кількість клонів. Ds – середня афінність між всіма антитілами і антигеном, D_{max} – максимальна афінність між антитілом і антигеном, отримана на даній ітерації імунного алгоритму.

Розглянуто основні підходи до автентифікації користувача, а також проаналізовано їх основні переваги та недоліки. Розглянуто основні методи

автентифікації за клавіатурним почерком, а саме алгоритми: нейромережевий, ймовірнісно-статистичний та гістограмний. Дано основні властивості, принципи функціонування цих алгоритмів, при описі яких були вказані основні використані ідеї. Для нейромережевих систем виділено і описано їх основні особливості, які можуть бути використані при вирішенні практичних проблем з широкого кола предметних областей, і конкретно для поставленої задачі. Розглянувши особливості роботи з клавіатурним почерком можна зробити висновок, що існує залежність психофізичного стану людини, яка робить загальновідомі методи обробки харакетирстик клавіатуного почерку мало ефективними. І навіть нейромережевий підхід при його можливості знаходити закономірності між даними не здатен в повній мірі вирішити поставлену задачу. Тому є актуальним використання такої нейроподібної структури як штучна імунна система яку можна використати для вирішеня поставленої задачі, через можливість зміни початкових даних.

2 МАТЕМАТИЧНА МОДЕЛЬ ЗАСОБУ АВТЕНТИФІКАЦІЇ ЗА КЛАВІАТУРНИМ ПОЧЕРКОМ З ВИКОРИСТАННЯМ ШТУЧНИХ ІМУННИХ СИСТЕМ

Поширення сучасних інформаційних технологій і розвиток глобальної компютерної мережі Internet робить привабливим метод автентифікації користувача за клавіатурним почерком в компютерних мережах. В першу чергу це обумовлено тим, що пристрої для зчитування інформації про параметри клавіатурного почерка користувача присутні на кожному персональному компютері, що призводить до низької економічної вартості використання даного способу автентифікації на практиці. Автентифікація користувача по клавіатурному почерку також дозволяє суттєво посилити парольний захист, що використовується в сучасних операційних системах для доступа до локальних чи глобальних інформаційних ресурсів.

2.1 Модель нейоподібної мережі для розпізнавання клавіатурного почерку

Структура нейроподібної мережі зображена на рис. 2.1

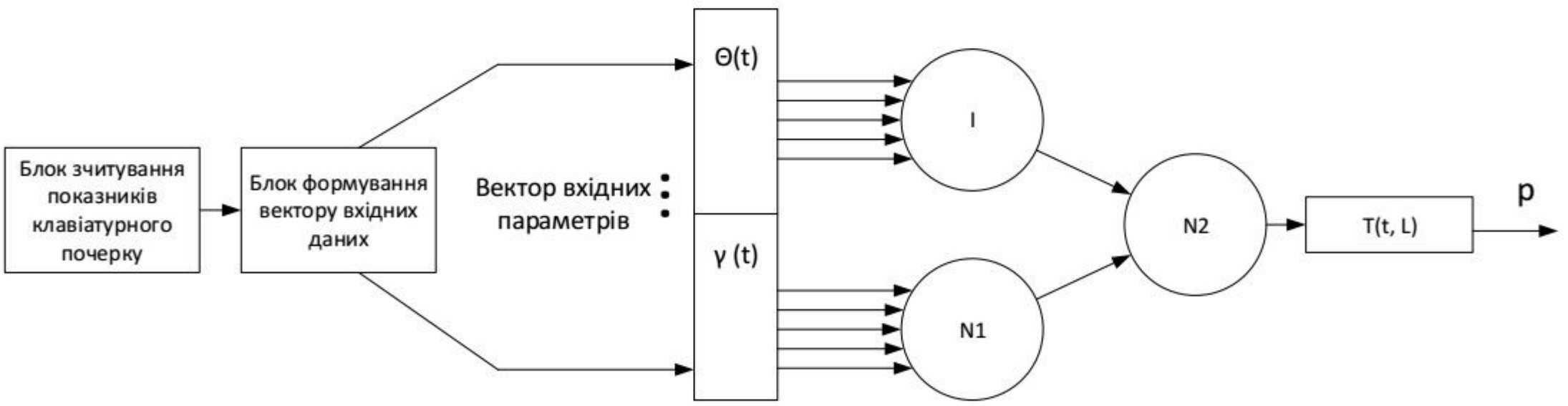


Рисунок 2.1 – Блок-схема модуля для збору даних

Вектор вхідних параметрів, що формується в результаті визначення часових характеристик клавіатурного почерку користувача подається на блок

в якому відбувається розділеня на компоненти, що описують підсвідомі і свідомі процеси мислення. Отриманий вектор $\theta(t)$, що включає в себе час між натисканнями клавіш, надходить на вхід блоку I, що представляє собою нейроподібну мережу імунного типу. Вектор $\gamma(t)$, що відповідає за час утримання клавіш, надходить на блок N_1 – нейромережу персептронного типу. Нейромережа N_3 виконує остаточну автентифікацію користувача і подає на вихід значення ймовірності, що характеризує з якою ймовірністю вхідні параметри відповідають параметрам зареєстрованого користувача. Отриманий результат корелюється за допомогою функції T(t, L), в залежності від досвіду роботи з клавіатурою.

2.1.1 Блок збору даних для формування вектору вхідних даних

Програма для збору даних про часові характеристики користувача виконана за допомогою механізму хуків за наступним алгоритмом. Спочатку відбувається завантаження тексту для збору інформації через відкриття готового текстового файлу. При завантажені файлу починає виконуватись функція перехоплення: очікується натискання клавіші від користувача, якщо клавіша натиснута перехоплюється ASCII код натиснутої клавіші а також фіксується час в мілісекундах, ці дані записуються до фунції логування з типом події KEY_DOWN, якщо натиснуту клавішу відпустили також відбувається перехоплення коду клавіші та час відпускання, отримані дані записуються з типом події KEY_UP. Після закінчення роботи з програмою результати роботи функції логування необхідно записати до log файлу, де буде зберігатись інформація про події, час та коди клавіш.

На рисунку 2.2 зображена блок-схема модуля для збору даних.

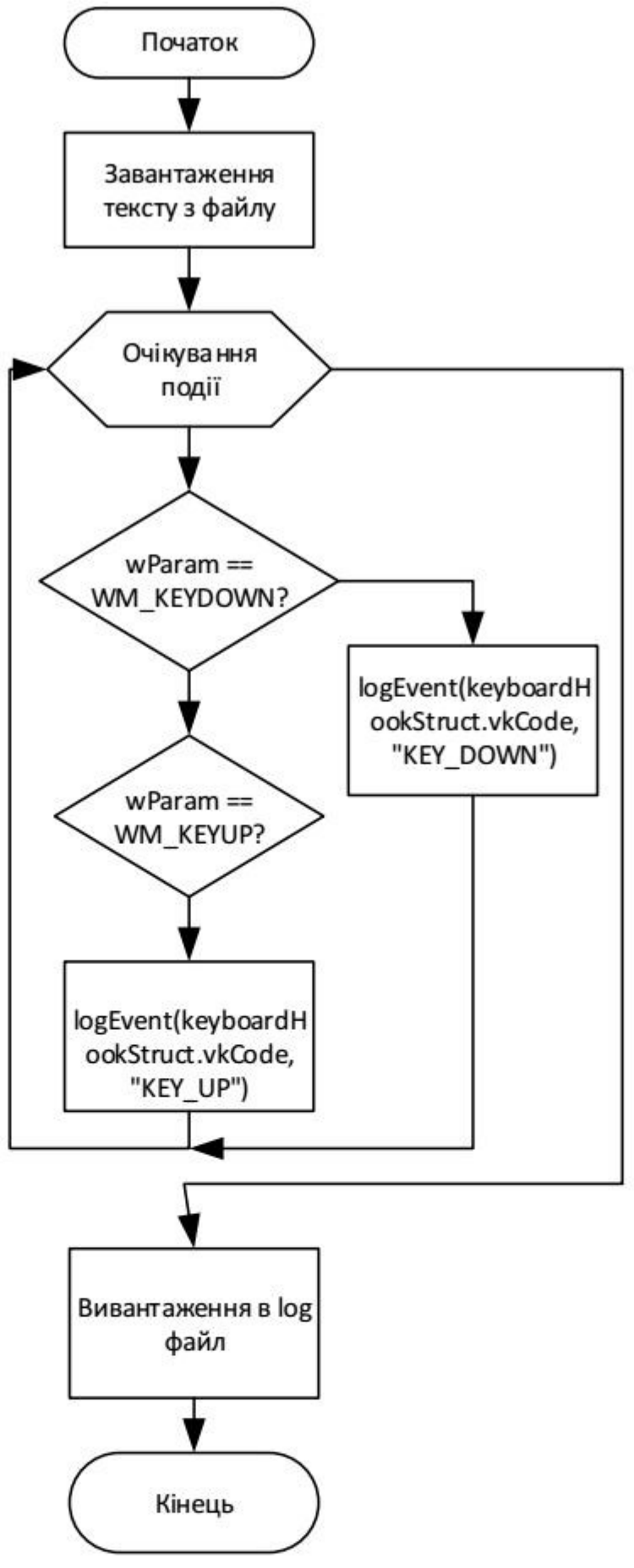


Рисунок 2.2 – Блок-схема модуля для збору даних

2.1.2 Блок нейромережі персептронного типу

На вхід нейромережі надходять дані про час утримання клавіш які обраховуються наступним чином: $t_i^{pause} = t_{i+1}^{down} - t_i^{up}$. Так як нейромережа може працювати лише з даними в діапазоні від 0 до 1 всередині нейрону відбувається нормалізація вхідних значень: $t_i^{press} = \frac{t_i^{up} - t_i^{down}}{t_{max}^{press}}$. В якості

вагових коефіцієнтів використовується аритмічність набору тексту та математиче сподівання: $\bar{m}^{press} = \frac{\sum_{i=1}^{n} t_i^{press}}{n t_{max}^{press}}$; $\beta = \sqrt{\frac{\sum_{i=1}^{n}\left(\frac{t_i^{press}}{t_{max}^{press}}\right) - \bar{m}^{press})}{n-1}}$.

На рисунку 2.3 зображена блок-схема нейромережі персептронного типу.

Рисунок 2.3 – Блок-схема нейро-мережі персептронного типу

Дані завантажуються на вхід нейромережі з Exel таблиці. Вхідні дані представляються у вигляді матриці. Відбувається процес перевірки даних: на вхідному шарі виконується нормалізація даних матриці, в прихованому та вихідному шарах відбувається обрахування даних з урахуванням ваг та функції активації. Виходом нейронної мережі є масив ймовірнісних

рахактеристик, що надсилаються до наступної нейромережі для прийнятя остаточного рішення.

2.1.3 Блок нейроподібної мережі імунного типу

На рисунку 2.4 зображена блок-схема нейроподібної мережі імунного типу.

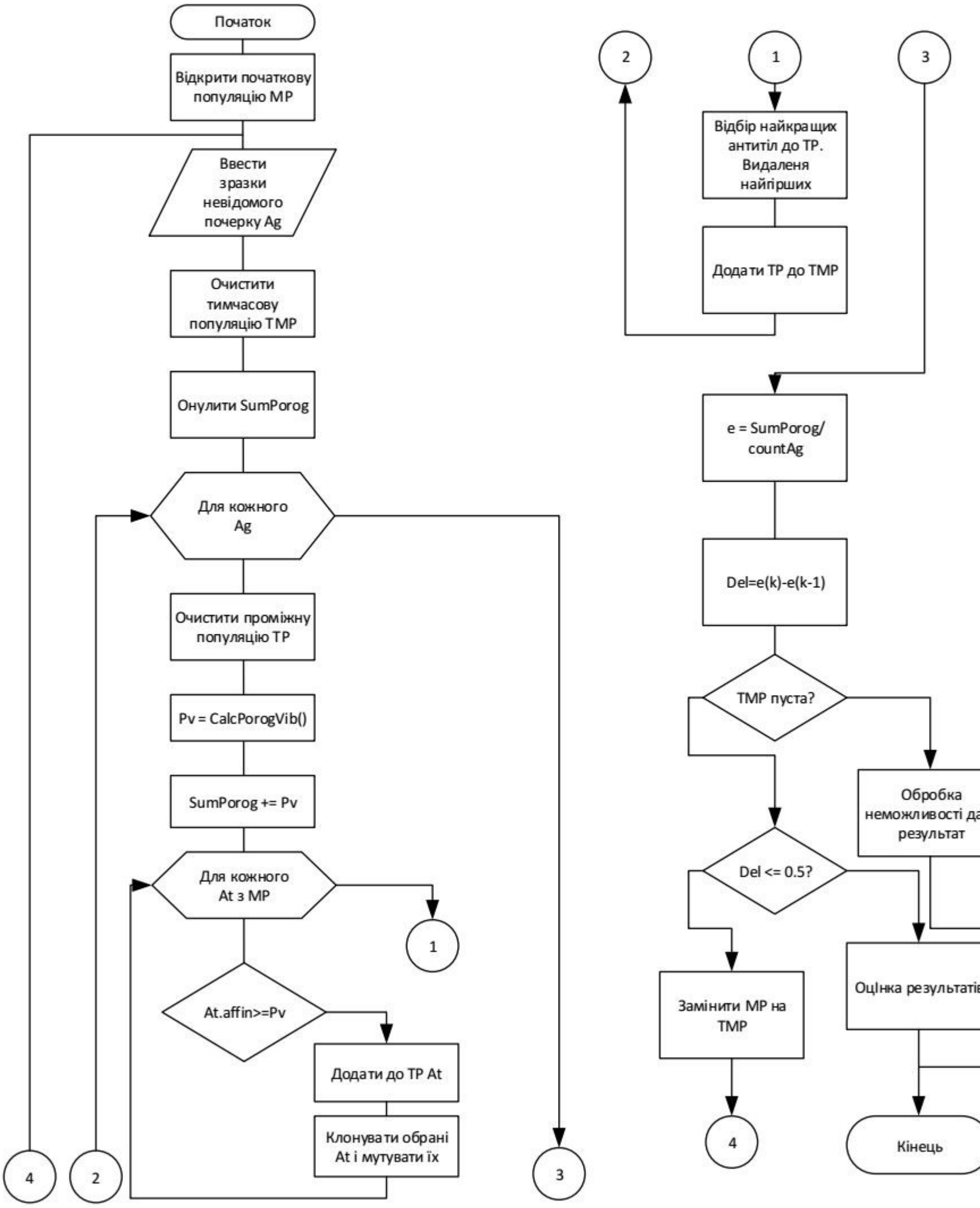


Рисунок 2.4 – Блок-схема штучної імунної мережі

Розглянемо процеси генерації, навчання, відбору та функціонування імунних детекторів на основі нейронних мереж. Генерується початкова популяція імунних детекторів, кожен з яких представляє собою штучну нейронну мережу. Уявімо нейромережевий імунний детектор у вигляді чорного ящика, який має n-входів і два виходи.

