Міністерство освіти і науки України

Національний університет «Києво-Могилянська академія»

Факультет інформатики

Кафедра інформатики

Магістерська робота

освітній ступінь – магістр

на тему: «**АВТЕНТИФІКАЦІЯ КОРИСТУВАЧІВ У ВЕБ-ЗАСТОСУНКАХ ЗА АКТИВНІСТЮ ВКАЗІВНОГО ПРИСТРОЮ**»

Виконав: студент 2-го року навчання,

Спеціальності
121 Інженерія програмного забезпечення

Каруна Даниїл Геннадійович

Керівник Ковалюк Т. В.,
кандидат технічних наук, доцент.

Рецензент _____
(прізвище та ініціали)

Магістерська робота захищена
з оцінкою _____

Секретар ЕК _____
«____» _____ 20___ р.

Київ – 2021

Міністерство освіти і науки України

Національний університет «Києво-Могилянська академія»

Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

кандидат технічних наук, доцент

_____Т. В. Ковалюк

(підпис)

„____”_____2020 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на дипломну роботу

студенту Каруні Д.Г. факультету інформатики 2 курсу МП ІПЗ

Тема: Автентифікація користувачів у веб-застосунках за активністю вказівного пристрою

Зміст ТЧ до магістерської роботи:

Зміст

Анотація

Вступ

1 Отримання даних, трансформація та навчання моделі.

2 Огляд та результати експерименту

3 Порівняння результатів та подальше вдосконалення

Висновки

Список літератури

Додатки

Дата видачі „15” листопада 2020 р. Керівник _____

(підпис)

Завдання отримав _____

(підпис)

Тема: Автентифікація користувачів у веб-застосунках за активністю вказівного пристрою

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на дипломну роботу.	15.11.2020	
2.	Огляд існуючої літератури та робіт.	24.12.2020	
3.	Аналіз алгоритмів та методів дослідження.	18.01.2021	
3.	Розробка розширення та серверу для збору даних.	15.02.2021	
4.	Отримання та трансформація даних від учасників експерименту	15.03.2021	
5.	Навчання моделей, порівняння результатів	12.04.2021	
6.	Написання пояснювальної записки	10.05.2021	
7.	Попередній захист роботи	14.05.2021	
8.	Аналіз отриманих результатів з керівником та корегування роботи.	24.05.2021	
9.	Створення слайдів для доповіді та написання доповіді.	4.06.2021	
10.	Захист магістерської роботи (проекту)	16.06.2021	

Студент _____

Керівник _____

“_____”_____

ЗМІСТ

АНОТАЦІЯ

Робота складається с трьох розділів. В першому розділі розглянуто методи та алгоритми отримання даних, а також процес створення та навчання моделі. В другому розділі описаний процес дослідження та результати. У третьому розділі проведено оцінку результатів та ефективності, розглянуто подальші можливості застосування методу.

ВСТУП

У 2021 році більшість застосунків побудовано за допомогою веб-технологій. Це дозволяє користувачам отримувати потрібну інформацію, а також виконувати певні операції з будь-якого пристрою за допомогою веб-браузера. Перед розробниками постає задача створення повноцінного застосунку в обмеженому середовищі браузерів (неповний доступ до файлової системи, пристроїв, системних API тощо).

Одна з основних проблем - задача забезпечення безпеки користувачів у веб-застосунках. Безпека користування веб-застосунком складається з багатьох шарів, які можна поділити на дві основні групи: ті, що підконтрольні розробнику та ті, що повністю залежать від користувача та обставин користування застосунком. При розробці застосунку є можливість використати різні методи шифрування, застосувати безпечні протоколи, алгоритми перевірки даних, встановити багатофакторну автентифікацію тощо. За допомогою цих та інших інструментів розробник може мінімізувати загрозу перехоплення особистих даних та неавторизованих дій. Але ці міри не захищають користувача від ситуацій, де стороння людина може отримати фізичний доступ до пристрою з доступом до системи та виконати потенційно зловмисні дії. Для цього існують окремі методи та алгоритми.

В багатьох системах, що вимагають високий рівень безпеки користувача (банківські, корпоративні), вже тривалий час існують моделі поведінки користувача, що навчаються на аналізі дій користувачів. Вони знаходять залежності між кількістю, типом, часом операцій та багатьма іншими факторами для того, щоб ідентифікувати аномальні дії та запобігти потенційно шкідливим наслідкам.

В цій роботі розглядається один з методів аналізу поведінки та автентифікації користувача. Цей метод базується на припущенні, що кожна

людина має відмінну поведінку при користуванні вказівним пристроєм: мишею, пальцем, трекпадом. Проаналізувавши цю поведінку можна створити або розширити модель, яка зможе вказувати на аномальну поведінку та реагувати на неї різними способами. Наприклад, у багатьох банківських системах в залежності від показника аномальності дій створюються додаткові бар’єри для користувача: підтвердження за допомогою СМС-повідомлення, додатковий запит пароля та інші. В даній роботі розглянуто метод та алгоритм отримання даних користувачів, метод навчання моделі, а також проведене дослідження на реальних користувачах з оглядом результатів.

РОЗДІЛ 1. ОТРИМАННЯ ДАНИХ, ТРАНСФОРМАЦІЯ ТА НАВЧАННЯ МОДЕЛІ.

1.1 Визначення необхідних даних для отримання

В браузерному середовищі доступна дуже обмежена кількість інформації про вказівні пристрої. На відміну від повноцінних застосунків, в браузері доступна тільки інформація про координати миші та про стан кнопок та колеса миші.

Більшість інших робіт з аналізу активності вказівного пристрою [1][2][3], незалежно від середовища, використовують запис координат курсору та станів кнопок кожний фіксований проміжок часу. При отриманні даних не проводиться аналіз рухів, трансформація або розділення на дії. Дані передаються у виді масиву з координатами, часовими мітками та станами кнопок.

Після аналізу інших робіт та оцінки необхідних даних було вирішено збирати дані у вигляді масиву об’єктів, що складаються з: ідентифікатора користувача, ідентифікатора сесії, часової мітки сесії, координат курсора та станів кнопок вказівного пристрою.

1.2 Отримання даних

Для отримання даних користувачів було створено розширення (плагін) для браузеру Google Chrome, що було встановлено користувачам, що погодилися на експеримент.

1.2.1 Модель подій у браузері

Сучасні браузери реалізують доступні розробникам API за стандартами та рекомендаціями W3C (World Wide Web Consortium). HTML-документи при завантаженні браузером перетворюються на DOM (Document Object Model), що є об’єктно-орієнтованим представленням документу у вигляді дерева. Після створення DOM браузер надає певний API для маніпуляцій деревом елементів, створення нових елементів, а також обробки подій.

В DOM кожен елемент має певну кількість можливих подій, що викликаються в залежності від дій користувача, а також подій самого браузера. Перелік можливих подій залежить від типу цього елементу. Завдяки мові JavaScript, розробники можуть створювати обробники цих подій та отримувати необхідні дані.

Для отримання даних вказівного пристрою є можливість використовувати події зі специфікацій Pointer Events [4] та UI Events [5]. В даній роботі дослідження проводиться на пристроях, що ведуть себе як комп’ютерна миша (тобто сама миша або тачпад чи трекпад). Існує можливість адаптувати алгоритм для сенсорних екранів та інших типів вводу, але це знаходиться за рамками цього дослідження. Серед подій, які було використано:

- mousemove – подія, що відбувається під час руху миші. З неї можна отримати інформацію про координати миші та значення відстані, що була пройдена мишею по осям.
- mousedown – подія, що відбувається коли користувач натискає будь-яку кнопку миші.
- mouseup – подія, що відбувається коли користувач відпускає будь-яку кнопку миші.

За допомогою спеціальних API від Google Chrome кожному користувачу було надано унікальний анонімний ідентифікатор, що представляє собою стрічку, створену з 32 випадково згенерованих байтів. Цей ідентифікатор зберігається поміж перезапусками браузеру та перезавантаженнями комп'ютеру. Він видаляється після видалення розширення з браузеру, не містить персональної інформації та не може пов'язати користувача чи його пристрій з його діями.

Це розширення складається з двох частин: змістового скрипту, який встановлюється в кожну відкриту сторінку та має доступ до її змісту (далі – Content Script), та фонового скрипту, який працює на рівні браузеру незалежно від відкритих сторінок (далі – Background Script).

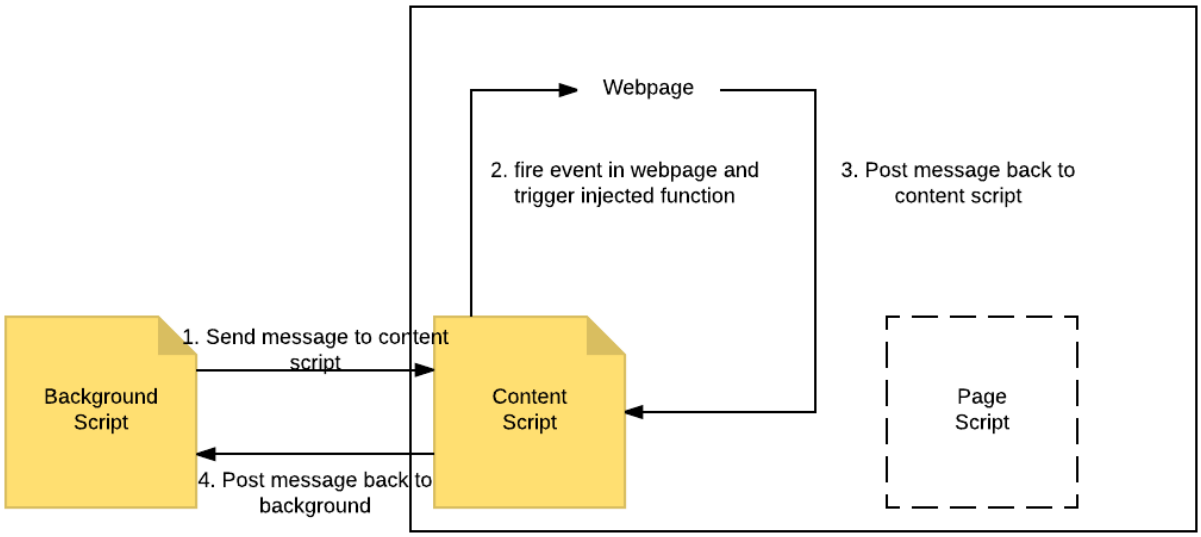

Рисунок 1.1. Діаграма комунікації між content та background сценаріями.