Вихідні значення детектора формуються після подачі всіх образів на нього у відповідності з наступним виразом:

$$Z_1 = \begin{cases} 1, & \text{якщо авторизований користувач} \\ 0, & \text{інше} \end{cases}$$

$$Z_2 = \begin{cases} 1, & \text{якщо не авторизований користувач} \\ 0, & \text{інше} \end{cases}$$

Для коректного функціонування нейромережеві імунні детектори (НІД) повинні пройти процес навчання. Навчальна вибірка формується з теснового користувача почерк якого класифіковано як авторизований (клас авторизованих користувачів) і набору даних інших користувачів що не проходять автентифікацію(клас клас неавторизованих користувачів). Присутність неавторизованих користувачів при навчанні дозволяє навченим імунним детекторів знаходити різницю між своїми і чужими користувачами. Очевидно, що чим більше різноманітних образів почерків присутні в навчальній вибірці, тим різноманітніше будуть імунні детектори. Бажано також мати декілька неавторизованих користувачів. Нейронна мережа навчається шляхом навчання з учителем [6], тобто ми вказуємо штучної нейронної мережі, де дані легітимних користувачів, а де - неавторизованих.

Нехай T - множина авторизованих користувачів, а F – множина неавторизованих користувачів. З них випадковим чином формується множина вхідних образів для навчання і-го детектора.

$$X_i = \begin{bmatrix} X_i^1 \\ X_i^2 \\ \dots \\ X_i^L \end{bmatrix} = \begin{bmatrix} X_{i1}^1 & X_{i2}^1 & X_{i3}^1 & \dots & X_{in}^1 \\ X_{i1}^2 & X_{i2}^2 & X_{i3}^2 & \dots & X_{in}^2 \\ & & \dots & & \\ X_{i1}^L & X_{i2}^L & X_{i3}^L & \dots & X_{in}^L \end{bmatrix}$$

де L - розмірність навчальної вибірки.

Відповідно, множина еталонних образів виглядає наступним чином:

$$l_i = \begin{bmatrix} l_i^1 \\ l_i^2 \\ \dots \\ l_i^L \end{bmatrix} = \begin{bmatrix} l_{i1}^1 & l_{i2}^1 \\ l_{i1}^2 & l_{i2}^2 \\ \dots & \\ l_{i1}^L & l_{i2}^L \end{bmatrix}$$

Еталонні вихідні значення для і-го детектора формуються так:

$$l_{i1}^k = \begin{cases} 1, & \text{якщо } X_i^k \in T \\ 0, & \text{інше} \end{cases}$$

$$l_{i2}^k = \begin{cases} 1, & \text{якщо } X_i^k \in F \\ 0, & \text{інше} \end{cases}$$

Навчання кожного детектора здійснюється з метою мінімізації сумарної квадратичної помилки детектора. Сумарна квадратична помилка і-го детектора визначається наступним чином:

$$E_i = \frac{1}{2} \sum_{k=1}^{L} \sum_{j=1}^{2} (Z_{ij}^k - l_{ij}^k)^2 \qquad (2.1)$$

Величина сумарної квадратичної помилки характеризує пристосованість детектора до виявлення неавторизованих користувачів. Чим менше її значення, тим більше пристосованість детектора. Тому величину сумарної квадратичної помилки можна використовувати для відбору кращих детекторів.

Набір навчених нейронних мереж утворює популяцію імунних детекторів, які циркулюють в комп'ютерній системі і виробляють виявлення неавторизованих користувачів. Наявність різноманітних вхідних даних для навчання і елемента випадковості у формуванні вхідних векторів дає можливість отримати велику кількість різних за своєю структурою імунних детекторів.

У процесі сканування невідомого файлу нейронна мережа ідентифікує невідомий образ, в результаті чого імунна детектор приймає рішення про

приналежність файлу до класу авторизованих користувачів або до класу неавторизованих корисувачів.

Загальний алгоритм функціонування нейромережевої імунної системи, можна представити у вигляді такої послідовності:

1. Генерація початкової популяції імунних детекторів, кожен з яких представляє собою штучну нейронну мережу з випадковими синаптическими зв'язками

2. Навчання сформованих імунних нейромережевих детекторів. Навчальна вибірка формується випадковим чином із сукупності зареєстрованих користувачів і з сукупності незареєстрованих користувачів. Еталонні вихідні значення нейронної мережі формуються відповідно (4).

3. Відбір (селекція) нейромережевих імунних детекторів на тестовій вибірці. На даній ітерації знищуються ті детектори, які виявилися нездатними до навчання, і детектори, в роботі яких спостерігаються різні недоліки (наприклад, помилкові спрацьовування). Для цього кожен детектор перевіряється на тестовій вибірці. В результаті для кожного детектора визначається значення квадратичної помилки Ei (5).

Селекція детектора проводиться таким чином:

$$D_i = \begin{cases} 0, & \text{якщо } D_i \neq 0 \\ D_i, & \text{інше} \end{cases},$$

де 0 – операція знащення детектора

4. Кожен детектор наділяється часом життя і випадковим чином вибирає користувача для сканування з сукупності користувачів, яких він не перевіряв.

5. Сканування кожним детектором обраного користувача, в результаті якого визначаються вихідні значення детекторів Z_{i1}, Z_{i2}, i=1, r.

6. Якщо i-й детектор не виявив неавторизованого користувача, тобто $Z_{i1} = 1$ і $Z_{i2} = 0$, то він вибирає наступного користувача для сканування. Якщо час життя i-го детектора закінчився, то він знищується, замість нього генерується новий детектор.

7. Якщо і-й детектор виявив неавторизованого користувача, тобто Z_{i1} = 0 і Z_{i2} = 1, то подається сигнал про виявлення неавторизованого користувача і здійснюються операції клонування і мутації відповідного детектора. Операція мутації полягає в додатковому навчанні детекторів-клонів на виявленому користувачеві. Так створюється сукупність детекторів, налаштованих на виявленого користувача.

8. Відбір клонованих детекторів, які є найбільш пристосованими до виявлення користувачів. Якщо Eij <Ei, то детектор пройшов відбір. Тут Eij - сумарна квадратична помилка j-го клону і-го детектора, яка обчислюється на неавторизованих користувачах.

9. Детектори-клони здійснюють сканування системи до тих пір, поки не відбудеться автентифікація усіх користувачів.

10. Формування детекторів імунної пам'яті. На цій ітерації визначаються нейромережеві імунні детектори, які показали найкращі результати при автентифікації користувачів. Детектори імунної пам'яті знаходяться в системі досить тривалий час і забезпечують швидку повторну автентифікацію.

Особливістю запропонованого алгоритму є те, що кожен нейромережевий імунний детектор є повністю самостійним об'єктом (автономним агентом), тобто сам вибирає собі область сканування. Для цього він отримує список користувачів, що зберігаються в просторі пам'яті, і випадковим чином вибирає користувача зі списку для його перевірки. Після перевірки одного користувача детектор переходить до наступного, також заданого випадковим чином з існуючого списку. Сканування нейромережевим імунним детектором триває до тих пір, поки детектор не може виявити автентифікацію жодного користувача, або до закінчення часу, відведеного для функціонування даного детектора [1]. Широка популяція нейромережевих імунних детекторів забезпечує своєчасне виявлення користувачів. Таким чином, дотримується принцип децентралізації системи безпеки, побудованої

на основі комбінації методів нейронних мереж і штучних імунних систем, що значно підвищує відмовостійкість і захищеність системи в цілому.

Основним завданням нейромережевого імунного детектора є поділ простору вхідних образів на два класи: авторизований клас і неавторизований клас.

Розглянемо вибір класу нейронної мережі, що лежить в основі нейромережевого імунного детектора. У процесі циркуляції НІД відбувається їх безперервна еволюція шляхом знищення старих і формування нових детекторів. Після генерації нових детекторів відбувається процес їх навчання, трудомісткість якого пропорційна розмірності навчальної вибірки. Тому, для збільшення швидкодії нейромережевої штучної імунної системи необхідно вибрати такий клас нейронної мережі, який характеризується мінімальним розміром навчальної вибірки. Розглянемо багатошаровий персептрон [6, 7], який складається з n нейронів розподільного шару, m нейронів прихованого шару і 2 нейронів вихідного шару. Загальна кількість параметрів, що настроюються (вагових коефіцієнтів і порогових значень) в такій мережі визначається наступним чином:

$$V = m \cdot (n + 3) + 2 \qquad (2.2)$$

Для гарної класифікації розмір навчальної вибірки повинен визначатися відповідно до наступного виразу:

$$L \approx V/\varepsilon, \qquad (2.3)$$

де ε – допустима точність класифікації.

Нехай n = 128, m = 10 і ε = 0,1. Тоді L≈13120. Аналогічний результат можна отримати для мультірекуррентних нейронних мереж.

Розглянемо аналогічну мережу зустрічного поширення з ідентичною кількістю нейронних елементів в шарах. У прихованому шарі будемо використовувати нейронні елементи Кохонена. У цьому випадку немає

жорстких вимог до розмірності навчальної вибірки. Досить, щоб розмір навчальної вибірки був наступним:

$$L \geq 2 \cdot m \qquad (2.4)$$

Тому виберемо в якості основи нейромережевого імунного детектора нейронну мережу зустрічного поширення.

На вхід такого детектора в режимі функціонування подаються фрагменти користувача що перевіряється, які формуються відповідно до методу ковзного вікна. Перший шар нейронних елементів є розподільним. Він розподіляє вхідні сигнали на нейронні елементи другого (прихованого) шару. Кількість нейронних елементів розподільного шару дорівнює розмірності ковзного вікна. Другий шар складається з нейронів Кохонена, які використовують конкурентний принцип навчання і функціонування відповідно до правила «переможець бере все» [6, 7]. Третій шар складається з двох лінійних нейронних елементів, які використовують лінійну функцію активації. Арбітр здійснює процедуру остаточного рішення про приналежність об'єкту до аворизованого або неавторизованого класу.

Розглянемо вибір кількості нейронів в шарі Кохонена. Нейронний шар Кохонена здійснює кластеризацию вхідного простору образів, в результаті чого утворюються кластери різних образів, кожному з яких відповідає свій нейронний елемент. Кількість нейронів шару Кохонена дорівнює m. Причому

$$m = p + r, \qquad (2.6)$$

де p - кількість перших нейронів шару Кохонена, які відповідають класу зареєстрованих користувачів; r - кількість останніх нейронів шару Кохонена, активність яких характеризує клас незареєстрованих користувачів.

При навчанні нейромережевих імунних детекторів використовується навчальна вибірка, що складається з 80% образів авторизованого класу і з 20% образів неавторизованого класу. Таким чином, співвідношення файлів в

навчальній вибірці дорівнює чотири до одного. Дане співвідношення було отримано експериментальним шляхом і показало найкращі результати.

Алгоритм формування навчальної вибірки складається з наступних кроків: 1) формується сукупність зареєстрованих і не зареєстрованих користувачів; 2) зі сформованої вибірки випадковим чином вибираються k зареєстрованих і h незареєстрованих користувачів; 3) з кожного файлу випадковим чином вибираються A фрагментів довжиною n, в результаті утворюється навчальна вибірка розмірністю $L = (k + h) \cdot A$.

Для навчання нейронів шару Кохонена використовується контрольоване конкурентне навчання [6, 7]. При такому навчанні вагові коефіцієнти нейрона переможця модифікуються тільки тоді, коли відбувається коректна класифікація вхідного образу, тобто вхідний образ відповідає заданій множині нейронів в шарі Кохонена. Так як в шарі Кохонена використовується p нейронів для зареєстрованих вхідних образів і r нейронів для не зареєстрованих вхідних образів, то коректна класифікація відбувається, якщо при подачі на вхід мережі валідного фрагмента переможцем є один з перших p нейронів шару Кохонена. Аналогічним чином коректна класифікація відбувається, якщо при подачі на вхід мережі незареєстрованого фрагмента переможцем є один з r останніх нейронів шару Кохонена. В інших випадках відбувається некоректна класифікація.

Нехай P і J характеризують відповідно авторизованого і неавторизованого користувача. Тоді правило коректної класифікації можна представити у вигляді такої імплікації:

$$P \wedge k = 1,2 \ldots p \rightarrow T, \qquad (2.7)$$
$$J \wedge k = p + 1, r \rightarrow T,$$

де T позначає коректну класифікацію.

При коректній класифікації вагові коефіцієнти нейрона-переможця посилюються:

$$\omega_{ck}(t+1) = \omega_{ck}(t) + \gamma(X_c - \omega_{ck}(t)) \qquad (2.8)$$

а при некоректній класифікації послаблюються:

$$\omega_{ck}(t+1) = \omega_{ck}(t) - \gamma(X_c - \omega_{ck}(t)) \qquad (2.9)$$

де γ – крок навчання.

Алгоритм навчання шару Кохонена складається з наступних кроків:

1. Випадкова ініціалізація вагових коефіцієнтів нейронів шару Кохонена.