1.2.2 Content script

Content script встановлює інтервал отримання інформації та додає обробники подій на кореневий елемент DOM – `document`. Це дає можливість збирати інформацію зі всієї веб-сторінки. Кожна нова або наново завантажена сторінка завантажує content script заново, що створює нову сесію з новим

ідентифікатором. Кожна сесія має власний відлік часу у мілісекундах починаючи з нуля. Розбиття на сесії полегшує обробку даних та допомагає виключити можливу аномальність значень під час переходу між сторінками, переключення між вкладками тощо.

Інтервалом отримання інформації було обрано 100 мілісекунд, що є оптимальним для економії трафіку, при цьому точність запису дій користувача залишається достатньою. Коли не наявна активність користувача, тобто він не рухає мишею та не натискає її кнопки, запис призупиняється. За наявності активності кожні 100 мілісекунд зберігаються координати миші, часова мітка сесії та стан миші. Було визначено п’ять станів миші, чотири з яких зберігаються в системі:

- M (move) – стан, коли користувач рухає мишею без нажатих кнопок.
- D (drag) – стан, коли користувач рухає мишею з зажатою кнопкою або кнопками.
- P (press) – стан, коли користувач натиснув кнопку миші. Записується незалежно від інтервалу, в момент натискання.
- R (release) – стан, коли користувач відпустив кнопку миші. Записується незалежно від інтервалу, в момент натискання.
- I (inactive) – стан, коли відсутня активність. Не записується, використовується для контролю виконання скрипту.

За допомогою події `mousemove` розширення постійно підтримує актуальні координати миші, а також, за допомогою збереженого в об’єкті події поля `buttons`, стан нажатих кнопок миші (M або D). Події `mousedown` та `mouseup` створюють окремі записи (зі станами P та R відповідно) незалежно від інтервалу. Таким чином завжди відомий стан миші та координати курсору. Кожну секунду, або коли кількість записів перевищує 10, масив цих записів та ідентифікатор сесії відправляється до background script. В той же час масив у

content script очищується, що дозволяє використовувати мінімальну кількість пам’яті пристрою.

1.2.3 Background script

Background script зберігає ідентифікатор користувача, а також збирає інформацію з усіх сторінок, що виконують в собі content script. Кожну секунду активний content script відсилає масив с записами координат та станів миші, який об’єднується з попередніми даними в цьому скрипті. Ця інформація зберігається з використанням Chrome Extension API, що захищає її від ситуацій, коли в скрипті чи в браузері могла статися помилка. Цей API дозволяє зберігати дані в файловій системі, викликати таймери, отримувати доступ до пристроїв, а також синхронізувати дані з хмарним сховищем.

Коли в background script знаходиться достатньо даних (кількість записів на одній сторінці перевищує 100 або кількість відвіданих сторінок перевищує 5) або проходить 30 секунд з останнього моменту відправки, скрипт збирає всі дані та відправляє їх разом с ідентифікатором сесії та користувача на сервер, який зберігає сесії у файли.

1.2.4 Продуктивність розширення

Оскільки метою цієї роботи є потенційна можливість використовувати її в реальних веб-застосунках, окрему увагу було звернено на кількість ресурсів, що витрачається на отримання та відправку інформації. Зазвичай, JavaScript сценарії на відкритих веб-сторінка працюють в одному потоці, тому будь-яка складна операція може заблокувати весь потік та негативно вплинути на досвід користувача.

Для цього було використано логічне розділення на content script та background script, а також якомога більше обчислень було перенесено саме у background script, тому що він працює в окремому та незалежному від веб-сторінок процесі. Content script, в свою чергу, через відносно низьку частоту відправки даних (один раз в секунду) та запису тільки при наявності подій

браузера, не витрачає багато ресурсів та його вплив важко помітити серед інших процесів та частин сторінки.

У випадку запровадження цієї системи в реальному веб-застосунку, такі міри, як розділення сценаріїв на background та content можуть бути неможливими. В такому випадку є можливість окремо налаштувати частоту отримання та частоту відправки інформації для зменшення навантаження без суттєвої втрати точності.

1.2.5 Сервер отримання даних користувачів

Для зберігання зібраних даних від усіх користувачів було створено простий сервер на Python та Flask. Цей сервер було розміщено на хостингу Hetzner на час дослідження для безперервного доступу та отримання даних. При отриманні даних сервер трансформує кожну сесію у окремий CSV файл, що названо за ідентифікатором сесії. Цей файл розміщується в директорії певного користувача, яку названо за його ідентифікатором (див. рисунок 1.1).

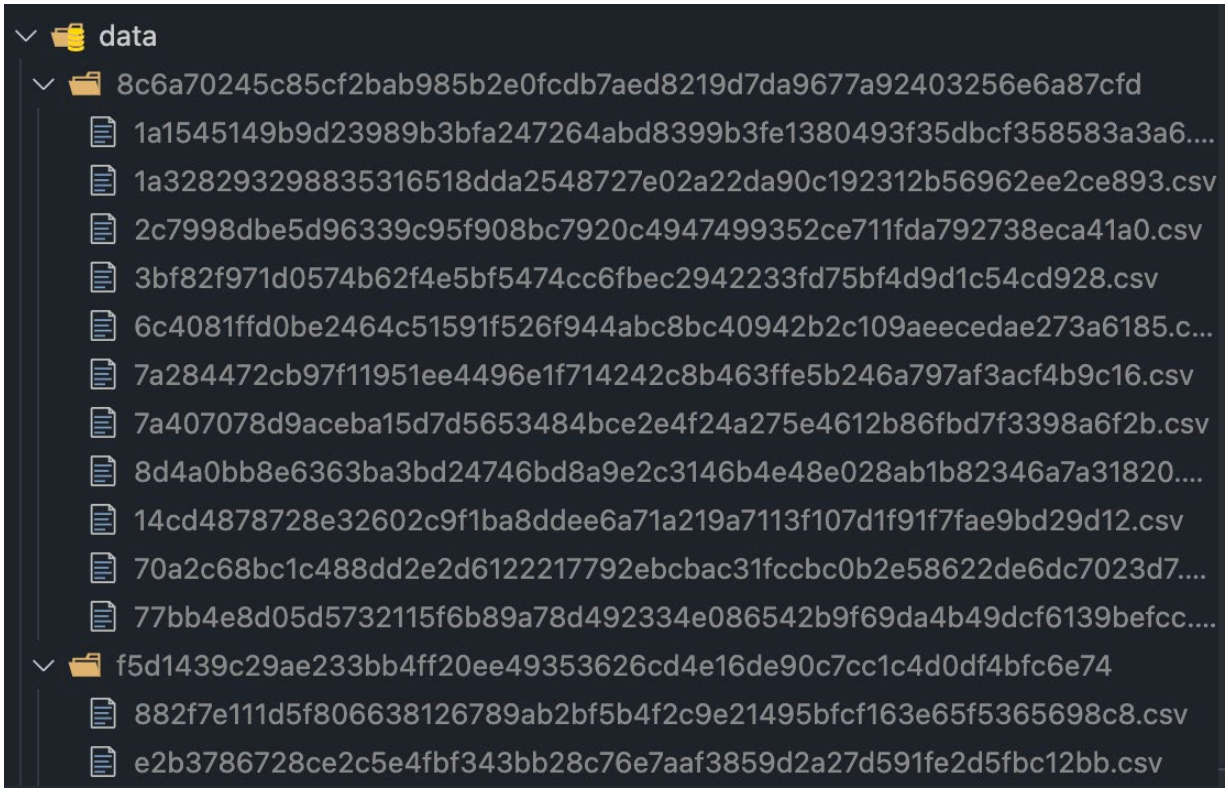

Рисунок 1.1. Структура файлів.

CSV (comma-separated values) – формат, який дозволяє зберігати дані в зручному та лаконічному форматі, а також дозволяє зручно дописувати нові записи в кінець файлу, адже кожен запис представляє собою один рядок. Таким чином, при довгих сесіях або при поверненні користувача на стару вкладку, нова інформація про сесію буде дописана у відповідні файли. Структура, яку використовує сервер, складається с чотирьох полів: часової мітки, стану миші, x-координати та y-координати (див. рисунок 1.2).

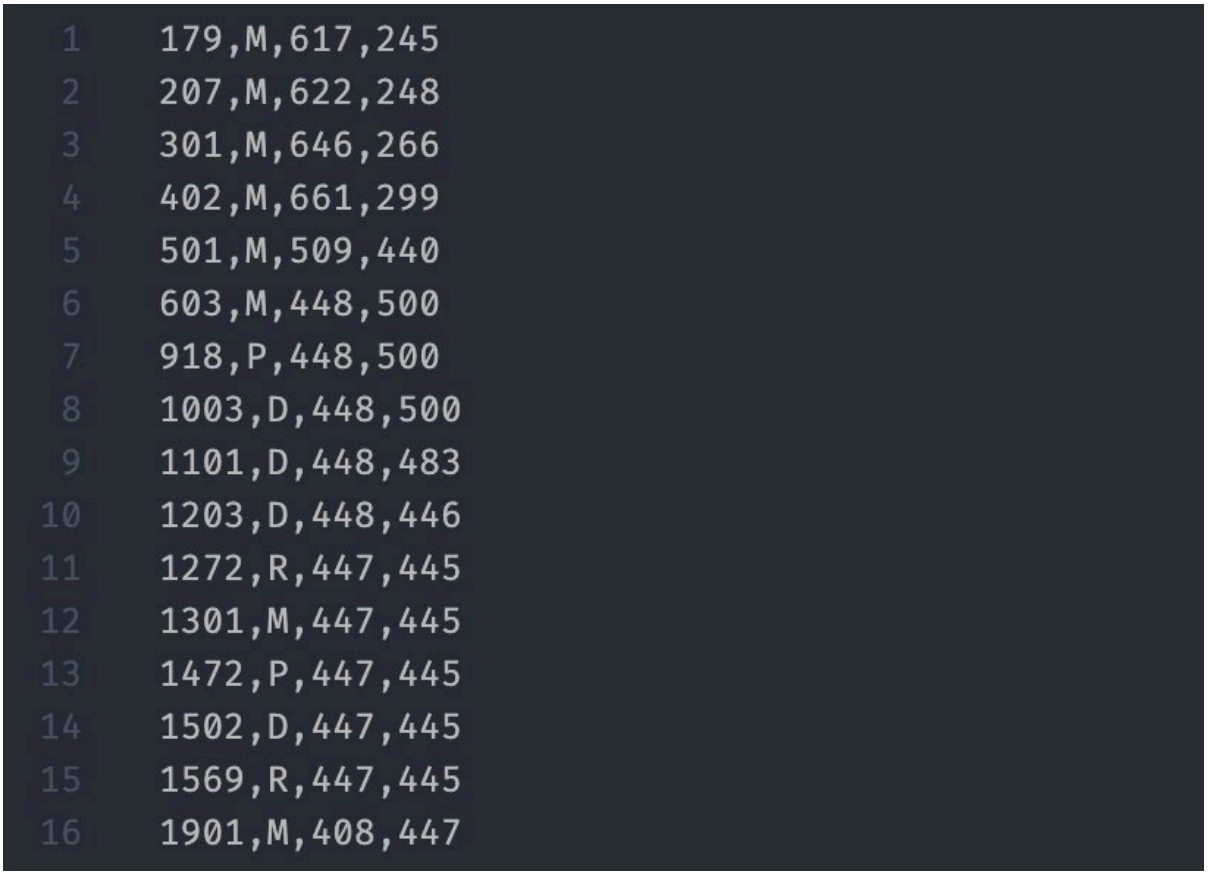

Рисунок 1.2. Приклад CSV-файлу з даними сесії.

Після збереження інформації у файловій системі файли зчитуються окремим сценарієм, який трансформує окремі події з координатами у більш наочний формат дій, який можливо використати для навчання моделі користувача.

1.3 Трансформація даних

Використання отриманих даних напряму для навчання моделі не є ефективним, адже координати та часові мітки самі по собі не вказують на якусь з рис користувача. Тому перед початком ‘навчання моделі необхідним є очищення та трансформація даних, а також виділення певних рис, що характеризують користувача. Усі операції були проведені за допомогою мови Python та її стандартної бібліотеки.

1.3.1 Очищення даних

Перед трансформацією є необхідним видалення дублікатів та надлишкових дій. Для цього було використано поступове порівняння двох послідовних дій на предмет однакових часових міток, типів подій та координат. Через специфіку обробників подій в браузерному середовищі деякі події дублювались с різними типами. Для цього було додану низку умов, що видаляли надлишкові події.

Для того, щоб не враховувати короткі та аномальні дії, всі сесії, що складалися з дій менше трьох секунд, було видалено. Також було видалено елементи, де швидкість між двома точками була в багато разів вища за середню (аналіз показав, що це зазвичай відбувається через зміну фокусу вікна та переносу курсору в іншу позицію).

1.3.2 Відокремлення дій

Для того, щоб обрахувати риси, потрібно розділити безперервний потік подій на окремі частини. Ahmed та Traore [6] рекомендують виділити три види дій користувача: PC, DD, MM (в цій роботі – move, click, drag відповідно). PC (pointer click) – описує клік кнопки миші, може включати в себе певну кількість подій руху миші. DD (drag & drop) – описує перетягування мишею з зажатою

кнопкою, може також включати в себе події руху миші, що передували цій події. MM (mouse move) – описує простий рух мишею без натискань кнопок.

Алгоритм розділення на вищеописані дії, запропонований Antal та Egyed-Zsigmond [7] та дещо модифікований, складається з двох частин.

По-перше, весь потік подій розділяється на послідовності, що закінчуються подією з типом R (release, коли користувач відпускає кнопку миші). Таким чином можна ідентифікувати кінець дії move або drag. Після чого в кожній послідовності перевіряються дії з кінця. Якщо одразу перед останньою R подією відбулася подія P, то це означає, що це дія кліку (click). Якщо перед останньою R подією знаходяться події з типом D, то це означає дію перетягування (drag).

По-друге, кожна з послідовностей розділяється на менші послідовності, якщо в ній знаходиться дві події, між часовими мітками яких є відстань у якнайменш 3 секунди. Таким чином від інших дій відокремлюються дії руху миші (move). Усі дії, що мають менше ніж 5 подій, відфільтровуються, оскільки відокремлену дію довжиною в 500 мілісекунд важко трактувати як ту, що має значення. В результаті для кожного користувача отримано список дій, який підлягає подальшому аналізу та виокремленню рис.

1.4 Виокремлення рис

Наявність координат та часових міток подій дозволяє вивести певні числові характеристики (показники швидкості, відстані, варіанти похідних тощо), що можуть покращити точність моделі. Після аналізу варіантів було вирішено використовувати наступні риси для кожної дії:

- Тип дії (move, drag, click) – 1 риса
- Швидкість (мінімальна, максимальна, медіана, стандартне відхилення) – 4 риси

- Швидкість по осі X (мінімальна, максимальна, медіана, стандартне відхилення) – 4 риси
- Швидкість по осі Y (мінімальна, максимальна, медіана, стандартне відхилення) – 4 риси
- Прискорення (мінімальне, максимальне, медіана, стандартне відхилення) – 4 риси
- Швидкість по осі Y (мінімальна, максимальна, медіана, стандартне відхилення) – 4 риси
- Ривок – похідна прискорення по часу (мінімальний, максимальний, медіана, стандартне відхилення) – 4 риси
- Кутова швидкість (мінімальна, максимальна, медіана, стандартне відхилення) – 4 риси
- Кривина (мінімальна, максимальна, медіана, стандартне відхилення) – 4 риси
- Довжина траєкторії
- Сумарний час дії
- Пряма відстань від початкової до кінцевої точки
- «Прямота» траєкторії (результат ділення прямої відстані на довжину траєкторії)
- Сума всіх кутів між точками
- Тривалість прискорення (перша частина дії, де прискорення більше нуля)

Після обрахування всіх значень вони зберігаються у список для кожної дії, таким чином формуються вектори, що можна використовувати для навчання моделей (див. рисунок 1.3).