2. Подається вхідний образ з навчальної вибірки на нейронну мережу і виробляються наступні обчислення:

- обчислюється Евклідова відстань між вхідним образом і ваговими векторами нейронних елементів шару Кохонена

$$D_i = |X - \omega_i| = \sqrt{(X_1 - \omega_{1i})^2 + (X_2 - \omega_{2i})^2 + (X_n - \omega_{ni})^2}, \qquad (2.10)$$

де $i=\overline{1,m}$;

- визначається нейронний елемент переможець з номером k

- проводиться модифікація вагових коефіцієнтів нейрона-переможця відповідно до (14), якщо при подачі на вхід мережі користувача що є в системі переможцем є один з перших p нейронів або при подачі на вхід мережі користувача якого немає в базі переможцем є один з r останніх нейронів мережі Кохонена. В іншому випадку проводиться модифікація вагових коефіцієнтів нейрона-переможця відповідно до (14).

Процес повторюється, починаючи з пункту 2 для всіх вхідних образів.

Навчання проводиться до бажаного ступеня узгодження між вхідними і ваговими векторами, тобто до тих пір, поки значення сумарної квадратичної помилки не стане рівною заданому порогу.

Третій шар, що складається з двох лінійних нейронних елементів, здійснює відображення кластерів, сформованих шаром Кохонена, в два класи, які характеризують авторизовані і неавторизовані вхідні образи. У загальному випадку вихідне значення j-го нейрона третього шару визначається наступним чином:

$$Y_j = \sum_{i=1}^{m} \omega_{ij} * Y_i \quad (2.11)$$

де ω_{ij} - ваговий коефіцієнт між і-м нейроном шару Кохонена і j-м нейроном лінійного шару.

Якщо нейрон-переможець в шарі Кохонена має номер k, то вихідне значення j-го нейрона третього шару дорівнює:

$$Y_j = \omega_{kj} * Y_k \quad (2.12)$$

Для відповідного відображення вхідних образів в два класи матриця вагових коефіцієнтів третього шару повинна формуватися наступним чином:

$$\omega_{kj} = \begin{cases} 1, \text{якщо } k = 1,2 \dots p \text{ і } j = 1 \\ \text{або } k = p+1 \dots r \text{ і } j = 2 \\ 0, \text{якщо } k = 1,2 \dots p \text{ і } j = 2 \\ \text{або } k = p+1 \dots r \text{ і } j = 1 \end{cases}$$

Арбітр приймає остаточне рішення про те, чи є сканований образ чужим. Для цього він обчислює кількість своїх і чужих користувачів відповідно до таких виразів:

$$\overline{Y}_1 = \sum_{k=1}^{L} Y_1^k, \quad (2.13)$$

$$\overline{Y}_2 = L - \overline{Y}_1 = \sum_{k=1}^{L} Y_2^k,$$

де L – множина образів сканованого користувача, Y_1^k – вихідне значення і-го нейрона лінійного шару при подачі на вхід мережі k-го образу.

Далі визначаються ймовірності приналежності об'єкту сканування файлу відповідно до авторизованого і неавторизованого класу:

$$P_T = \frac{\overline{Y}_1}{L} * 100\%, \quad (2.14)$$

$$P_F = 1 - P_T = \frac{\overline{Y}_2}{L} * 100\%$$

Остаточне рішення про приналежність файлу до авторизованого класу арбітр приймає в такий спосіб:

$$Z_1 = \begin{cases} 1, \text{якщо } P_T > 80\% \\ 0, \text{інше} \end{cases}$$

Відповідно, рішення про приналежність об'єкту сканування файлу до неавторизованого класу приймається у відповідності з наступним виразом:

$$Z_2 = \begin{cases} 1, \text{якщо } P_F > 20\% \\ 0, \text{інше} \end{cases}$$

Таким чином, простір вихідних значень арбітра можна представити таблицею.

Якщо вихідні значення арбітра мають нульові значення, то сканований файл відправляється на додаткову перевірку іншому нейромережевому імунному детектору.

В загальному принцип функціонування мережі в режимі сканування можна привести до наступної послідовності кроків:

1. Встановлюються такі початкові значення:

$$\overline{Y}_1(k-1) = 0, \overline{Y}_2(k-1) = 0 \qquad (2.15)$$

2. За методом ковзного вікна послідовно подаються вхідні образи (k = 1, L) з сканованого потоку даних на нейронну мережу і для кожного вхідного образу виробляються наступні обчислення:

- визначається Евклідова відстань між вхідним образом і ваговими векторами нейронів шару Кохонена (15);

- визначається нейронний елемент-переможець з номером k (16);

- обчислюються вихідні значення лінійних нейронних елементів третього шару (18);

- визначається кількість авторизованих і неавторизованих фрагментів сканованого файлу:

$$\overline{Y}_1(k) = \overline{Y}_1(k-1) + Y_1^k, \qquad (2.16)$$

$$\overline{Y}_2(k) = \overline{Y}_2(k-1) + Y_2^k.$$

3. Обчислюються імовірності належності сканованого файлу відповідно до авторизованого і неавторизованого класу (22) і (23) відповідно.

4. На підставі обчислень ймовірностей приймається рішення про належність об'єкту сканування файлу до одного з класів, відповідно до (24) і (25).

5. Якщо Z1 = 0 і Z2 = 0, то призначається інший нейромережевий імунний детектор для повторної перевірки файлу.

2.2 Висновки

Розглянуто біологічно орієнтовані алгоритми: нейронні та імунні системи. Проаналізовано принципи їх роботи, а також інтерпретовано до вимог роботи. Проаналізувавши переваги та недоліки цих алгоритмів розроблено структуру, що дозволяє в повній мірі використати всі необхідні параметри користувача для виконання задачі автентифікації. Структура представлена у вигляді обєднання нейромережі персептронного типу для аналізу та обробки даних користувача, що відображають підсвідомі характеристики клавіатурного почерку та нейроподібної мережі імунного типу для аналізу та обробки даних, що відображають свідомі впливи, якими при використанні альтернативних методів обробки нехтують для забезпечення стабільної роботи системи автентифікації.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСОБУ

Засіб автентифікації за клавіатурним почерком представлений у вигляді програми, що складається з двох частин, клієнтської та серверної.

3.1 Клієнтська частина засобу

Клієнтська частина засобу представляє собою програму, що дозволяє збирати дані про користувача, а саме час натискання та відпускання кожної клавіші, а також номер цих клавіш на клавіатурі, що дозволяє не тільки збирати дані про часові показники користувача, але також дані самих клавіш, що в свою чергу підвищує надійність системи, за рахунок додання до вектору вхідних даних додаткової еталонної характеристики.

Програма збирає дані про користувача за допомогою механізму хуків.

```
public Logger()
{
    hookKeyboard();
    reset();
}
private void hookKeyboard()
    KeyboardHookProcedure = new HookProc(KeyboardHookProc);
    hKeyboardHook = SetWindowsHookEx(
        WH_KEYBOARD_LL,
        KeyboardHookProcedure,
Marshal.GetHINSTANCE(Assembly.GetExecutingAssembly().GetModules()[0]),
        0);
```

Перехоплені натиснення заносяться до log файлу.

```
private void logEvent (int keyCode, string eventType)
{
    string timeStamp = (DateTime.UtcNow.Ticks/
TimeSpan.TicksPerMillisecond).ToString();
    string logLine = string.Format("{0} {1} {2}", timeStamp, keyCode,
eventType);
    _log.WriteLine(logLine);
}
```

Процес натискання клавіші логується подією KEY_DOWN, а процес відпускання клавіші – KEY_UP.

```
                if (wParam == WM_KEYDOWN || wParam == WM_SYSKEYDOWN)
                {
                    int keyCode = keyboardHookStruct.vkCode;
                    logEvent(keyboardHookStruct.vkCode, "KEY_DOWN");
                }

                if (wParam == WM_KEYUP || wParam == WM_SYSKEYUP)
                {
                    logEvent(keyboardHookStruct.vkCode, "KEY_UP");
                }
            }
```

Головне вікно програми зображено на рис. 3.1.

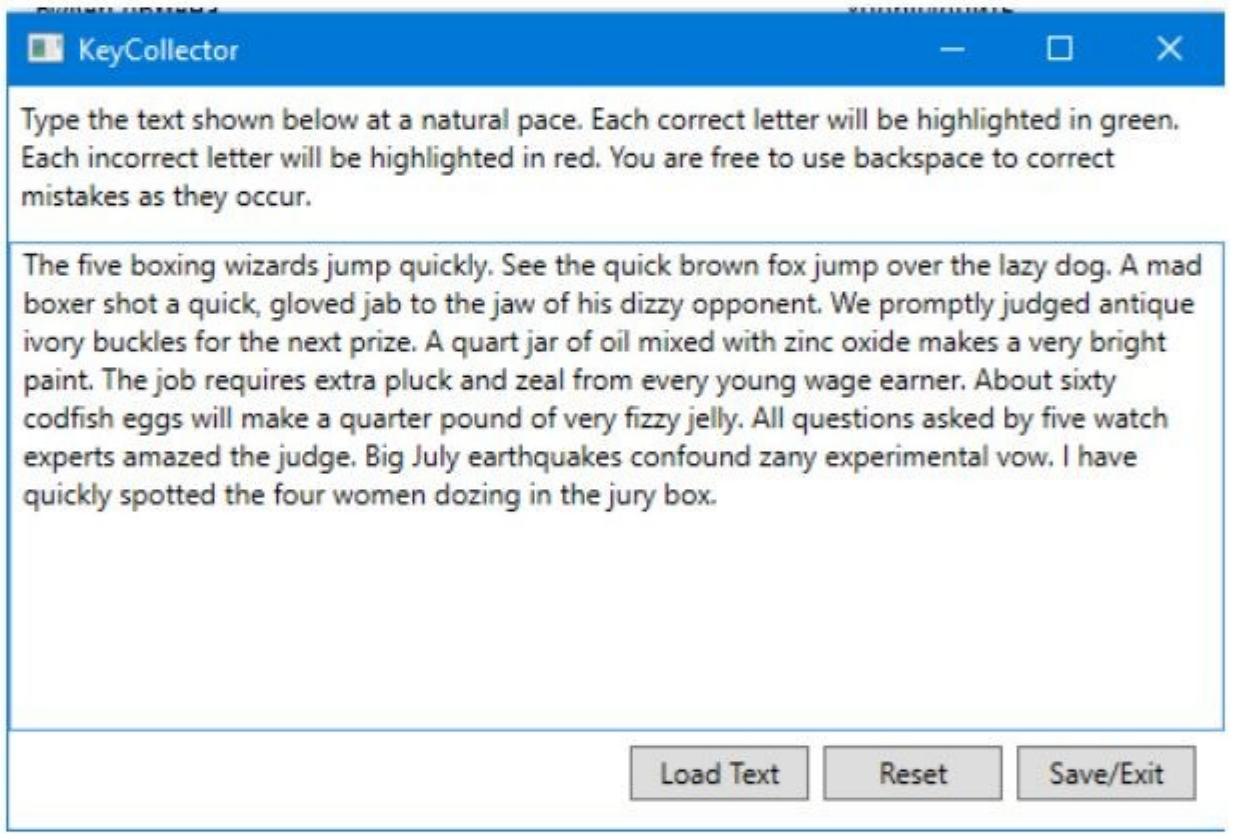


Рисунок 3.1 – Головне вікно програми

За замовчуванням в головному вікні програми відображається текст на 600 символів, проте за допомогою кнопки Load Text можна выдкрити будь-який текстовий файл.

Робота з текстом виглядає як в програмах для навчання друку на клавіатурі: правильні натискання маркуються зеленим, не правильні червоним.

Приклад роботи з текстом зображений на рис. 3.2

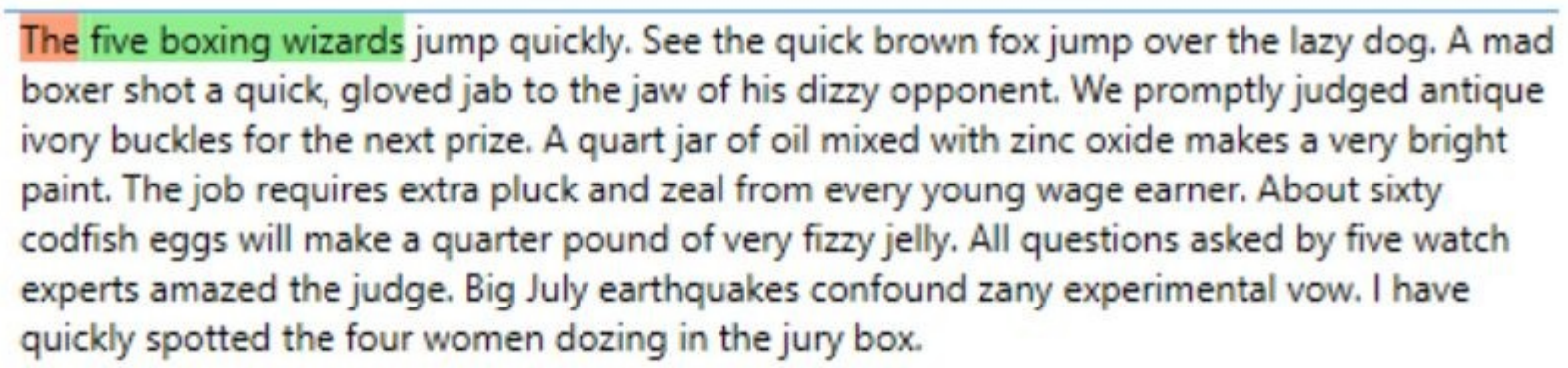


Рисунок 3.2 – Робота з текстом

По закінченню збору даних потрібно натиснути кнопку Save/Exit для збереження log файлу. Файл зберігається в тій самій теці звідки було завантажено текстовий файл, log файл зберігається у текстовому вигляді:

```
63664252324699 164 KEY_DOWN
63664252324771 160 KEY_DOWN
63664252324836 164 KEY_UP
63664252324838 162 KEY_UP
63664252324852 160 KEY_UP
63664252326917 160 KEY_DOWN
63664252327115 84 KEY_DOWN
63664252327170 160 KEY_UP
63664252327223 84 KEY_UP
63664252328006 72 KEY_DOWN
63664252328081 72 KEY_UP
63664252328133 69 KEY_DOWN
63664252328194 69 KEY_UP
63664252328271 32 KEY_DOWN
```

Дані файлу представлені у вигляді часу натискання/відпускання клавіші, ASCII коду клавіші а також події, що відбулась.

3.2 Серверна частина засобу

Серверна частина засобу відповідає за порівняння наданих даних з еталонними і відправляє користувачеві результат автентифікації.. Серверна частина була змодельована в середовищі розробки PyCharm. Проміжні обчислення були обраховані в Microsoft Exel, а не в клієнтському додатку

через можливість завантаження даних в Python напряму з Exel. Вигляд проміжних даних зображений на рис. 3.3.

492	1058	1509	1315	340	1590	240	851	135	847
324	892	1180	1341	1371	1027	1492	1364	34	872
587	338	1150	254	508	645	1064	253	84	1515
490	749	1492	87	738	535	748	729	335	468
204	340	189	644	1265	341	743	453	1369	658
644	293	793	463	1039	619	832	1481	441	147
445	1057	243	826	1321	359	472	731	254	830
925	40	1005	661	208	251	930	1173	1433	764
290	1448	484	1019	589	645	714	1294	1536	408
199	801	595	1079	408	69	1000	1104	324	881

Рисунок 3.3 – Проміжні дані

Для моделювання нейронної мережі було обрано мову прогрограмування Python а саме бібліотеку pybrain. PyBrain оперує мережевими структурами, які можуть бути використані для побудови практично всіх підтримуваних бібліотекою складних алгоритмів.

PyBrian являє собою модульну бібліотеку призначену для реалізації різних алгоритмів машинного навчання на мові Python. Основною його метою є надання дослідникуу гнучких, простих у використанні, але в той же час потужних інструментів для реалізації завдань з області машинного навчання, тестування і порівняння ефективності різних алгоритмів.

Архітектура мережі представляє собою Pattern Recognition тобто мережа яка відповідає за розпізнавання образів та класифікацію і на вихідному шарі містить softmax функцію, що «стискує» K-вимірний вектор z із довільним значеннями компонент до K-вимірного вектора σ (z) з дійсними значеннями компонентів в області від 0 до 1. Виходи softmax роблять її придатними для ймовірнісної інтерпретації.

Імпортовані дані ділять на дві частини: дані для формування вхідного вектора inputs та для перевірки відповідей нейронної мережі targets.

Спочатку потрібно навчити кожну нейросистему. Для навчання буде створено декілька нейродетекторів з різною архітектурою щоб забезпечити найефективніше навчання

```
model = Sequential()
model.add(Dense(25, input_dim=20, activation='sigmoid'))
model.add(Dense(34, activation='sigmoid'))
model.add(Dense(45, activation='sigmoid'))
model.add(Dense(71, activation='sigmoid'))
model.add(Dense(46, activation='sigmoid'))
```

У PyBrain використана концепція тренерів (trainers) для навчання мереж з учителем. Тренер отримує екземпляр мережі і екземпляр набору зразків і потім навчає мережу по отриманому набору. Класичний приклад це зворотне поширення помилки (backpropagation). Для спрощення реалізації цього підхід в PyBrain існує клас BackpropTrainer.

Навчальний набір зразків (targets) і цільова мережа (model) вже створені, тепер вони будуть об'єднані.

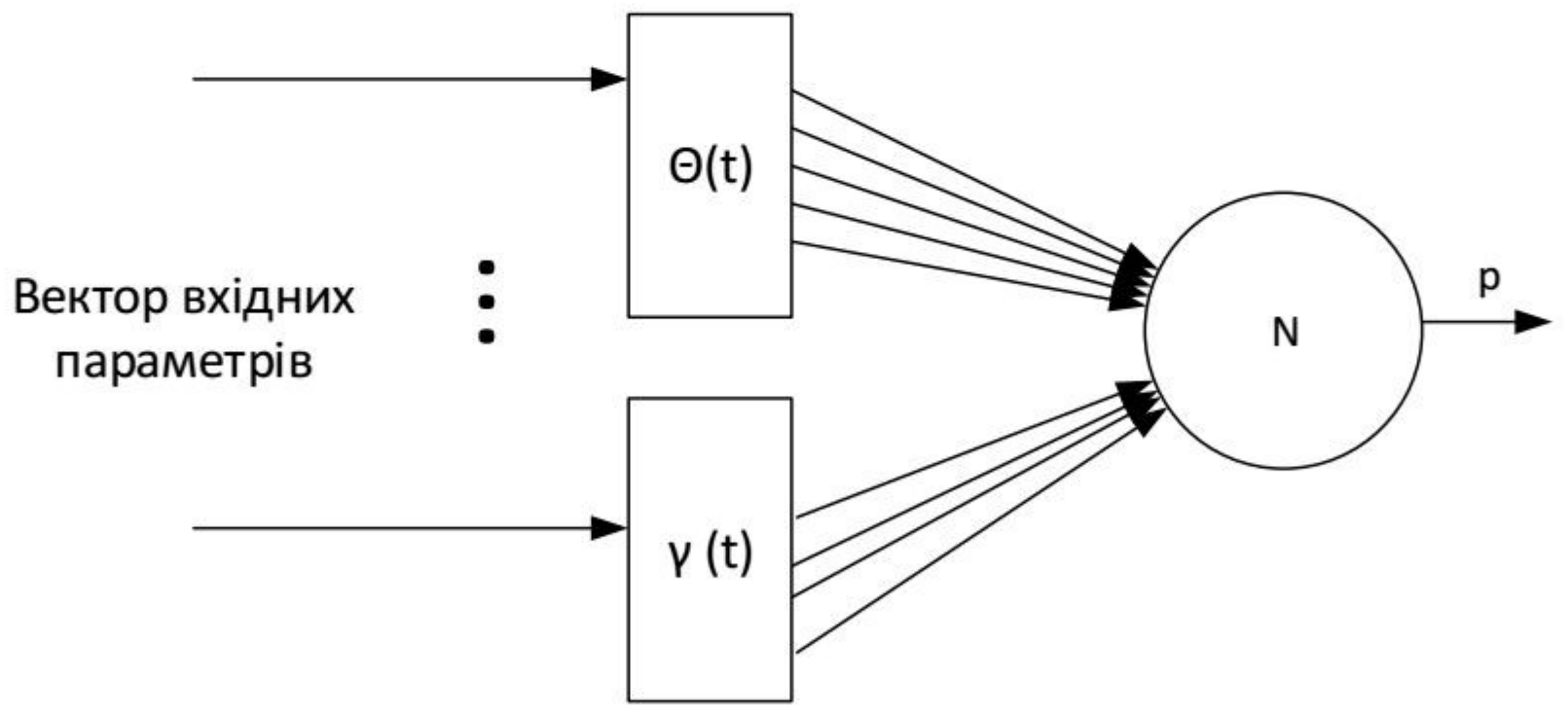
```
trainer = BackpropTrainer(model, targets)
```

Тренер отримав посилання на структуру мережі і може її тренувати.

```
trainer.train()
```

3.3 Аналіз отриманих результатів

В загальному випадку для рішення задачі автентифікації користувача за клавіатурним почерком використовується наступна структура нейромережі:

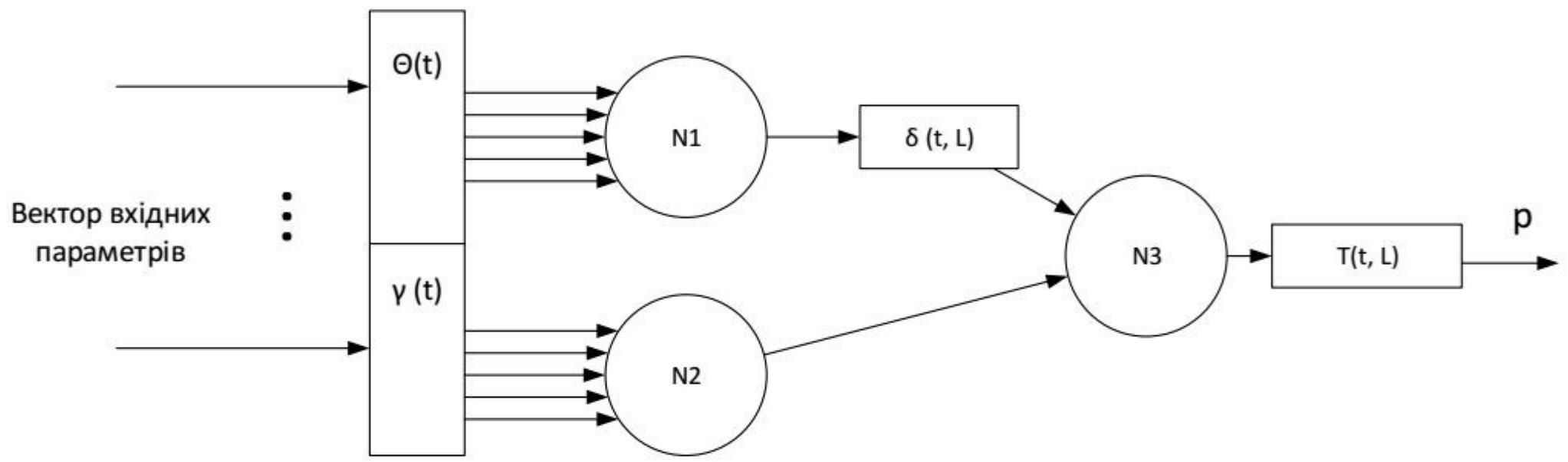


На вхід нейромережі подається вектор вхідних даних який представляє собою часові характеристики клавіатурного почерку користувача. Нейромережа представляє собою одношаровий персептрон. На виході

нейромережі значення ймовірності, що характеризує з якою ймовірністю вхідні параметри відповідають параметрам зареєстрованого користувача.

Така структура добре себе зарекомендувала в ситуаціях коли довжина паролю не змінюється і не перевищує 20 символів, в іншому ж випадку кількість ітерацій які проходить нейромережа буде збільшуватись і враховуючи кількість параметрів, що входять до вектору вхідних даних, нейромережа буде довго навчатись, а порядок помилки буде збільшуватись.

Тому необхідно змінити структуру нейромережі на більш оптимальну:



В даній структурі вектор вхідних даних подається на блок в якому параметри розбиваються на дві компоненти, що описують підсвідомі і свідомі процеси мислення відповідно. Получені вектори подаються на вхід блоків N_1 і N_2, що представлені штучними імунними мережами. Нейромережа N_3 виконує остаточну автентифікацію користувача і подає на вихід значення ймовірності, що характеризує з якою ймовірністю вхідні параметри відповідають параметрам зареєстрованого користувача. Отриманий результат корелюється за допомогою функції T(t, L), в залежності від досвіду роботи з клавіатурою.

Результати роботи програми наведені в таблиці 3.1

Користувач	Ідентифікатор користувача	Результат роботи нейромережі
User A	1	Was identyfied as User A
User B	2	Was identyfied as User B
User C	3	Was identyfied as User C
User D	4	Was identyfied as User D
User E	5	Was identyfied as User E
User F	6	Was identyfied as User F
User G	7	Wasn't identified
User J	8	Was identyfied as User J
Unknown user A	0	Wasn't identified
Unknown user B	0	Wasn't identified

Проведені тести показали, що програма працює коректно на 90% так як з 10 тестів 9 закінчились успішно, також програма виконує завдання для яких була розроблена, а саме автентифікацію за клавіатурним почерком.

4 ЕКОНОМІЧНИЙ РОЗДІЛ

4.1 Аналіз комерційного потенціалу розробки (технологічний аудит розробки)методу автентифікації за клавіатурним почерком.

4.1.1 Визначення рівня комерційного потенціалу розробки методу автентифікації за клавіатурним почерком

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу методу методу автентифікації за клавіатурним почерком. В результаті оцінювання можна буде зробити висновок щодо напрямів (особливостей) організації подальшого її впровадження з врахуванням встановленого рейтингу.

Для проведення технологічного аудиту залучимо 3-х незалежних експертів. У нашому випадку такими експертами будуть керівник магістерської роботи та провідні викладачі випускової та споріднених кафедр.

Оцінювання комерційного потенціалу розробки методу методу автентифікації за клавіатурним почерком будемо здійснювати за 12-ю критеріями згідно рекомендацій.

Результати оцінювання комерційного потенціалу розробки методу методу автентифікації за клавіатурним почерком заносимо до таблиці 4.1.

Таблиця 4.1. - Результати оцінювання комерційного успіху розробки методу автентифікації за клавіатурним почерком

Критерії	Експерти		
	д. т. н., проф., Лужецький В.А.	к.т.н., проф. Кондратенко Н.Р.	к.т.н., доцент Войтович О.П.
	Бали, виставлені експертами		
1	2	2	3
2	3	3	2
3	4	4	3
4	4	2	3

5	4	3	3
6	4	4	3
7	3	3	2
8	3	3	3
9	3	4	4
10	2	3	3
11	3	3	3
12	2	3	3
Сума балів	37	37	35
Середньоарифметична сума балів, СБ	36		

За даними таблиці 4.1 робимо висновок щодо рівня комерційного потенціалу розробки методу автентифікації за клавіатурним почерком. При цьому користуємося рекомендаціями, наведеними в таблиці 4.2.

Таблиця 4.2 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 50	Високий

Таким чином, робимо висновок, щодо рівня комерційного потенціалу нашої розробки методу автентифікації за клавіатурним почерком вище середнього.

4.1.2 Визначення рівня якості розробки методу автентифікації за клавіатурним почерком

Оцінювання рівня якості розробки методу автентифікації за клавіатурним почерком проводиться з метою порівняльного аналізу і визначення найбільш ефективного, з технічної точки зору, варіанта інженерного рішення.

Рівень якості – це кількісна характеристика міри придатності певного виду продукції для задоволення конкретного попиту на неї при порівнянні з відповідними базовими показниками за фіксованих умов споживання.