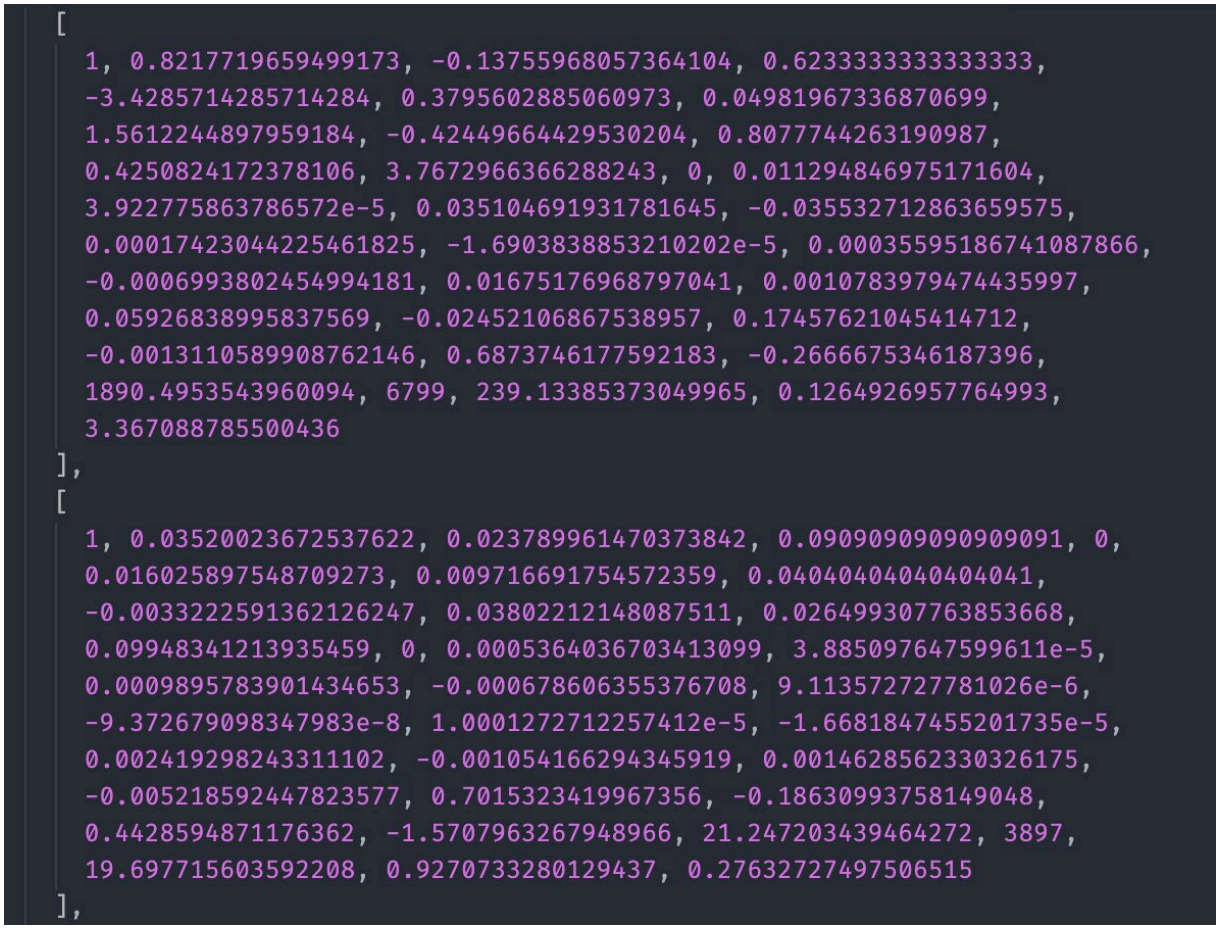

Рисунок 1.3. Приклад двох векторів після виділення рис.

Усього було виокремлено 39 рис, за допомогою яких в подальшому буде виконуватися класифікація. Деякі роботи, зокрема Ahmed та Traore [6], також використовують деякі інші риси, наприклад, напрямок дії. Використовуючи початкову та кінцеву точки визначається вектор напрямку, який знаходиться в одному з восьми секторів, по 45 градусів кожен. В цій роботі це не використовується, тому що вона мала негативний вплив на точність моделі.

1.5 Навчання моделі

Наступним кроком для автентифікації користувача за активністю миші є навчання моделі за трансформованими в Розділі 1 даними.

Задача автентифікації користувача може бути поставлена як задача класифікації з одним класом (унарна класифікація), так і як задача класифікації з двома класами (бінарна класифікація). В цій роботі використовується два класи: user (користувач) та impostor (самозванець).

1.5.1 Унарна класифікація

У випадку унарної класифікації для навчання моделі використовуються лише дані, належні до одного певного класу. Такий метод успішно використовується для розпізнавання обличь, а також автентифікації за рухами миші та натисканнями клавіатури [8]. Також, у випадку певних бізнес-обмежень та на відміну від бінарних методів класифікації, унарна класифікація не потребує доступу до інформації про інших користувачів.

В цій роботі серед унарних методів було розглянуто SVM (Support Vector Machine) [14], а саме SVC – опорно-векторне кластерування, з одним класом. Для навчання кожної моделі користувачів була використана частина даних самих користувачів, для конкретного експерименту було обрано 50% від загальної кількості дій користувача. Інші 50% дій були використані для тестування та обрахування відсотку позитивних та хибно негативних результатів. Для кожної дії SVM може повернути два значення:

- **1 –** у випадку, якщо модель вважає, що дія належить користувачу.
- **-1 –** у випадку, якщо модель вважає, що дія не належить користувачу.

В результаті експериментів SVM з одним класом продемонструвала низьку точність, тому експеримент включає в себе тільки дослідження бінарних класифікаторів.

1.5.2 Бінарна класифікація

За наявності можливості доступу до даних інших користувачів, їх можна використовувати для навчання моделі з двома класами. Такий вид класифікації

має значно більший спектр доступних методів навчання В цій роботі було перевірено декілька методів для бінарної класифікації:

- Random forest
- Багатошаровий перцептрон (multi-layer perceptron)
- Дерево рішень (decision tree)
- Класифікація опорних векторів (support vector classification)

Кожен з цих методів приймає вектор рис та вектор міток для навчання. Для кожної моделі користувача було створено масив, який складається з усіх рис користувача та такої ж кількості випадково обраних рис інших користувачів. Таким чином був отриманий набір даних, в якому в рівній кількості знаходяться риси, помічені як user та риси, помічені як impostor. кожен елемент з якого було помічено як user та impostor відповідно. При реалізації класу impostor було задане значення **0**, а класу user – **1**.

Для навчання з цього масиву зі збереженням співвідношення класів було випадково обрано 50% даних. Після чого для тестування моделі було використано інші 50% даних. На відміну від SVM з одним класом, реалізація бінарних класифікаторів у бібліотеці scikit-learn надає велику кількість функцій оцінки ймовірностей, метрик тощо.

1.5.3 Нормалізація рис

Через специфіку алгоритмів машинного навчання для підвищення точності рекомендовано нормалізувати дані перед навчанням (іноді називається масштабуванням).

Без нормалізації даних деякі функції можуть працювати некоректно, або менш точно, оскільки значення можуть сильно варіюватися. Якщо якась з рис має широкий спектр значень, то вона буде значно впливати на відстань між двома точками в просторі класифікатора, у порівнянні з іншими рисами. Тому,

має сенс нормалізувати спектр кожної риси, щоб кожна риса мало приблизно однаковий вплив на результат.

Для масштабування було використано StandardScaler, що видаляє з кожної риси медіану та масштабує до одиничної дисперсії.

1.5.4 Калібрування класифікатору

Для того, щоб отримати коректні ймовірності, після навчання моделі доцільним є калібрування.

Відкалібрована модель дозволяє не тільки отримати ймовірність належності певного елементу до певного класу, але й також пов’язати це значення з ймовірністю знаходження цього класу. Наприклад, у добре відкаліброваній моделі, якщо ймовірність класу дорівнює приблизно 0.6, то приблизно 60% елементів, що отримали цю ймовірність, будуть мати цей клас. Таким чином у відкаліброваній моделі значення ймовірності класу можна інтерпретувати як рівень впевненості належності елемента до цього класу.

На рис. 1.4 можна побачити показники моделі без калібрування. По осі абсцис знаходиться ймовірність позитивного класу, а по осі ординат – частина елементів, що належить до позитивного класу. Як можна побачити, серед елементів з ймовірністю позитивного класу 0.8 тільки приблизно 50% належать до цього класу. Таким чином стверджувати, що при ймовірності 0.8 шанс, що користувач не є зловмисником дорівнює 80%, не можна. Для оцінки ризику в багатофакторній системі наявність коректних ймовірностей критична, тому доцільно завжди калібрувати модель.

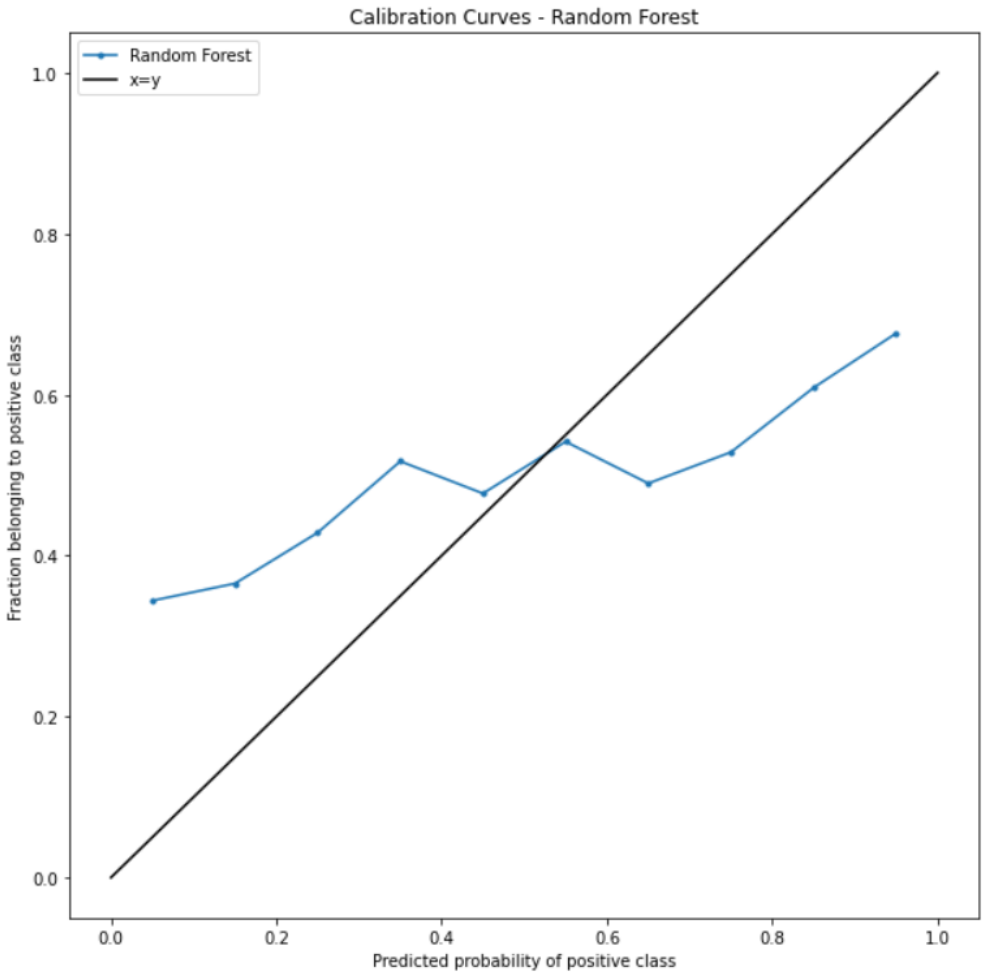

Рисунок 1.4. Приклад погано відкаліброваної моделі.

Для калібрування було обрано CalibratedClassifierCV, що надається бібліотекою scikit-learn. Методом калібрування було обрано ізотонічний, оскільки він надав більш точні результати. Для оцінки було використано 5-кратне перехресне затверджування.

РОЗДІЛ 2. ОГЛЯД ТА РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТУ

2.1 Огляд експерименту

2.1.1 Загальний опис

В дослідженні брало участь 7 користувачів. Отримання даних тривало 3 дні, всього було зібрано біля 10 годин чистої інформації про рухи та дії мишею (без врахування періодів, де активність користувачів була відсутня). Для кожного користувача була навчена окрема модель. Усі користувачі використовували власні пристрої, браузер Google Chrome та розширення для браузеру, що було описано у підрозділі 1.2. Після завершення отримання інформації сервер було вимкнено та видалено, а кожного учасника проінструктовано про видалення розширення з браузеру.

Кожен учасник експерименту використовував браузер як звичайно, без виконання особливих дій та зміни поведінки. Значна кількість робіт [1][2][3][6][7] використовує контрольоване середовище, що позитивно впливає на однорідність набору даних для навчання, а отже й на точність моделей. В цій роботі кожен учасник користувався власним пристроєм та в стандартному для нього оточенні (вдома, в офісі тощо). Неоднорідні характеристики пристроїв, такі як розмір та роздільна здатність екрану, тип миші, оточення тощо – симулюють реальну ситуацію, де потенційний зловмисник, що здобув реквізити доступу іншого користувача, використовує інший пристрій для отримання доступу до системи.

Для імітації зловмисних дій та перевірки точності моделі в кожну модель випадковим чином певну кількість записаних дій інших учасників експерименту. Таким чином кожна модель була протестована проти 6 інших учасників, що

дозволяє отримати результати, які більш точно відображають реальні ситуації, а також знижують вплив окремого учасника при навчанні моделі.

Для обробки даних, навчання моделі, отримання результатів та побудови таблиць використовується мова Python у середовищі Jupyter та такі бібліотеки, як scikit-learn, numpy, та pandas. Scikit-learn надає велику кількість функцій підготовки масивів даних, а також реалізує низку популярних методів машинного навчання. Numpy дає можливість оперувати векторами різної розмірності, обраховувати статистичні значення та зручно обробляти дані. Pandas дозволяє працювати з табличними даними, трансформувати та виводити їх.

2.1.2 Алгоритм оцінювання

Для оцінювання результатів моделей було обраховано такі основні показники:

- FPR (false positive rate) – відсоток хибно позитивних результатів
- FNR (false positive rate) – відсоток хибно негативних результатів
- AUC (area under curve) – площа під кривою ROC (Receiver operating characteristic), див. 2.1.2.1.

Ці показники демонструють ефективність моделі. FPR описує шанс зловмисника бути ідентифікованим як користувач, FNR описує обернену ситуацію – шанс користувача бути ідентифікованим як зловмисник, AUC – показник якості бінарної класифікації.

Оскільки проводиться аналіз точності для різної кількості дій, для цього обирається випадковий набір дій. Тому, для підвищення точності результатів, кожен вибір повторюється 50 разів та використовується середнє арифметичне результатів. Для кожного типу класифікатора були підібрані окремі граничні значення, що розділяють класи, тому що розподіл ймовірностей відрізняється в залежності від типу.

2.1.2.1 ROC-крива

ROC-крива (receiver operating characteristic, з англ. робоча характеристика приймача) – крива, що характеризує якість бінарної класифікації. Оскільки значення ймовірностей, що повертаються бінарним класифікатором, можуть бути дійсним числом від 0 до 1, необхідним є встановлення певного граничного значення, що буде встановлювати межу між двома класами.

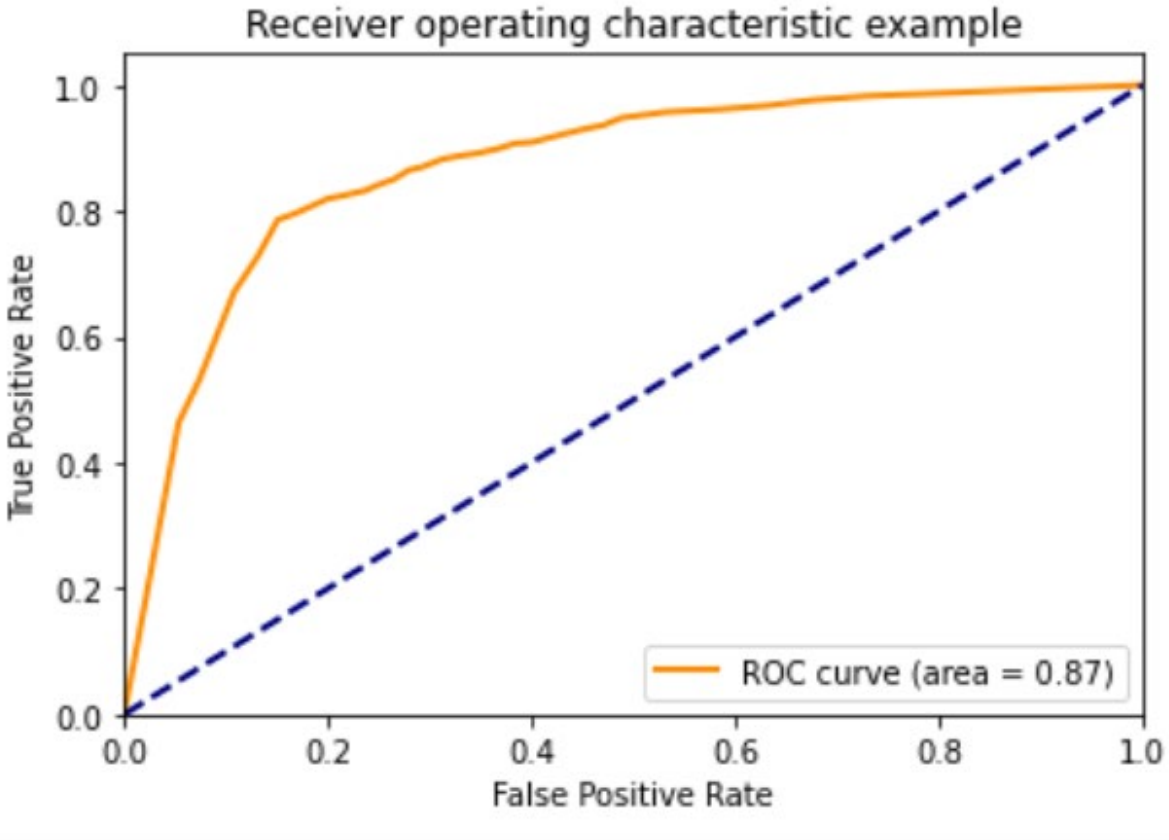

Рисунок 2.1. Приклад ROC-кривої для Decision Tree для користувача з ідентифікатором 45aaee4.

Щоб побудувати ROC-криву, необхідно мати інформацію про два показники: TPR (true positive rate, показник істинно позитивних результатів, також – чутливість) та FPR (false positive rate, показник хибно позитивних результатів, обраховується як `1 - специфічність`). В залежності від

граничного значення ці два показники будуть змінюватися, адже змінюється баланс між класами.

Якщо бінарний класифікатор видає ідеальні результати, то у ROC-просторі це виглядало би як точка (0, 1) у верхньому лівому куту графіку. Це означало би 100% чутливість та 100% специфічність, тобто відсутність неправильних результатів. У випадку неідеального класифікатору це співвідношення виглядає як крива (див. рис. 2.1). Результатом повністю випадкового класифікатору є діагональ на графіку.

Оскільки візуалізація є менш зручним для порівняння методом, для вимірювання наближеності до ідеального значення використовується площа під кривою – area under curve або AUC. Її значення знаходяться в інтервалі від 0 до 1, де 1 – ідеальний класифікатор.

2.2 Результати експерименту

2.2.1 Перевірка на окремих діях

Для кожного методу та для кожного користувача було навчено окрему модель. Після навчання моделей було обраховано точність та оцінку Брайера для кожної з них на випадковому наборі дій.

Оцінка Брайера – показник точності ймовірнісного передбачення. Вона дозволяє оцінити ступінь калібрації моделі, що впливає на точність ймовірностей для оцінки ризику. Ідея калібрації та використані для цього методи було описано в підпункті 1.5.4.

Як можна побачити на рис. 2.1, точність моделей варіюється в рамках 65%, а оцінка Брайера – в рамках 0.25. Ці значення знаходяться далеко від задовільних, але вони описують точність для однієї дії.

	Точність	Оцінка Брайера
Random Forest	0.6681	0.2069
SVM/SVC	0.6346	0.2213
Decision Tree	0.6236	0.2347
MLP	0.6264	0.2755

Таблиця 2.1 - Середня точність та оцінка Брайера для кожного методу навчання для однієї випадкової дії.

Одної дії недостатньо для того, щоб надійно оцінити ймовірність та автентифікувати користувача. Тому, в подальших частинах експерименту проводиться аналіз точності для різної кількості агрегованих дій. Було обрано три основних кількості дій: 10, 50 та 100. У випадку демонстрації високої точності, також було перевірено більшу кількість дій. Важливим аспектом є те, що у користувачів може значно відрізнятися кількість дій, а це впливає на отриману точність моделей, навіть при коректності інших параметрів та методів навчання.