Абсолютний рівень якості розробки адаптивного методу для автентифікації користувачів мобільних пристроїв знаходимо обчисленням вибраних для її вимірювання показників, не порівнюючи їх із відповідними показниками аналогічних виробів. Для цього необхідно визначити зміст основних функцій, які повинні реалізовувати розробка, вимоги замовника до неї, а також умови, які характеризують експлуатацію, визначають основні параметри, які будуть використані для розрахунку коефіцієнта технічного рівня виробу. Система параметрів, прийнята до розрахунків, повинна достатньо повно характеризувати споживчі властивості інноваційного товару (його призначення, надійність, економічне використання ресурсів, стандартизація тощо).

Далі визначаємо величину параметрів якості в балах та встановлюємо граничні його значення (кращі, гірші, середні). Всі ці дані для кожного параметра заносимо в табл. 4.3.

Таблиця 4.3 – Основні параметри методу автентифікації за клавіатурним почерком

Параметри	Абсолютне значення параметра			Коефіцієнт вагомості параметра
	Краще +5…+4	Середнє +3	Гірше +1…+2	
Низька відноснапохибка			2	0,2
Безпека			2	0,5
Ресурсозатратність		3		0,2
Варіативність		3		0,1

Із врахуванням коефіцієнтів вагомості відповідних параметрів можна визначити абсолютний рівень якості інноваційного рішення за формулою:

$$К_{я.а.} = \sum_{i=1}^{n} Рні \cdot аі, \quad (4.1)$$

де Рні – числове значення і-го параметра інноваційного рішення, n – кількість параметрів інноваційного рішення, що прийняті для оцінювання, аі – коефіцієнт вагомості відповідного параметра (сума коефіцієнтів вагомості всіх параметрів повинна дорівнювати 1).

Отже, абсолютний рівень якості методу та засобу завадостійкого розподілу секрету становитиме – 2,3 бали.

Одночасно визначаємо відносний рівень якості методу автентифікації за клавіатурним почерком, що виробляється (проектується), порівнюючи її показники з абсолютними показниками якості найліпших вітчизняних та зарубіжних аналогів (товарів-конкурентів) (табл. 4.4).

Таблиця 4.4 – Основні параметри методу автентифікації за клавіатурним почерком та товару-конкурента

Параметри	Варіанти		Відносний показник якості	Коефіцієнт вагомості параметра
	Базовий (конкурент)	Новий		
Низька відноснапохибка	2	1	2	0,2
Безпека	2	3	1,5	0,5
Ресурсозатратність	5	4	0,8	0,2
Варіативність	3	2	0,7	0,1

Відносний рівень якості методу та засобу завадостійкого розподілу секрету визначаємо за формулою:

$$К_{я.в.} = \sum_{i=1}^{n} qi \cdot ai, \quad (4.2)$$

За розрахунками відносний рівень якості методу автентифікації за клавіатурним почерком становитиме – 1,78. Це означає, що наша розробка краща за якістю на 78% від товару-аналога.

4.1.3 Визначення конкурентоспроможності розробки методу автентифікації за клавіатурним почерком

У найширшому розумінні конкурентоспроможність товару – це можливість його успішного продажу на певному ринку і в певний проміжок часу. Водночас конкурентоспроможною можна вважати лише однорідну продукцію з технічними параметрами і техніко-економічними показниками, що ідентичні аналогічним показникам уже проданого товару. Для того, щоб високоякісний товар був одночасно і конкурентоспроможним, він має

відповідати критеріям оцінювання споживачів конкретного ринку в конкретний час.

Дані для розрахунку загального показника конкурентоспроможності розробки необхідно занести до таблиці 4.5.

Таблиця 4.5 – Нормативні, технічні та економічні параметри методу автентифікації за клавіатурним почерком і товару-конкурента

Параметри	Варіанти		Відносний показник якості	Коефіцієнт вагомості параметра
	Базовий (конкурент)	Новий		
Низька відносна похибка	2	1	2	0,2
Безпека	2	3	1,5	0,5
Ресурсозатратність	5	4	0,8	0,2
Варіативність	3	2	0,7	0,1
Ціна за продукт, тис. грн.	3000	2000	0,7	-

Загальний показник конкурентоспроможності розробки (К) з урахуванням вищезазначених груп показників визначаємо за формулою:

$$К = \frac{Іт.п.}{Іе.п.} = \frac{1,78}{0,7} = 2,5, \quad (4.3)$$

де Іт.п. – індекс технічних параметрів (відносний рівень якості інноваційного рішення); Іе.п. – індекс економічних параметрів.

$$Іе.п. = \frac{РНеі}{РБеі} = \frac{2000}{3000} = 0,7, \quad (4.4)$$

де РНеі, РБеі – економічні параметри (ціна придбання та споживання товару) відповідно нового та базового товарів.

Згідно розрахунків загальний показник конкурентоспроможності –2,5. Це означає, що наша розробка методу автентифікації за клавіатурним почерком більш конкурентна майже в2,5 рази від товару-аналога.

4.2. Прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи

4.2.1 Розрахунок витрат, що стосуються виконавців розробки методу автентифікації за клавіатурним почерком

Основна заробітна плата кожного із розробників (дослідників) Зо, якщо вони працюють в наукових установах бюджетної сфери:

$$Зо = \frac{М}{Тр} \cdot t, \qquad (4.5)$$

де М – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.

У 2019 році величини окладів (разом з встановленими доплатами і надбавками) рекомендується брати в межах (5000…10000) грн. за місяць; Тр – число робочих днів в місяці; приблизно Тр = (21…23) дні; t – число робочих днів роботи розробника (дослідника).

Зроблені розрахунки зводимо до таблиці 4.6.

Таблиця 4.6 – Заробітна плата розробників

Посада	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	8000	381	5	1905
Інженер-програміст	4500	214	5	1070
Всього:				2975

Основна заробітна плата робітників Зр, якщо вони беруть участь у виконанні даного етапу роботи і виконують роботи за робочими професіями у

випадку, коли вони працюють в наукових установах бюджетної сфери, розраховується за формулою:

$$Зр = \sum_{i=1}^{n} ti \cdot Ci, \quad (4.6)$$

де ti – норма часу (трудомісткість) на виконання конкретної роботи, годин; n – число робіт по видах та розрядах; Сі – погодинна тарифна ставка робітника відповідного розряду, який виконує дану роботу. Сі визначається за формулою:

$$Ci = \frac{Мм \cdot Ki}{Тр \cdot Тзм}, \quad (4.7)$$

де Мм – розмір мінімальної заробітної плати за місяць, грн.; в 2019 році мінімальна заробітна плата становить – 4173 грн., Кі – тарифний коефіцієнт робітника відповідного розряду, Тр – число робочих днів в місяці; приблизно Тр = 21…23 дні; Тзм – тривалість зміни, зазвичай Тзм = 8 годин.

Таблиця 4.7 – Заробітна плата робітників

Найменування робіт	Трудомісткість, н-год.	Розряд роботи	Погодинна тарифна ставка	Тариф. коеф.	Величина, грн.
Програмувальні	7	5	34	1,36	170
Всього					170

Додаткова заробітна плата Зд всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10…12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Зд = 0{,}1 \cdot (Зр + Зо) = 0{,}1 \cdot (2975 + 170) = 314{,}5 \text{ грн.} \quad (4.8)$$

Нарахування на заробітну плату Нзп розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

де Зо – основна заробітна плата розробників, грн.; Зр – основна заробітна плата робітників, грн.; Зд – додаткова заробітна плата всіх розробників та робітників, грн.; β – ставка єдиного внеску на загальнообов'язкове державне

соціальне страхування, % (приймаємо для 1-го класу професійності ризику 22%).

$$Нзп = 0{,}22 \cdot (Зр + Зо + Зд) = 0{,}22 \cdot (2975 + 170 + 314{,}5) = 761 \text{ грн.} \quad (4.8)$$

Амортизація обладнання, комп'ютерів та приміщень А, які використовувались під час (чи для) виконання даного етапу роботи.

Дані відрахування розраховують по кожному виду обладнання, приміщенням тощо.

У спрощеному вигляді амортизаційні відрахування А в цілому бути розраховані за формулою:

$$А = \frac{Ц \cdot На}{100} \cdot \frac{Т}{12}, \quad (4.9)$$

де Ц – загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн.; На – річна норма амортизаційних відрахувань. Для нашого випадку можна прийняти, що На = (10...25)%; Т – термін, використання обладнання, приміщень тощо, місяці.

Таблиця 4.8 - Амортизаційні відрахування

Найменування	Ціна, грн.	Норма амортизації, %	Термін використання, м.	Сума амортизації
ПК	7000	20	2	233
Інше обладнання	3000	10	1	25
Всього	258			

Витрати на силову електроенергію Ве, якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою:

$$Ве = В \cdot П \cdot Ф \cdot Кп, \text{ грн}$$

В – вартість 1 кВт-год. електроенергії, в 2019 р. В ≈ 8,45 грн./кВт; П – установлена потужність обладнання, кВт; Ф – фактична кількість годин роботи обладнання, годин, Кп – коефіцієнт використання потужності; Кп< 1.

Потужність обладнання складає –0,5 кВт.

Кількість годин роботи складає – 100годин.

Коефіцієнт викор. потужності -0,9.

Ве= В · П · Ф · Кп = 380 грн.

Інші витрати Він охоплюють: витрати на управління організацією, оплата службових відряджень, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, охорону праці тощо.

Інші витрати Ів можна прийняти як (100…300)% від суми основної заробітної плати розробників та робітників, які були виконували дану роботу, тобто:

$$Ів = 1{,}5 \cdot (Зо + Зр) = 1{,}5 \cdot (2975 + 170) = 4718 \text{ грн.} \qquad (4.10)$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини (розділу, етапу) роботи – В.

$$В = 2975+170+314+761+380+4718=9577 \text{ грн.}$$

4.2.2 Розрахунок собівартості розробки методу автентифікації за клавіатурним почерком

Витрати на силову електроенергію Ве, якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою:

$$Ве = В \cdot П \cdot Ф \cdot Кп, \text{ грн}$$

В – вартість 1 кВт-год. електроенергії, в 2019 р. В ≈ 8,45 грн./кВт; П – установлена потужність обладнання, кВт; Ф – фактична кількість годин роботи обладнання, годин, Кп – коефіцієнт використання потужності; Кп< 1.

Потужність обладнання складає – 0,5 кВт.

Кількість годин роботи складає – 100 годин.

Коефіцієнт викор. потужності -0,9.

Ве= В · П · Ф · Кп =380 грн.

Основна заробітна плата робітників Зр, якщо вони беруть участь у виконанні даного етапу роботи і виконують роботи за робочими професіями у випадку, коли вони працюють в наукових установах бюджетної сфери, розраховується за формулою:

$$Зр = \sum_{i=1}^{n} ti \cdot Ci, \quad (4.11)$$

де ti – норма часу (трудомісткість) на виконання конкретної роботи, годин; n – число робіт по видах та розрядах; Сі – погодинна тарифна ставка робітника відповідного розряду, який виконує дану роботу. Сі визначається за формулою:

$$Ci = \frac{Мм \cdot Ki}{Тр \cdot Тзм}, \quad (4.12)$$

де Мм – розмір мінімальної заробітної плати за місяць, грн.; в 2019 році мінімальна заробітна плата становить – 4173 грн., Кі – тарифний коефіцієнт робітника відповідного розряду, Тр – число робочих днів в місяці; приблизно Тр = 21…23 дні; Тзм – тривалість зміни, зазвичай Тзм = 8 годин.

Таблиця 4.9 – Заробітна плата робітників

Найменування робіт	Трудомісткість, н-год.	Розряд роботи	Погодинна тарифна ставка	Тариф. коеф.	Величина, грн.
Програмувальні	7	5	34	1,36	170
Всього					170

Додаткова заробітна плата Зд всіх робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10…12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Зд = 0{,}1 \cdot (Зо) = 0{,}1 \cdot (170) = 17 \text{ грн.} \quad (4.13)$$

Нарахування на заробітну плату Нзп розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

де Зо – основна заробітна плата розробників, грн.; Зр – основна заробітна плата робітників, грн.; Зд – додаткова заробітна плата всіх розробників та робітників, грн.; β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, % (приймаємо для 1-го класу професійності ризику 22%).

$$Нзп = 0{,}22 \cdot (Зо + Зд) = 0{,}22 \cdot (170 + 17) = 41 \text{ грн.} \quad (4.14)$$

«Загальновиробничі витрати» належать витрати: пов'язані з управлінням виробництвом (утримання працівників апарату управління виробництвом, оплата службових відряджень персоналу цехів, витрати на інформаційне забезпечення управління тощо); на повне відновлення та капітальний ремонт основних фондів загальновиробничого призначення; витрати некапітального характеру, пов'язані з удосконаленням технологій та організацією виробництва, поліпшенням якості продукції; на утримання, обслуговування, поточний ремонт виробничих приміщень; на контроль за виробничими процесами та кістю продукції.

Крім того, загальновиробничі витрати з розрахунку на одиницю продукції можна розрахувати за нормативами відносно до основної заробітної плати основних робітників, які виготовляють продукцію:

$$ЗВВ = Нв \cdot Зо, \quad (4.15)$$

Норматив загальновиробничих витрат для програмних продуктів становить 230-270%.

$$ЗВВ = 2{,}5 \cdot 170 = 425 \text{ грн},$$

Сума попередніх витрат утворює виробничу собівартість розробки:

$$S_В = 1033 \text{ грн.}$$

4.3 Розрахунок ціни та чистого прибутку від реалізації розробки методу автентифікації за клавіатурним почерком

Ціна – це грошовий вираз вартості товару (продукції, послуги). Вона завжди коливається навколо ціни виробництва (перетвореної форми вартості одиниці товару, що дорівнює сумі витрат виробництва й середнього прибутку) та відображає рівень суспільне необхідних витрат праці.