2.2.2 Результати SVM/SVC

Результати роботи SVM залишались незадовільними незалежно від кількості дій, з достатньо високим числом хибно позитивних результатів.

Ідентифікатор	FPR	FNR	AUC
0e3244	0.6233	0.20	0.7872
4375f2	0.5333	0.40	0.6217
45aaee4	0.6700	0.42	0.7065
aad4e1	0.7300	0.18	0.8328
aea835	0.6733	0.22	0.7925
dc83a9	0.6633	0.30	0.7386
f1abe3	0.5700	0.44	0.6219

Таблиця 2.2 - Результати для SVM/SVC при 5 випадкових діях кожного користувача

Ідентифікатор	FPR	FNR	AUC
0e3244	0.3800	0.14	0.7365
4375f2	0.3500	0.34	0.6857
45aaee4	0.4900	0.34	0.7451
aad4e1	0.6400	0.18	0.8124
aea835	0.5667	0.30	0.7591
dc83a9	0.5333	0.38	0.7349
f1abe3	0.4333	0.54	0.6879

Таблиця 2.3 - Результати для SVM/SVC при 10 випадкових діях кожного користувача

Динаміка точності в залежності від кількості дій дуже низька, також для деяких користувачів точність знижується. Було використано різні параметри для навчання, а саме: ядро (лінійне, rbf, сігмоїдальне), гамма-коефіцієнт та коефіцієнт толерантності, який визначає, коли зупинити навчання. Це не змінило в значній мірі точність моделі. Різні комбінації були перевірені на різних кількостях дій.

Ідентифікатор	FPR	FNR	AUC
0e3244	0.0667	0.06	0.7470
4375f2	0.1033	0.58	0.6732
45aaee4	0.3000	0.34	0.7424
aad4e1	0.4900	0.10	0.8072
aea835	0.3700	0.28	0.7490
dc83a9	0.3200	0.50	0.7323
f1abe3	0.1967	0.56	0.7156

Таблиця 2.4 - Результати для SVM/SVC при 50 випадкових діях кожного користувача

Навіть при 50 діях, чого достатньо для точних результатів інших класифікаторів, SVM показує середній FPR у 26.38%, FNR у 34.57%. Тому подальші дослідження за допомогою цього методу не проводились.

2.2.3 Результати MLP

Багатошаровий перцептрон – популярний метод машинного навчання, який продемонстрував більшу точність, аніж SVM/SVC. Перцептрон мав два прихованих шари: 16 та 8 вузлів.

Ідентифікатор	FPR	FNR	AUC
0e3244	0.3233	0.26	0.7135
4375f2	0.3300	0.48	0.7072
45aaee4	0.4300	0.34	0.7452
aad4e1	0.5567	0.30	0.8203
aea835	0.3233	0.30	0.7382
dc83a9	0.4233	0.44	0.7622
f1abe3	0.2667	0.34	0.7138

Таблиця 2.5 - Результати для MLP при 5 випадкових діях кожного користувача

Ідентифікатор	FPR	FNR	AUC
0e3244	0.3667	0.06	0.7598
4375f2	0.3633	0.52	0.6436
45aaee4	0.4000	0.10	0.7533
aad4e1	0.6367	0.20	0.7991
aea835	0.3767	0.16	0.7304
dc83a9	0.5133	0.18	0.7609
f1abe3	0.3100	0.24	0.6985

Таблиця 2.6 - Результати для MLP при 10 випадкових діях кожного користувача

Ідентифікатор	FPR	FNR	AUC
0e3244	0.0433	0.00	0.7449
4375f2	0.1000	0.28	0.6807
45aaee4	0.2133	0.22	0.7388
aad4e1	0.5667	0.04	0.8083
aea835	0.0700	0.18	0.7239
dc83a9	0.3967	0.16	0.7553
f1abe3	0.0633	0.12	0.7056

Таблиця 2.7 - Результати для MLP при 50 випадкових діях кожного користувача

При 50 діях багатошаровий перцептрон показав середній FPR у 20.76%, а FNR – 14.29%. Це помітно краще, ніж SVM/SVC, але все ще недостатньо точно.

2.2.4 Результати Random Forest

Random Forest – метод, де будується велика кількість дерев прийняття рішень, після чого передбачення цих побудованих дерев об’єднується та обирається середнє між ними.

Після перевірки різних параметрів було визначено кількість максимальних рис (max_features) на рівні 3, мінімальну кількість прикладів для листового вузла також на рівні 3, кількість дерев дорівнює 200. Ці параметри дозволили знизити кількість помилок в декілька разів у порівнянні з MLP та SVM/SVC.

Точність була помітно підвищена, але процесс тренування та валідації моделі займає значно багато часу, може досягати багатьох хвилин на відносно невеликій кількості даних учасників. Це може впливати на практичність застосування цієї моделі в реальному часі, адже автентифікація може займати більше часу, ніж необхідно користувачу для виконання необхідних дій.

Ідентифікатор	FPR	FNR	AUC
0e3244	0.2633	0.04	0.8651
4375f2	0.2300	0.18	0.8098
45aaee4	0.4400	0.20	0.8488
aad4e1	0.5567	0.08	0.9059
aea835	0.4933	0.16	0.8617
dc83a9	0.5433	0.22	0.8414
f1abe3	0.4067	0.14	0.8404

Таблиця 2.8 - Результати для Random Forest при 5 випадкових діях кожного користувача

На відміну від попередніх методів навчання, динаміка зниження кількості помилок значно помітніша для Decision Tree. Було також перевірено різні варіанти максимальної глибини дерева, критеріїв оцінки розділення та різні налаштування для листкових вузлів дерева.

Ідентифікатор	FPR	FNR	AUC
0e3244	0.1333	0.10	0.8640
4375f2	0.0750	0.00	0.8458
45aaee4	0.2750	0.20	0.8639
aad4e1	0.3667	0.00	0.9098
aea835	0.3750	0.20	0.8565
dc83a9	0.3833	0.15	0.8540
f1abe3	0.2500	0.25	0.8392

Таблиця 2.9 - Результати для Random Forest при 10 випадкових діях кожного користувача

Ідентифікатор	FPR	FNR	AUC
0e3244	0.0000	0.00	0.8485
4375f2	0.0000	0.20	0.8080
45aaee4	0.0833	0.00	0.8646
aad4e1	0.1167	0.00	0.8920
aea835	0.0333	0.10	0.8574
dc83a9	0.2167	0.00	0.8807
f1abe3	0.0417	0.25	0.8330

Таблиця 2.10 - Результати для Random Forest при 50 випадкових діях кожного користувача

Як можна побачити, для деяких учасників (особливо, тих, де було отримано значно більше інформації) кількість помилкових результатів може дорівнювати нулю.

Ідентифікатор	FPR	FNR	AUC
0e3244	0.0000	0.00	0.8472
4375f2	0.0000	0.00	0.8202
45aaee4	0.0917	0.00	0.8680
aad4e1	0.1333	0.00	0.9009
aea835	0.0250	0.15	0.8473
dc83a9	0.1583	0.00	0.8636
f1abe3	0.0083	0.25	0.8298

Таблиця 2.11 - Результати для Random Forest при 100 випадкових діях кожного користувача

При 50 та 100 діях Random Forest дає низькі показники FPR та FNR та високу точність. Важливо також звернути увагу на таблицю 2.11, де видно, що для деяких користувачів цей метод працює значно точніше, ніж для інших. Різниця між рисами цих користувачів може бути предметом для подальшого дослідження.

2.2.5 Результати Decision Tree

Decision Tree – дерево ухвалення рішень. Вузли дерева представляють собою перевірку атрибута, гілки – результати тесту, а кожен листок – мітку класу. Під час навчання будується дерево, після чого за допомогою його можна передати перелік рис та отримати необхідний клас.

Також, на відміну від багатьох методів машинного навчання, Decision Tree дозволяє візуалізувати свою структуру для подальшого аналізу та пошуку різних залежностей. На рис. 2.2 відображена частина дерева рішень для моделі одного з користувачів, за якою можна прослідкувати за процесом прийняття рішень.

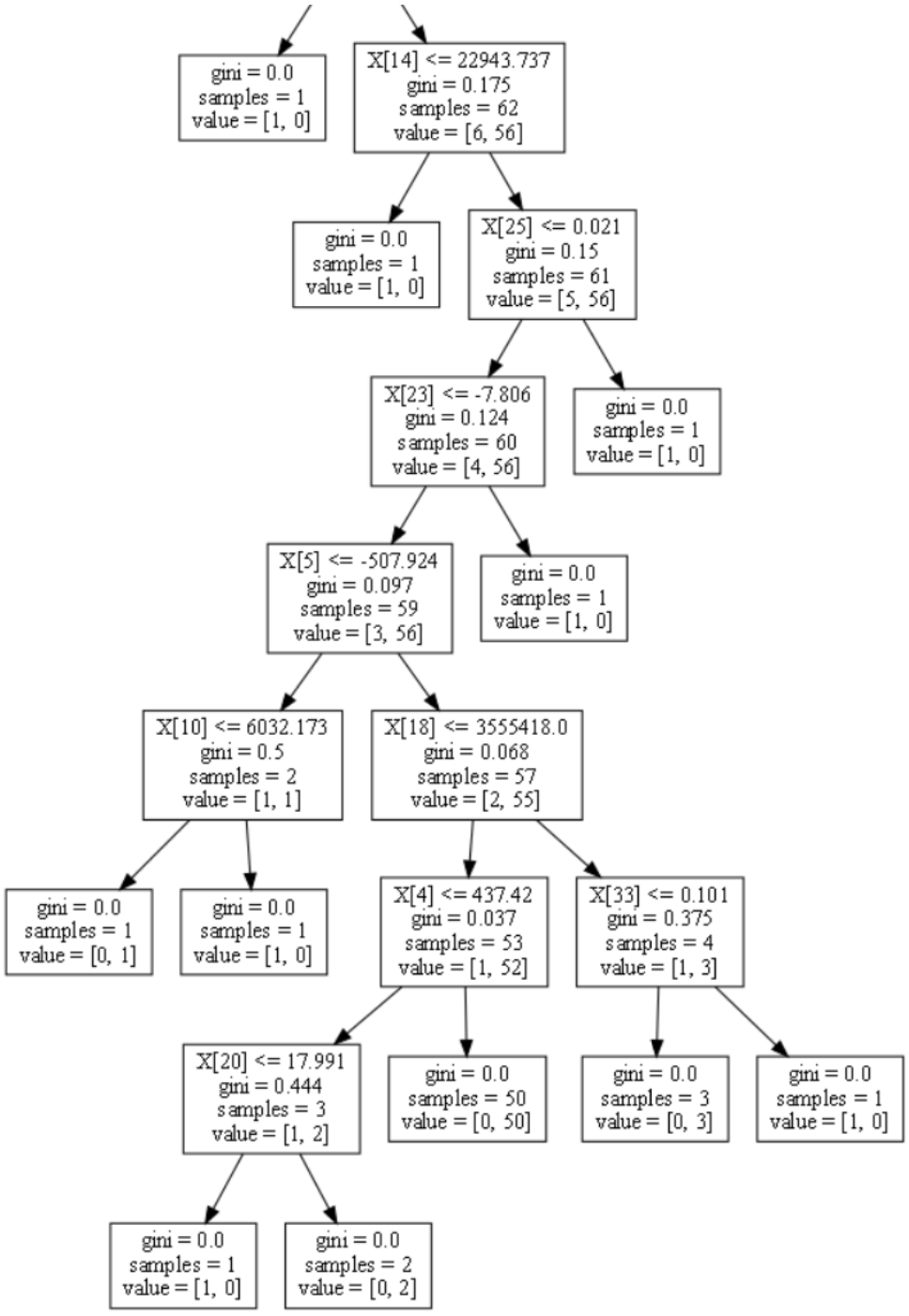

Рисунок 2.2 – Частина дерева прийняття рішень для користувача aad4e1

На рисунку можна побачити, як в залежності від певної риси (певний елемент n масиву X, що позначається як X[n]) модель обирає лівий чи правий вузол для подальшого порівняння або, якщо це листковий вузол – фінальне рішення.

Ідентифікатор	FPR	FNR	AUC
0e3244	0.2533	0.16	0.7915
4375f2	0.2800	0.22	0.7835
45aaee4	0.4267	0.20	0.8516
aad4e1	0.4700	0.04	0.8896
aea835	0.3767	0.16	0.8158
dc83a9	0.4100	0.16	0.8312
f1abe3	0.2367	0.16	0.8281

Таблиця 2.12 - Результати для Decision Tree при 5 випадкових діях кожного користувача

При п’яти діях результати для Decision Tree незначно відрізняються від інших методів навчання. Помітна різниця між результатами проявляється при 10-20 діях та вище.

Ідентифікатор	FPR	FNR	AUC
0e3244	0.1000	0.16	0.8078
4375f2	0.1100	0.20	0.7713
45aaee4	0.3400	0.14	0.8541
aad4e1	0.3633	0.12	0.8678
aea835	0.2400	0.30	0.8148
dc83a9	0.2400	0.12	0.8439
f1abe3	0.1367	0.26	0.7981

Таблиця 2.13 - Результати для Decision Tree при 10 випадкових діях кожного користувача

Ідентифікатор	FPR	FNR	AUC
0e3244	0.0133	0.02	0.8027
4375f2	0.0067	0.08	0.7686
45aaee4	0.1367	0.02	0.8556
aad4e1	0.1767	0.00	0.8770
aea835	0.0600	0.04	0.8312
dc83a9	0.0800	0.02	0.8392
f1abe3	0.0100	0.02	0.8057

Таблиця 2.14 - Результати для Decision Tree при 50 випадкових діях кожного користувача

Ідентифікатор	FPR	FNR	AUC
0e3244	0.0000	0.00	0.8162
4375f2	0.0000	0.04	0.7680
45aaee4	0.0800	0.00	0.8532
aad4e1	0.1033	0.00	0.8768
aea835	0.0000	0.16	0.8177
dc83a9	0.0167	0.00	0.8498
f1abe3	0.0000	0.00	0.8044

Таблиця 2.15 - Результати для Decision Tree при 100 випадкових діях кожного користувача

Decision Tree демонструє дуже високу точність при кількості дій більше за 50. Середній FPR та FNR коливається між 1% та 3%. При підвищенні кількості дій до 200 та вище FPR та FNR знижуються до 0.5% та нижче. Цей показник є достатнім для подальшого вивчення цього класифікатору, збільшення точності

та його практичного використання в реальних застосунках. Незважаючи на те, що Random Forest має схожі результати та теоретично може бути більш точним, для подальшого порівняння було обрано саме Decision Tree, оскільки різниця не є великою, а Decision Tree значно швидке та простіше для аналізу.

Цей же класифікатор використали в своєму дослідженні Pusara та Brodley [2], вони отримали більш точні результати з використанням інших рис користувачів. Порівняння з цими дослідженнями описане в підрозділі 3.2.

РОЗДІЛ 3. ПОРІВНЯННЯ РЕЗУЛЬТАТІВ ТА ПОДАЛЬШЕ ВДОСКОНАЛЕННЯ

3.1 Порівняння результатів різних методів навчання

Всього було використано чотири види класифікаторів: SVM/SVC, MLP, Random Forest та Decision Tree. Кожна дія користувача в середньому триває від трьох до п’яти секунд. Тому, в залежності від обраних кількостей дій є відповідна тривалість сесії користувача:

- 10 дій – 30-50 секунд активності
- 50 дій – 3-4 хвилині активності
- 100 дій – 5-10 хвилин активності

Більшість моделей навіть при використанні більше 10 хвилин активності продукували 10-20% помилок, при цьому тривалість валідації могла займати до хвилини на кожного користувача. Для порівняння, в деяких інших роботах [7] було достатньо одної хвилини для отримання значно більшої точності.

Серед всіх обраних методів найточнішим виявився Decision Tree. Для практичного застосування як додаткового фактору автентифікації або вимірювання ризику, бажано, щоб показники FNR та FPR були нижче за 3-5%. Decision Tree досягає FNR та FPR нижче 1% при аналізу 5 хвилин активності та більше. При цьому рішення приймається за 1-3 секунди.

У випадку використання цього методу автентифікації як основного, Європейський стандарт для біометрії в комерційному середовищі вимагає максимум FPR 0.001% та максимум FRR 1% [8]. Для цього необхідно покращити модель та методи виокремлення рис та трансформації даних.

3.2 Порівняння з іншими дослідженнями

В розглянутих дослідженнях результати значно варіюються, FPR та FNR можуть бути від 1-2% до 26%. Отримані результати дуже залежать від наявності великого об’єму даних, умов дослідження, методів навчання моделей тощо.

Такі роботи, як Aksari та Artuner [9] або Hashia [10] досягали FPR та FNR у 5.9% та 15% відповідно, при цьому було використано однорідні пристрої, контрольоване середовище та встановлений порядок дій для учасників.

В той же час дослідження Ahmed та Traore [6] та Pusara та Brodley [2] досягли 2.46% та 0.43% FPR в неконтрольованому середовищі, схожим с умовами цієї роботи. В цій роботі було отримано FPR та FNR біля 1% після аналізу 300 секунд активності, коли Ahmed та Traore знадобилось 1033 секунди для досягнення менш точних показників хибних результатів.

Для навчання моделей в різних дослідженнях було використано як схожі, так і різні види рис користувачів. Більшість робіт об’єднує використання таких рис: швидкість вказівного пристрою, прискорення вказівного пристрою, напрямок руху та час на виконання дії. В деяких роботах було використано кількість та час натискання кнопок миші, поведінку колеса миші тощо.

Також треба зазначити, що в інших роботах, як правило, було використано отримання інформації на рівні операційної системи, що дозволяє зібрати більшу кількість дій та зменшує кількість аномальних даних, які можуть з’являтися в браузерному середовищі. Таким чином, можна стверджувати, що отримані в цій роботі результати можуть конкурувати з результатами попередній досліджень, адже отримана висока точність автентифікації за менший час, ніж в інших роботах.

3.3 Обмеження та можливі недоліки експерименту

Результатом цього дослідження є достатньо точна модель, що може швидко та з низьким показником помилок визначити користувача. Але точність цього дослідження має великий простір для покращення.

3.3.1 Кількість учасників

Кількість учасників була дуже обмежена через специфіку отримання інформації та обставин самого дослідження. Тому результати експериментів автентифікації можуть бути занадто високими або низькими. Оскільки кожен учасник використовував власний пристрій, то низька кількість учасників підвищує загальну гетерогенність середовищ та обставин, що може як підвищити, так и понизити точність.