Виходячи з того, що розробки, як правило, приймаються та впроваджуються за завданням замовника, або коли результатом розробки є продукція, що підлягає державному регулюванню, то нижню межу ціни реалізації розробки можна розрахувати за формулою:

$$Ц = S_В \cdot \left(1 + \frac{Р}{100}\right) \cdot \left(1 + \frac{\omega}{100}\right), \quad (4.16)$$

де Sв – виробнича собівартість інноваційного рішення, грн.; Р – норматив рентабельності узгоджений із замовником або встановлений державою, (Р=30…60%); w – ставка податку на додану вартість, % (в 2019 році w=20%).

$$Ц = 1033 \cdot \left(1 + \frac{60}{100}\right) \cdot \left(1 + \frac{20}{100}\right) = 1983 \text{ грн.}$$

Чистий прибуток від реалізації розробки можна розрахувати за формулою:

$$П = \left(Ц - \frac{(Ц - МР) \cdot f}{100} - S_В - \frac{q \cdot S_В}{100}\right) \cdot \left(1 - \frac{h}{100}\right) \cdot РП, \quad (4.17)$$

де Ц – ціна розробки, грн.; МР – вартість матеріальних та інших ресурсів, що були придбані виробником для виготовлення розробки (МР=(0,1…0,2) Цр), грн.; f – зустрічна ставка податку на додану вартість, %; Sв – виробнича собівартість розробки, грн.; q – норматив, який визначає величину адміністративних витрат, витрат на збут та інші операційні витрати, % (рекомендовано q=5…10%); h – ставка податку на прибуток, %, РП – прогнозований попит продажів:

$$П = 8360 \text{ грн.}$$

4.4 Розрахунок терміну окупності коштів, вкладених в наукову розробку методу автентифікації за клавіатурним почерком

Термін окупності вкладених у реалізацію наукового проекту інвестицій Ток можна розрахувати за формулою:

$$Ток = \frac{В}{П} = \frac{9577}{8360} = 1{,}15 \text{ роки.} \quad (4.18)$$

Оскільки Ток < 3 років, то фінансування даної наукової розробки методу автентифікації за клавіатурним почерком доцільне.

Таким чином в результаті всіх проведених вище обрахунків можна говорити про економічну доцільність розробки математичної моделі та засобу автенитифікації за клавіатурним почерком з використанням штучних імунних систем

ВИСНОВКИ

Результатом виконання магістрської кваліфікаційної роботи є засіб для автентифікації користувачів на основі нейромережевого підходу. Було проведено аналіз літературних джерел серед яких: дослідження клавітурного почерку користувачів та методи обробки отриманих даних, аналіз нейромережевого підходу обробки та його модифікації у вигляді імунних систем також була опрацьована література щодо проектування нейронних мереж.

В результаті аналізу було наведено класифікацію відомих методів автентифікації, основні їх переваги і недоліки, розглянуті різні біометричні методи автентифікації і в часності за клавіатурним почерком.

При виконанні магістрської кваліфікаційної роботи для створення нейронної мережі було обрано мову програмування Python. Було проведено підготовку навчальної мережі, проаналізовані ознаки які дають змогу визначити ідентфікатор користувача по його характеристиках клавіатурного почерку відповідно до еталонних даних . Проведено декілька моделювань з різними параметрами мережі: тип архітектури, кількості нейронів та шарів, різні функції активації. В результаті тестування засобу на разній довжині ключової фрази при різних налаштуваннях мережі отримано результати автентифікації в межах 90%, відповідно результати похибки є дуже малими.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. SafeNet, Inc. 2004 annual password survey results, 2005. URL: www.safenetinc.com/news/view.asp?newsID= 239.
2. K.-P. L. Vu, R. W. Proctor, A. Bhargav-Spantzel, B.-L. B. Tai, J. Cook, and E. E. Schultz. Improving password security and memorability to protect persona and organizational information. Int. J. Hum.-Comput. Stud., 65(8): 2007. -C 744–757
3. Шаньгин В. Ф. Защита компьютерной информа-ции. Эффективные методы и средства. – М.: ДМК Пресс, 2008. – 544 С.
4. Ричард Э. Смит. Автентификация: от паролей до открытых ключей./ Р.Э. Смит. – М.: «Вильямс», 2002. – 432 с.
5. Захист інформації з використанням біометричних систем [Електронний ресурс]. – Режим доступу: http://www.rusnauka.com/35_OINBG_2010/ Informatica/76206.doc.htm - Суть і значення інформаційних біометричних систем.
6. Пропуск котрий всегда з собой [Електронний ресурс]. – Режим доступу: http://compress.ru/article.aspx?id=10007. – Клавіатурний почерк.
7. Біометричні системи безпеки [Електронний ресурс]. – Режим доступу: http://uadoc.zavantag.com/text/23710/index-1.html. – Клавіатурний почерк.
8. Інформаційні системи і технології в юридичній діяльності. [Електронний ресурс]. – Режим доступу: http://ubooks.com.ua/books/000166/inx18.php. – Біометричний захист інформації.
9. Анатомия человека Э.И. Борзяк, В.Я. Бочаров, Л.И. Волкова – М.: Медицина, 1987.
10. Біометрична автентифікація. [Електронний ресурс]. – Режим доступу: http://ukrdoc.com.ua/text/5859/index-11.html. – Клавіатурний почерк.
11. Расторгуев С.П. Програмные методы защиты информации в компютерах и сетях. – М.: Изд. Агенства «Яхтсмен», 2004.
12. Корнеева А.П. Машинопись и основы современного делопроизводства. – М.: Просвещение, 2005.
13. Coltell O., Badia J. M., Torres G. Biometric Identification System Based in Keyboard Filtering // Proc. of XXXIII Annual IEEE International Carnahan Conference on Security Technology. – 2008. – P. 203-209.
14. *Брюхомицкий Ю.А., Казарин М.Н.* Параметрическое обучение биометрических систем контроля доступа / Ю.А. Брюхомицкий, М.Н.

Казарин // Вестник компьютерных и информационных технологий. – М.: Машиностроение, 2006. – № 2 (20). – С. 6-13.
15. Нейронные сети: персептронные сети, обратное распространение ошибки, сети хопфилда» [Електронний ресурс].– Режим доступу: http://compscicenter.ru/program/lecture/6296 - Нейронние сети.
16. Федотов А. В. Моделирование нейронних сетей / Р.Э. Смит. – М.: «Вильямс», 2010. – 32 с.
17. Исследование еффективности применения вероятносних нейронних сетей для решения задачи аутентификаци пользователя. [Електронний ресурс]. – Режим доступу: http://pnzzi.kpi.ua. – Нейронние сети.
18. Петров Р. В. Иммунология / Р. В. Петров — М.: Медицина, 1987. — 416 с.
19. Harty J. CD8+ T cell effector mechanisms in resistance to infection / J. Harty, A. Tvinnereim, D. White // Annu Rev Immunol. — Vol. 18. — 2010. — P. 275-308.
20. Kephart J.O. A biologically inspired immune system for computers / J.O. Kephart // Proceedings of Artificial Life IV: The Fourth International Workshop on the Synthesis and Simulation of Living Systems. — 2008. — P. 130–139.
21. Utpal Garain, Mangal P. Chakraborty, Dipankar Dasgupta.≫ Recognotion of handwritten indic script using Clonal Selection Algorithm≫. H. Bersini and J.Carneiro(Eds.): ICARIS 2006, LNCS 4163, pp.256-266, 2006.
22. Литвиненко В.И., Дидык А.А., Захарченко Ю.А. Компьютерная система для решения задач классификации на основе модифицированных иммунных алгоритмов // Информационноизмерительные системы. - ААЭКС. - 2008. - Т.22. - №2.
23. Julie Greensmith, Uwe Aickelin, Gianni Tedesco. Information Fusion for Anomaly Detection with the Dendritic Cell Algorithm. Information Fusion 11 (1). 2010. - 21-34pp.
24. Neural Network Toolbox - MATLAB - MathWorks [Електронний ресурс]. – Режим доступу: https://www.mathworks.com/products/neural-network.html – Назва з екрану

Додаток А

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

ЗАТВЕРДЖУЮ
Завідувач кафедри ЗІ,
д. т. н., проф.
_____ В. А. Лужецький

"____" _____ 2019 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на магістрську кваліфікаційну роботу
"Модель і засіб автентифікації за клавіатурним почерком з використанням штучних імунних мереж"
08-20.МКР.013.00.000 ТЗ

Керівник: к.т.н., проф. каф. ЗІ
_____Кондратенко Н.Р.
_____ _____ 2019р.

Вінниця 2019

1 Назва та область використання

Засіб автентифікації за клавіатурним почерком з використанням нейромереж.

2 Основа для розробки

Робота проводиться на підставі наказу ректора ВНТУ №254 від 02.10.2019 р.

3 Мета та призначення розробки

Підвищення ефективності захисту комп'ютерних систем і мереж від несанкціонованого доступу.

4 Джерела розробки

1. Есин В. И. Безопасность информационных систем и технологий / В. И. Есин, А. А. Кузнецов, Л. С. Сорока. – Х. : ООО «ЭДЭНА», 2010. – 656 с.
2. Шаньгин В. Ф. Информационная безопасность компьютерных систем и сетей : учеб. пособие. – М. : ИД «ФОРУМ» : ИНФРА-М, 2011 – 416 с.
Інформаційні системи і технології в юридичній діяльності.
3. [Електронний ресурс]. – Режим доступу: http://ubooks.com.ua/books/000166/inx18.php. – Біометричний захист інформації.
4. Ричард Э. Смит. Автентификация: от паролей до открытых ключей./ Р.Э. Смит. – М.: «Вильямс», 2002. – 432 с.
5. Захист інформації з використанням біометричних систем [Електронний ресурс]. – Режим доступу: http://www.rusnauka.com/35_OINBG_2010/Informatica/76206.doc.htm - Суть і значення інформаційних біометричних систем.

5 Вимоги до програмного засобу

5.1 Параметри розроблюваної системи захист:

– операційна система – Windows;

– об'єкт захисту – інформаційно комунікаційна система підприємства;

– напрямок захисту – захист системи від несанкцонованого доступу;

– мова програмування – Python;

– середовище розробки – JetBrains PyCharm, MATLAB 2017.

5.2 Програма тестувалась на пристрої з наступними характеристиками:

– оперативна пам'ять – 8 Гб;

– середовище функціонування – ОС Windows 10.

6 Вимоги до програмної документації

Графічна і текстова документація повинна відповідати діючим стандартам України .

7 Стадії та етапи розробки

Робота з теми виконується в чотири етапи.

Етап	Назва етапів магістерської кваліфікаційної роботи	Початок	Закінчення	Результат
1	Аналіз завдання. Вступ	05.09.2019	09.09.2019	Чорновик вступу
2	Розробка технічного завдання	14.09.2019	20.09.2019	Проект технічного завдання
3	Аналіз літературних джерел за напрямком магістерської кваліфікаційної роботи	05.09.2019	15.09.2019	Чорновик першого розділу. Схеми та алгоритми
4	Практична реалізація, моделювання, експерментування, результати	14.10.2019	10.11.2019	Програмний засіб. Розділ пояснювальної записки
5	Аналіз виконання ТЗ, висновки	18.11.2019	24.11.2019	Висновки, інструкції
6	Оформлення пояснювальної записки	25.11.2019	30.11.2019	Пояснювальна записка, графічний матеріал

8 Порядок контролю та прийому.

До прийому і захисту магістерської кваліфікаційної роботи подається:

- остаточний звіт (пояснювальна записка);
- робоча програма;
- інструкції для роботи з програмою;
- графічні матеріали.

Початок розробки 01.09.2019
Крайній термін виконання магістерської кваліфікаційної роботи 20.12.2019

Розробив студент групи 1БС-18м _____ Рудик О.А.

Додаток Б

Лістинг програми збору даних

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
namespace KeyCollectorGUI{
   public partial class MainWindow : Window   {
      public MainWindow()  {
         InitializeComponent();
         this.Content = new LoginControl();        }
   public partial class TestTextControl : UserControl   {
      static Brush Good = Brushes.LightGreen;
      static Brush Bad = Brushes.LightSalmon;
      string sampleText = null;
      string defaultSample = "The five boxing wizards jump quickly. See the quick brown fox jump over the lazy dog. A mad boxer shot a quick, gloved jab to the jaw of his dizzy opponent. We promptly judged antique ivory buckles for the next prize. A quart jar of oil mixed with zinc oxide makes a very bright paint. The job requires extra pluck and zeal from every young wage earner. About sixty codfish eggs will make a quarter pound of very fizzy jelly. All questions asked by five watch experts amazed the judge. Big July earthquakes confound zany experimental vow. I have quickly spotted the four women dozing in the jury box.";
      string userString = null;
      Run building = null;
      LinkedList<Run> runs = null;
      Logger logger = null;
      string userName = null;
      public TestTextControl(string userName = "")
      {
         sampleText = defaultSample;

         InitializeComponent();
         this.userName = userName;

         testTextBox.TextInput += new TextCompositionEventHandler(testText_PreviewTextInput);
         testTextBox.PreviewKeyDown += new KeyEventHandler(testTextBox_PreviewKeyDown);
         logger = new Logger();

         reset();

         Application.Current.MainWindow.Closing += new
System.ComponentModel.CancelEventHandler(MainWindow_Closing);

         testTextBox.Focus()  }

      ~TestTextControl() {
   if (logger != null)        {
         logger.stop();
         logger = null;       }     }
```

```
protected void reset()
{

    sampleText = sampleText.Replace("\r", "");

runs = new LinkedList<Run>();
    building = new Run(string.Empty);
    building.Background = Good;
    userString = string.Empty;

    logger.reset();

    updateParagraph();      }

protected void updateParagraph()      {
    untouched.Text = sampleText.Substring(Math.Min(sampleText.Length, userString.Length));

    PWrapper.Inlines.Clear();

    foreach (Run r in runs)          {
       PWrapper.Inlines.Add(r);          }
    PWrapper.Inlines.Add(building);
    PWrapper.Inlines.Add(untouched);
    System.Windows.Documents.TextPointer docStart = testTextBox.Document.ContentStart;
    System.Windows.Documents.TextPointer docEnd = testTextBox.Document.ContentEnd;
    int trueLength = docStart.GetOffsetToPosition(docEnd);
    int userStringEndOffset = trueLength - (sampleText.Length - userString.Length) - 2;
    System.Windows.Documents.TextPointer tp = testTextBox.Document.ContentStart;
    tp = tp.GetPositionAtOffset(userStringEndOffset + 10);  // look ahead 10 characters
    if (tp == null)
    {
       tp = docEnd;
    }
    testTextBox.Selection.Select(tp, tp);      }

protected void addCharacter(string c)      {
    userString += c;
    Brush runType = null;

    if (userString.Length > sampleText.Length)          {
       c = " ";
       runType = Bad;          }
    else          {
       c = sampleText[userString.Length - 1].ToString();
       if (userString[userString.Length - 1] == sampleText[userString.Length - 1])          {
          runType = Good;          }
       else          {
          runType = Bad;          }          }

if (runType != null)          {
    if (building.Background == runType)          {
       building.Text += c;          }          else
 {
       if (building.Text.Length > 0)   // never add empty runs          {
          runs.AddLast(building);          }
       building = new Run(c);
       building.Background = runType;          }          }
```

```
    updateParagraph();      }

protected void deleteCharacter()      {
    if (userString.Length >= 1)      {
        userString = userString.Substring(0, userString.Length - 1);

        if (building.Text.Length == 0 && runs.Count > 0)          {
            building = runs.Last();
            runs.RemoveLast();          }

        if (building.Text.Length > 0)          {
            building.Text = building.Text.Substring(0, building.Text.Length - 1);          }

        updateParagraph();      }      }

protected void testText_PreviewTextInput(Object sender, TextCompositionEventArgs e)      {
    if (e.Text != Convert.ToChar(27).ToString())    // ESCAPE should not be a character      {
        if (e.Text == "\r")          {
            addCharacter("\n");          }
        else      {
            addCharacter(e.Text);          }      }      }

protected void testTextBox_PreviewKeyDown(Object sender, KeyEventArgs e)      {
    switch (e.Key)      {
        case Key.Space:
            addCharacter(" ");
            break;
        case Key.Tab:
            addCharacter("\t");
            break;
        case Key.Back:
            deleteCharacter();
            break;      }      }

protected void MainWindow_Closing(object sender, System.ComponentModel.CancelEventArgs e)      {
    if (logger != null)      {
        logger.stop();
        logger = null;      }      }

private bool log_saveas()      {
    string      regexSearch      =      new      string(System.IO.Path.GetInvalidFileNameChars())      +      new
string(System.IO.Path.GetInvalidPathChars());
    Regex r = new Regex(string.Format("[{0}]", Regex.Escape(regexSearch)));
    string proposed = r.Replace(userName, "");
    proposed += "-" + DateTime.Now.ToString("MM-dd-yy_HH-mm-ss");
    Microsoft.Win32.SaveFileDialog dlg = new Microsoft.Win32.SaveFileDialog();
dlg.FileName = proposed;    // default file name
dlg.DefaultExt = ".log";    // default file extension
dlg.Filter = "Log files|*.log|Text Documents|*.txt|All Files|*.*";  // file selection filters

Nullable<bool> result = dlg.ShowDialog();
if (result == true)      {
    using (StreamWriter logWriter = new StreamWriter(dlg.FileName))          {
        logWriter.Write(logger.Log);          }

    return true;      }
else      {
```

```
        return false;          }       }

private void exitButton_Click(object sender, RoutedEventArgs e)       {
   if (log_saveas())          {
      Application.Current.MainWindow.Close();          }       }

private void resetButton_Click(object sender, RoutedEventArgs e)       {
   reset();
   testTextBox.Focus();       }

private void loadButton_Click(object sender, RoutedEventArgs e)       {
   Microsoft.Win32.OpenFileDialog dial = new Microsoft.Win32.OpenFileDialog();
         Nullable<bool> open = dial.ShowDialog();
         if (open == true)          {
            string filename = dial.FileName;
            StreamReader sr = new StreamReader(filename);
            sampleText = sr.ReadToEnd();
            sr.Close();          }
         reset();
         testTextBox.Focus();       }    }}
   class Logger    {
      #region Hook Structures
      private class KeyboardHookStruct
      {
         public int vkCode;
         public int scanCode;
         public int flags;
         public int time;
public int dwExtraInfo;
      }
      #endregion
#region Windows function imports
         int idHook,
         HookProc lpfn,
         IntPtr hMod,
         int dwThreadId);
         CallingConvention = CallingConvention.StdCall, SetLastError = true)]
      private static extern int UnhookWindowsHookEx(int idHook);
      [DllImport("user32.dll")]
      private static extern int GetKeyboardState(byte[] pbKeyState);

      [DllImport("user32.dll", CharSet = CharSet.Auto, CallingConvention = CallingConvention.StdCall)]
      private static extern short GetKeyState(int vKey);

      [DllImport("Kernel32.dll", EntryPoint = "GetCurrentThreadId")]
      public static extern Int32 GetCurrentWin32ThreadId();

      #endregion
      public Logger()
      {
         hookKeyboard();

         reset();
      }
      public void reset()
      {
         _log = new StringWriter();
      }
```

```
public void stop()
{
   unhookKeyboard();
}
private void logEvent(int keyCode, string eventType)
{
   string timeStamp = (DateTime.UtcNow.Ticks / TimeSpan.TicksPerMillisecond).ToString(); // ToString("HH:mm:ss:ffff");
   string logLine = string.Format("{0} {1} {2}", timeStamp, keyCode, eventType);
   _log.WriteLine(logLine);
}
   hKeyboardHook = SetWindowsHookEx(
      WH_KEYBOARD_LL,
      KeyboardHookProcedure,
      Marshal.GetHINSTANCE(Assembly.GetExecutingAssembly().GetModules()[0]),
      0);
   private void unhookKeyboard()
   {
int retKeyboard = UnhookWindowsHookEx(hKeyboardHook);
      hKeyboardHook = 0;
      if (retKeyboard == 0)
      {
         int errorCode = Marshal.GetLastWin32Error();

         Console.WriteLine("[Error {0}: Failed to unhook keyboard]", errorCode);
      }
   }
      if (wParam == WM_KEYDOWN || wParam == WM_SYSKEYDOWN)
      {
         int keyCode = keyboardHookStruct.vkCode;
         logEvent(keyboardHookStruct.vkCode, "KEY_DOWN");
      }

         if (ToAscii(keyboardHookStruct.vkCode,
               keyboardHookStruct.scanCode,
               keyState,
               inBuffer,
               keyboardHookStruct.flags) == 1)
         {
      if (wParam == WM_KEYUP || wParam == WM_SYSKEYUP)
      {
         logEvent(keyboardHookStruct.vkCode, "KEY_UP");
      }
   }
      return CallNextHookEx(hKeyboardHook, nCode, wParam, lParam);
   }
}
```

Лістинг програми обробки даних

```
import sys
import argparse
import re
import os
import os.path
from PyQt4.QtGui import *
from PyQt4.Qt import *
from pybrain.tools.shortcuts import buildNetwork
from pybrain.datasets import ClassificationDataSet
from pybrain.structure.modules import SigmoidLayer, SoftmaxLayer, LinearLayer
from pybrain.supervised.trainers import BackpropTrainer
from pybrain.supervised.trainers import RPropMinusTrainer


def init_brain(learn_data, epochs, TrainerClass=BackpropTrainer):
    if learn_data is None:
        return None
    print ("Building network")
    # net = buildNetwork(20, 25, 9, hiddenclass=TanhLayer)
    # net = buildNetwork(20, 50, 9)
    net = buildNetwork(20, 75,9, hiddenclass=LinearLayer)
    # fill dataset with learn data
    trans = {
        'User A': 1, 'User B': 2, 'User C': 3, 'User D': 4, 'User E': 5 , 'User F':
6, 'User G': 7, 'User J': 8, 'Non user': 9
    }
    ds = ClassificationDataSet(4096, nb_classes=5, class_labels=['A', 'B', 'C',
'D', 'Z'])
    for inp, out in learn_data:
        ds.appendLinked(inp, [trans[out]])
    ds.calculateStatistics()
    print ("\tNumber of classes in dataset = {0}".format(ds.nClasses))
    print ("\tOutput in dataset is ", ds.getField('target').transpose())
    ds._convertToOneOfMany(bounds=[0, 1])
    print ("\tBut after convert output in dataset is \n", ds.getField('target'))
    trainer = TrainerClass(net, verbose=True)
    trainer.setData(ds)
    print("\tEverything is ready for learning.\nPlease wait, training in
progress...")
    trainer.trainUntilConvergence(maxEpochs=epochs)
    print("\tOk. We have trained our network.")
    return net


def loadData(dir_name):
    list_dir = os.listdir(dir_name)
    list_dir.sort()
list_for_return = []
print ("Loading data...")
for filename in list_dir:
    out = [None, None]
    print("Working at {0}".format(dir_name + filename))
    print("\tTrying get user data.")
    lett = re.search("\w+_(\w)_\d+\.png", dir_name + filename)
    if lett is None:
        print ("\tFilename not matches pattern.")
        continue
```

```
            else:
                print("\tFilename matches! User is '{0}'.
Appending...".format(lett.group(1)))
                out[1] = lett.group(1)
            print("\tTrying get user.")
            out[0] = get_data(dir_name + filename)
            print("\tChecking data size.")
            if len(out[0]) == 20:
                print("\tSize is ok.")
                list_for_return.append(out)
                print("\tInput data appended. All done!")
            else:
                print("\tData size is wrong. Skipping...")
        return list_for_return


    def get_data(png_file):
        img = QImage(64, 64, QImage.Format_RGB32)
        data = []
        if img.load(png_file):
            for x in range(64):
                for y in range(64):
                    data.append(qGray(img.pixel(x, y)) / 255.0)
        else:
            print ("img.load({0}) failed!".format(png_file))
        return data


    def work_brain(net, inputs):
        rez = net.activate(inputs)
        idx = 0
        data = rez[0]
        for i in range(1, len(rez)):
            if rez[i] > data:
                idx = i
                data = rez[i]
        return (idx, data, rez)


    def test_brain(net, test_data):
        for data, right_out in test_data:
            out, rez, output = work_brain(net, data)
            print ("For '{0}' our net said that it is '{1}'. Raw =
{2}".format(right_out, "ABCDZ"[out], output))
        pass


    def main():
        app = QApplication([])
        p = argparse.ArgumentParser(description='PyBrain example')
        p.add_argument('-l', '--learn-data-dir', default="./learn", help="Path to dir,
containing learn data")
        p.add_argument('-t', '--test-data-dir', default="./test", help="Path to dir,
containing test data")
        p.add_argument('-e', '--epochs', default="1000", help="Number of epochs for
teach, use 0 for learning until convergence")
        args = p.parse_args()
        learn_path = os.path.abspath(args.learn_data_dir) + "/"
        test_path = os.path.abspath(args.test_data_dir) + "/"
        if not os.path.exists(learn_path):
```

```
        print("Error: Learn directory not exists!")
        sys.exit(1)
    if not os.path.exists(test_path):
        print("Error: Test directory not exists!")
        sys.exit(1)
    learn_data = loadData(learn_path)
    test_data = loadData(test_path)
    # net = init_brain(learn_data, int(args.epochs),
TrainerClass=RPropMinusTrainer)
    net = init_brain(learn_data, int(args.epochs), TrainerClass=BackpropTrainer)
    print ("Now we get working network. Let's try to use it on learn_data.")
    print("Here comes a tests on learn-data!")
    test_brain(net, learn_data)
    print("Here comes a tests on test-data!")
    test_brain(net, test_data)
    return 0