Для подальших досліджень рекомендовано збільшити кількість учасників та проводити

3.3.2 Метод отримання інформації

Браузерне розширення та браузерне середовище в цілому не має жодних гарантій стабільного виконання сценаріїв. Оскільки в будь-який час браузер може перезавантажити будь-яку сторінку або сценарій, то неможливо гарантувати безперебійну передачу та зберігання інформації про рухи мишею. Це зменшує загальну кількість даних та підвищує кількість більш коротких відрізків подій, виділення рис з яких менш ефективне.

Також, браузер не гарантує, що задані інтервали часу будуть точно та рівномірно виконуватися. Наприклад, функція `window.setInterval` може призупинитися або затриматися через інші процеси. Таким чином точність похідних від часу даних (а майже всі риси в той чи іншій мірі залежать від часу) може також втрачатися, що має прямий вплив на точність моделей.

Координати миші також можуть створювати аномальні записи, якщо курсор покидає рамки документу (наприклад до поля вводу адресу чи іншого вікна) та повертається в іншу частину документу.

3.3.3 Метод трансформації даних та виділення рис

Логіка розділення масиву подій на окремі дії визначає якість подальшого обчислення рис. Якщо результатом розділення будуть занадто довгі, занадто короткі або неоднорідні дії, це буде впливати на обрахування показників швидкості, прискорення, відстані тощо.

Кількість та тип використаних рис для навчання є постійною проблемою машинного навчання, адже саме від цього залежить, наскільки точною буде отримана модель. Незважаючи на те, що було обрано риси, ефективність яких була вже раніше доведена в інших роботах, завжди існує простір для покращення. Різні методи отримання та трансформації даних можуть позитивно впливати на важливість певних рис та негативно – на інші. Дослідження інших робіт [1][2] показало, що на точність позитивно впливає вимір показників часу натискань кнопок миші: час одного кліку, час подвійного кліку, час тримання кнопки зажатою тощо.

3.4 Простір для вдосконалення

Отримані результати є достатньо точними у порівнянні з іншими роботами, але не є достатніми для запровадження як окремий метод автентифікації в комерційних системах. Як було вказано в підрозділі 3.1, кількість хибно позитивних результатів має бути у від 500 до 1000 разів менша, ніж отримана в цій роботі. Такої точності не досягла жодна з розглянутих робіт, тому необхідно продовжувати вдосконалення алгоритмів та методів класифікації користувачів.

Серед додаткових методів можливо використати аналіз натискань клавіатури. Як показують дослідження, такі як S. Salmeron-Majadas et al. [11], поведінка натискань клавіатури може дуже точно характеризувати користувача, тому є доцільним проводити подальші дослідження з врахуванням цієї інформації.

3.5. Можливості застосування в індустрії

З поточними показниками точності та рівнем помилок, цей метод можна використовувати як додатковий фактор у вже активних системах для оцінки ризику та більш точного прийняття рішень.

Перспективним прикладом є банківська система. Більшість систем вже аналізують більшість доступної їм інформації, такої як IP-адреса, браузер, кількість дій різних типів, рахунки для переказів, суми переказів тощо. Сучасні мобільні застосунки також використовують такі показники, як тип пристрою, проведений час в застосунку та навіть статистику заряду акумулятора мобільного пристрою. Будь-який ризик зловмисного втручання має враховуватися, тому що наслідки можуть бути критичними для користувача та компанії, що володіє системою.

В залежності від рівню впевненості у діях користувача, системи додають певні перевірки (додаткове введення паролю, СМС-підтвердження, CAPTCHA, підтвердження телефонним дзвінком тощо), а також можуть обмежувати функціонал, максимальні суми переказів, кредитні ліміти тощо. Активність вказівного пристрою може бути корисним додатком до вказаних вище факторів, що може підвищити безпечність користувача.

Також, цей метод можна використовувати у двохфакторній автентифікації. Наприклад, такі компанії, як AliExpress, після введення паролю вимагають

натиснути та протягнути елемент зліва направо (див. рис. 3.1). Після чого аналізується динаміка цієї дії та вирішується, чи надавати доступ користувачу до системи. У конкретному випадку цього магазину це використовується для захисту від автоматизованого використання веб-застосунку (різні сценарії збору даних з сайту, боти-користувачі). Таким чином вони значно зменшують кількість зловмисних автоматизованих дій, при цьому не використовуючи загальноприйняті методи як CAPTCHA, де користувачу необхідно розпізнати викривлений текст чи знайти певний об’єкт серед певної кількості фотографій.

Рисунок 3.1 – Додатковий фактор автентифікації в веб-застосунку онлайн-магазину AliExpress

Але можливим є використання схожих перевірок для автентифікації користувача. Наприклад, задача декілька разів перетягнути об’єкт в декілька різних точок може бути достатньою для аналізу поведінки користувача та перевірки, чи це він.

3.6 Використання в мобільних пристроях

Все більше користувачів використовують мобільні пристрої як основні. Тому дуже актуальним може бути автентифікація саме за допомогою активності на мобільних пристроях.

Більшість таких пристроїв використовує сенсорні екрани, тому потенційним методом може бути аналіз рухів пальцями. Веб-застосунки мають доступ до схожих подій для сенсорних екранів (Pointer Events), що може дозволити аналізувати рухи під час прокручування сторінки, збільшення зображень, натискання елементів тощо. Потенційна точність моделі може бути вищою, оскільки кількість різних дій, які виконує користувач на пристроях з сенсорним екраном, вища.

Також, специфіка сенсорних екранів дозволяє використовувати відносні позиції для підвищення точності [15]. Наприклад, різні користувачі, через різну форму рук та різне сприйняття інформації, постійно розміщують палець в схожому місці екрану при прокручуванні сторінки. Також різні користувачі натискають елементи та посилання в різних частинах цього елементу чи посилання. Це дозволяє виявити більше різних рис поведінки конкретного користувача та підвищити точність автентифікації.

ВИСНОВКИ

В цій роботі було розглянуто проблему автентифікації користувача за допомогою аналізу активності комп’ютерної миші та трекпаду. Було розглянуто попередні роботи за цією темою, а також проаналізовано та визначено, які саме показники необхідно використовувати для аналізу поведінки користувача та навчання моделі.

В практичній частині цієї роботи було створене браузерне розширення для отримання активності миші, сервер для збору цих даних та застосунок, який трансформує ці дані та навчає моделі. За допомогою отриманої інформації для кожного користувача було створено декілька моделей з різними методами класифікації та проведено експеримент, де було визначено та порівняно різні показники цих моделей.

Результатом роботи є модель, що з показниками помилок менш ніж 1%, може автентифікувати користувача після 5 хвилин активності комп’ютерної миші. Були розглянуті потенційні недоліки та можливості вдосконалення системи.

Задача автентифікації користувача за допомогою активності вказівного пристрою є актуальною, оскільки можливо досягнути високої точності та використовувати це як додатковий чи окремий метод забезпечення безпеки користування веб-застосунками.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. C. Shen, Z. Cai, X. Guan, Y. Du and R. A. Maxion, "User Authentication Through Mouse Dynamics," in IEEE Transactions on Information Forensics and Security, vol. 8, no. 1, pp. 16-30, Jan. 2013, doi: 10.1109/TIFS.2012.2223677.
2. Maja Pusara and Carla E. Brodley. 2004. User re-authentication via mouse movements.
3. Á., Kovács, L., Kurics, T., Windhager-Pokol, E. (2016). *Balabit Mouse Dynamics Challenge data set*. Available at: https://github.com/balabit/Mouse-Dynamics-Challenge
4. Pointer Events, Level 2. W3C World Wide Web Consortium Recommendation 04 April 2019. (https://www.w3.org/TR/2019/REC-pointerevents2-20190404/)
5. UI Events, W3C Working Draft. W3C World Wide Web Consortium Recommendation 30 May 2019. (https://www.w3.org/TR/2019/WD-uievents-20190530/)
6. Ahmed, A. A., & Traore, I. (2010). Mouse Dynamics Biometric Technology. In L. Wang, & X. Geng (Ed.), Behavioral Biometrics for Human Identification: Intelligent Applications (pp. 207-223). IGI Global. http://doi:10.4018/978-1-60566-725-6.ch010
7. Antal, Margit & Egyed-Zsigmond, Elod. (2018). Intrusion Detection Using Mouse Dynamics.
8. European Standard EN 50133-1: Alarm Systems. Access Control Systems for Use in Security Applications, Part 1: System requirements, Standard Number EN 50133-1:1996/A1:2002, Technical Body CLC/TC 79, 2002, CENELEC, European Committee for Electrotechnical Standardization (CENELEC).
9. Aksari, Y., & Artuner, H. (2009). Active authentication by mouse movements. 2009 24th International Symposium on Computer and Information Sciences, 571-574.
10. Shivani Hashia (2004). Authentication by Mouse Movements CS 297 Report.

11. S. Salmeron-Majadas, R. S. Baker, O. C. Santos, and J. G. Boticario, "A Machine Learning Approach to Leverage Individual Keyboard and Mouse Interaction Behavior from Multiple Users in Real-World Learning Scenarios," in IEEE Access, vol. 6, pp. 39154-39179, 2018, doi: 10.1109/ACCESS.2018.2854966.
12. Soumik Mondal, Patrick Bours (2013). Continuous authentication using mouse dynamics.
13. «How to make your chrome extension access webpage?». https://waitingphoenix.com/how-to-make-your-chrome-extension-access-webpage/
14. Cristianini, Nello; Shawe-Taylor, John (2000). An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press. ISBN 0-521-78019-5.
15. Margit Antal, László Zsolt Szabó, Biometric Authentication Based on Touchscreen Swipe Patterns, Procedia Technology, Volume 22, 2016, Pages 862-869, ISSN 2212-0173.