if __name__ == "__main__":
    sys.exit(main())
```

ІЛЮСТРАТИВНА ЧАСТИНА

ТАБЛИЦЯ СТАТИСТИЧНИХ ДАНИХ РЕЗУЛЬТАТІВ РОБОТИ ШТУЧНОЇ ІМУННОЇ СИСТЕМИ

Користувач	Ідентифікатор користувача	Результат роботи нейромережі
User A	1	Was identyfied as User A
User B	2	Was identyfied as User B
User C	3	Was identyfied as User C
User D	4	Was identyfied as User D
User E	5	Was identyfied as User E
User F	6	Was identyfied as User F
User G	7	Wasn't identified
User J	8	Was identyfied as User J
Unknown user A	0	Wasn't identified
Unknown user B	0	Wasn't identified

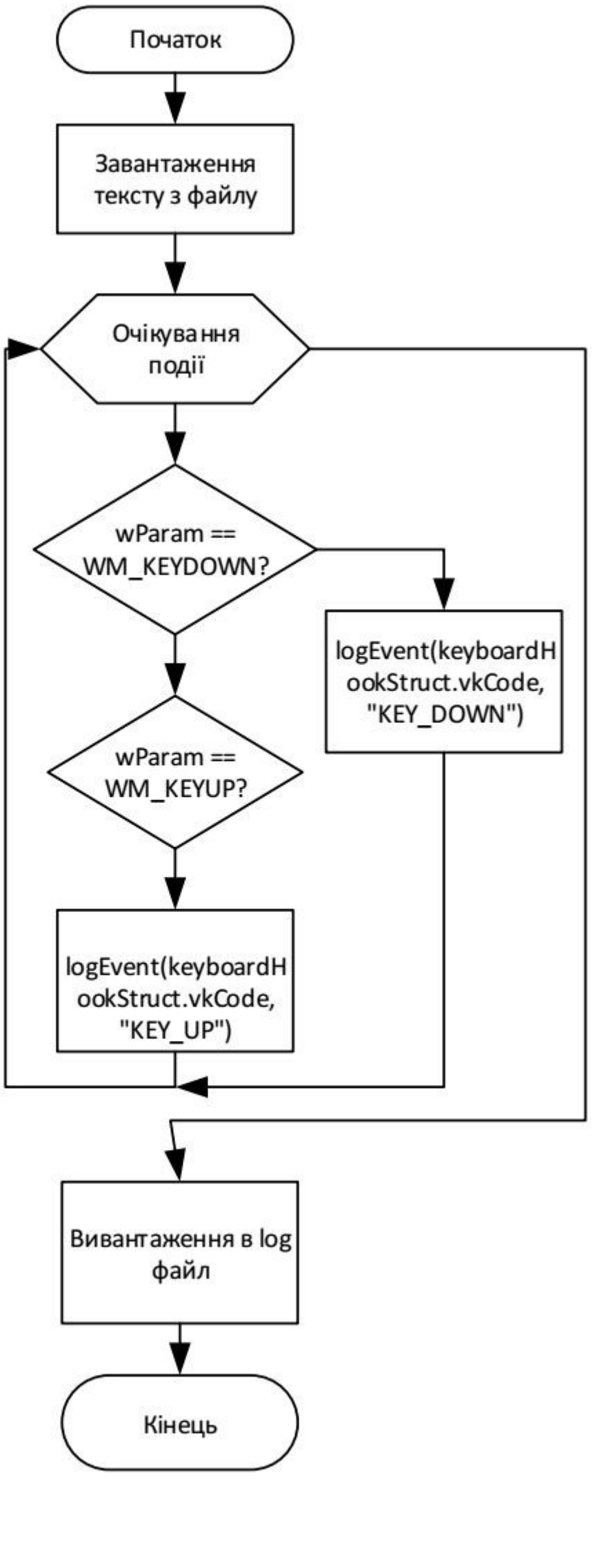
					08-20.МКР.013.00.000 ІЧ1			
Змн.	Арк.	№ докум.	Підпис	Дат				
Розроб.		Рудик О.А.			Засіб автентифікації за клавіатурним почерком. Результати роботи штучної імунної системи	Літ.	Арк.	Аркушів
Перевір.		Кондратенко					1	1
Реценз.		Крупельницький				ВНТУ, БС-18		
Н. Контр.		Кондратенко						
Затверд.		Лужец						

ТАБЛИЦЯ СТАТИСТИЧНИХ ДАНИХ ЗАЛЕЖНОСТІ НАВИЧОК КОРИСТУВАЧА І ЕФЕКТИВНОСТІ СИСТЕМИ РОЗПІЗНАВАННЯ

Помилки, %	Аритмічність, %	Швидкість зн./хв.	Характеристика перекриття		Оцінка
			Число перекриття, %	Число задіяних пальців	
менше 2	менше 10	більше 200	більше 50	Всі	Відмінно
менше 4	менше 15	більше 150	більше 30	Більшість	Добре
менше 8	менше 20	більше 100	більше 10	Декілька	Задовільно
більше	більше 20	менше 100	менше 10	По одному	Незадовільно

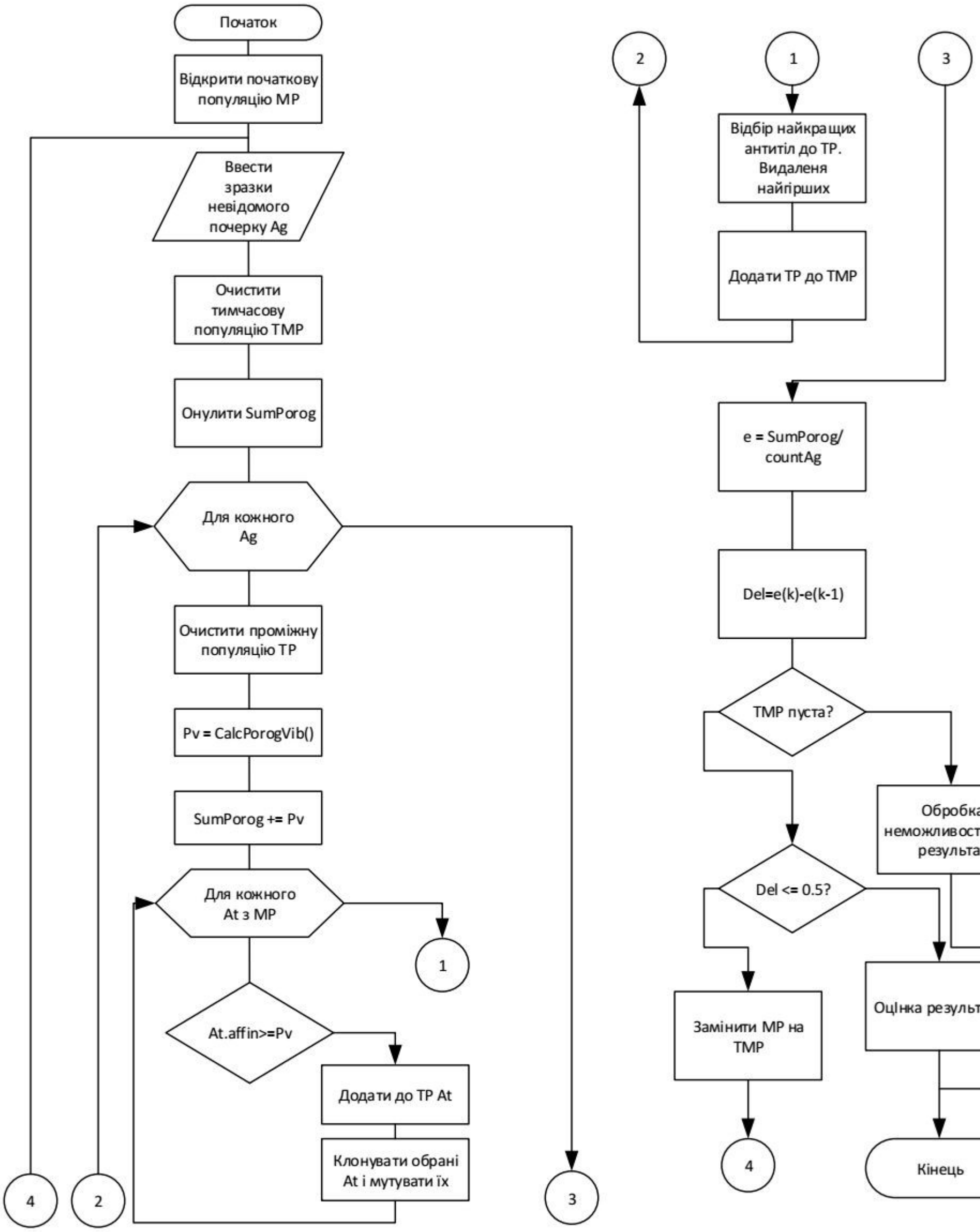
					08-20.МКР.013.00.000 ІЧ2			
Змн.	Арк.	№ докум.	Підпис	Дат				
Розроб.		Рудик О.А.			Засіб автентифікації за клавіатурним почерком. Залежність ефективності системи від навичок користувача	Літ.	Арк.	Аркушів
Перевір.		Кондратенко					1	1
Реценз.		Крупельницький				ВНТУ, БС-18		
Н. Контр.		Кондратенко						
Затверд.		Лужец						

БЛОК-СХЕМА КЛІЄНТСЬКОЇ ЧАСТИНИ ЗАСОБУ



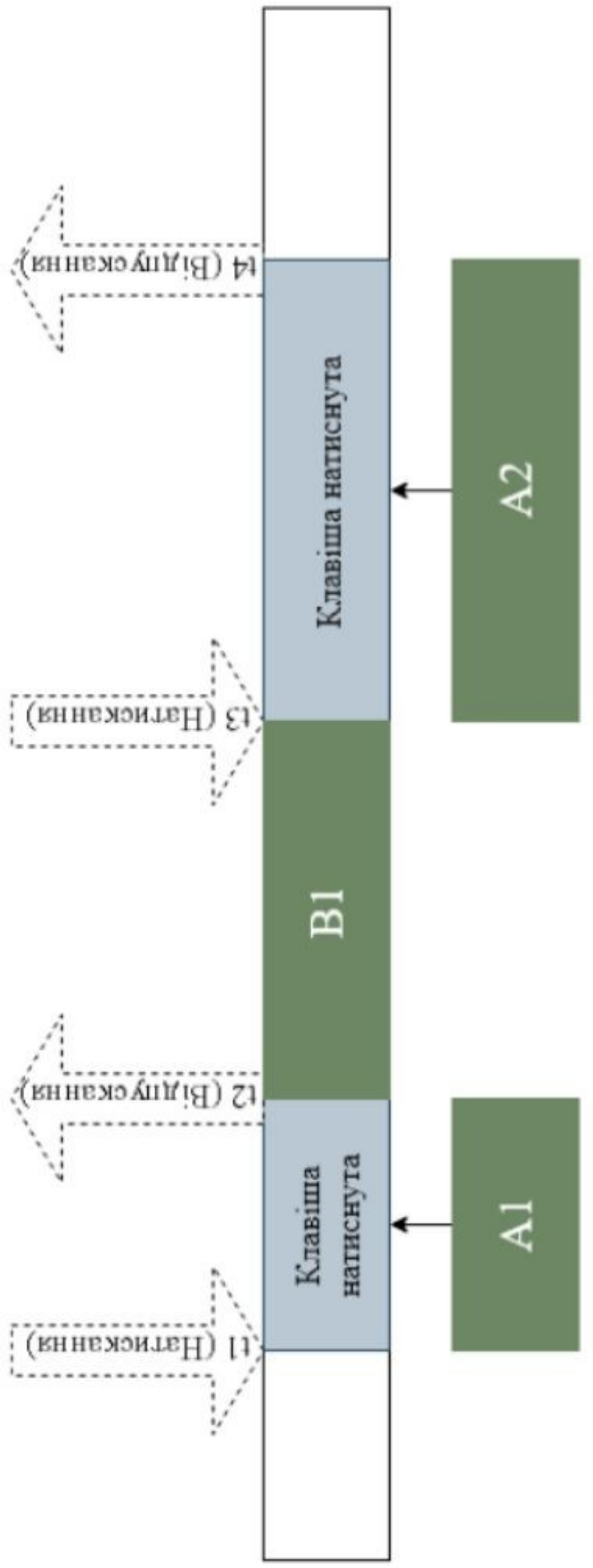
					08-20.МКР.013.00.000 ІЧ3			
Змн.	Арк.	№ докум.	Підпис	Дат				
Розроб.		Рудик О.А.			Засіб автентифікації за клавіатурним почерком. Блок-схема клієнтської частини засобу	Літ.	Арк.	Аркушів
Перевір.		Кондратенко					1	1
Реценз.		Крупельницький				ВНТУ, БС-18		
Н. Контр.		Кондратенко						
Затверд.		Лужец						

БЛОК-СХЕМА СЕРВЕРНОЇ ЧАСТИНИ ЗАСОБУ



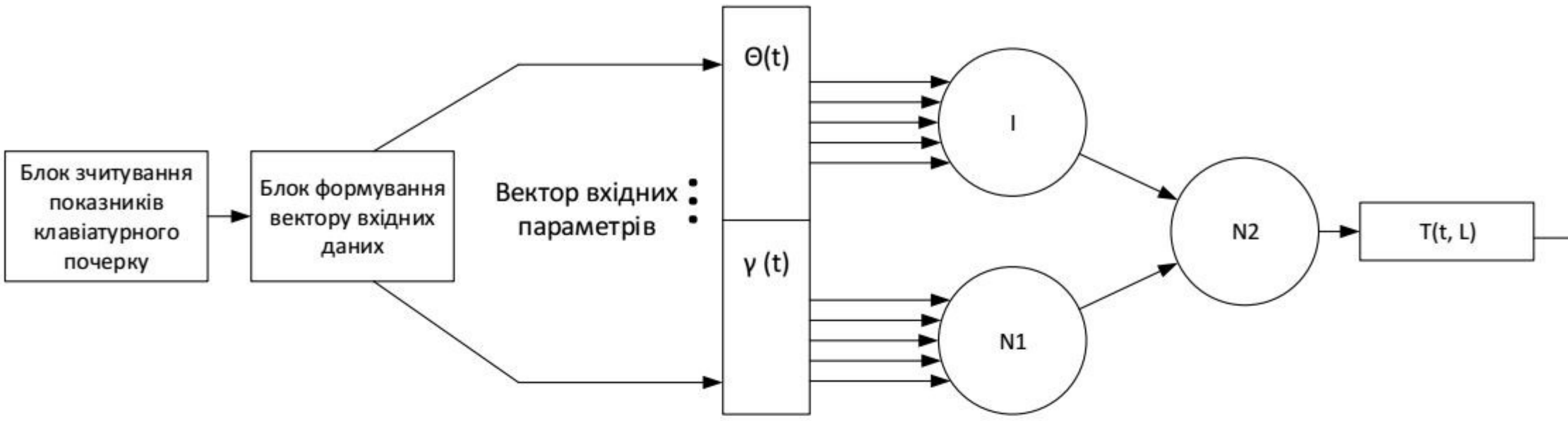
					08-20.МКР.013.00.000 ІЧ4			
Змн.	Арк.	№ докум.	Підпис	Дат				
Розроб.		Рудик О.А.			Засіб автентифікації за клавіатурним почерком. Блок-схема серверної частини засобу	Літ.	Арк.	Аркушів
Перевір.		Кондратенко					1	1
Реценз.		Крупельницький				ВНТУ, БС-18		
Н. Контр.		Кондратенко						
Затверд.		Лужец						

ЧАСОВА ДІАГРАМА КЛАВІАТУРНОГО ПОЧЕРКУ



					08-20.МКР.013.00.000 ІЧ5			
Змн.	Арк.	№ докум.	Підпис	Дат				
Розроб.		Рудик О.А.			Засіб автентифікації за клавіатурним почерком. Часова діаграма клавіатурного почерку	Літ.	Арк.	Аркушів
Перевір.		Кондратенко					1	1
Реценз.		Крупельницький				ВНТУ, БС-18		
Н. Контр.		Кондратенко						
Затверд.		Лужец						

СТРУКТУРНА СХЕМА ЗАСОБУ АВТЕНТИФІКАЦІЇ ЗА КЛАВІАТУРНИМ ПОЧЕРКОМ



					08-20.МКР.013.00.000 ІЧ6			
Змн.	Арк.	№ докум.	Підпис	Дат				
Розроб.		Рудик О.А.			Засіб автентифікації за клавіатурним почерком. Структуна схема засобу автентифікації за клавіатурним	Літ.	Арк.	Аркушів
Перевір.		Кондратенко					1	1
Реценз.		Крупельницький				ВНТУ, БС-18		
Н. Контр.		Кондратенко						
Затверд.		Лужец						