**Реферат**

Отчет 43 страниц, 6 частей, 15 источников;

АУТЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЯ, ДВУХФАКТОРНАЯ АУТЕНТИФИКАЦИЯ, ДВИЖЕНИЕ МЫШИ, КУРСОРНЫЙ ПОЧЕРК.

**Объект исследования:** аутентификация пользователя.

**Цель работы:** оценка эффективности применения курсорного почерка как второго фактора аутентификации пользователя.

**Методы** **исследования:**

1. Анализ научной литературы;
2. Математические и графические методы обработки данных;
3. Расчет параметров движения курсора;
4. Построение модели аутентификации пользователя по курсорному почерку при помощи методов машинного обучения.

**Результаты работы:** применение анализа курсорного почерка является достаточно эффективным методом аутентификации пользователя, так как результаты испытания модели, примененной в данном исследовании, показали точность: 0% ошибку I рода и 10% ошибку II рода для движения *mouse\_click* и 0% ошибку I рода и 20% ошибку II рода для движения *mouse\_move*.

**ОГЛАВЛЕНИЕ**

[ОПРЕДЕЛЕНИЯ 7](#_Toc531692265)

[ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ 9](#_Toc531692266)

[ВВЕДЕНИЕ 10](#_Toc531692267)

[ОСНОВНАЯ ЧАСТЬ 13](#_Toc531692268)

[1 Состояние вопроса и обзор существующих разработок 13](#_Toc531692269)

[1.1 Биометрическая система поведения, основанная на взаимодействии человека с компьютером 13](#_Toc531692270)

[1.2 Новая технология, основанная на динамике мыши 13](#_Toc531692271)

[1.3 Повторная аутентификация пользователя с помощью движений мыши 16](#_Toc531692272)

[1.4 Разработка и исследование компьютерной модели динамики системы «пользователь-мышь» 17](#_Toc531692273)

[1.5 Предлагаемый подход 18](#_Toc531692274)

[2 Написание программного продукта для сбора данных 19](#_Toc531692275)

[2.1 Написание программы для записи действий мыши 19](#_Toc531692276)

[2.2 Написание программного модуля для установки программы записи движения мыши 20](#_Toc531692277)

[3 Написание программного продукта для обработки данных 21](#_Toc531692278)

[3.1 Обработка данных 21](#_Toc531692279)

[3.2 Визуализация 23](#_Toc531692280)

[3.3 Отчистка данных и выявления особенностей движений 24](#_Toc531692281)

[3.3.1 Отчистка данных для движения типа MC 27](#_Toc531692282)

[3.3.2 Отчистка данных для движения типа MM 30](#_Toc531692283)

[4 Построение модели многоклассовой классификации пользователей и оценка ее эффективности 35](#_Toc531692284)

[5 Построение модели одноклассовой классификации пользователей и оценка ее эффективности 38](#_Toc531692285)

[6 Построение модели одноклассовой классификации пользователей и оценка ее эффективности 41](#_Toc531692286)

[ЗАКЛЮЧЕНИЕ 43](#_Toc531692287)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 45](#_Toc531692288)

# **ОПРЕДЕЛЕНИЯ**

Курсорный почерк – характеризует уникальный стиль работы пользователя с манипулятором «мышь» и отражает индивидуальные особенности. Курсорный почерк отображается областью многомерного пространства, координатами которого являются параметры динамики компьютерной «мыши»;

Метод k-ближайших соседей – метрический алгоритм для автоматической классификации объектов или регрессии;

Случайный лес – алгоритм машинного обучения, заключающийся в использовании ансамбля решающих деревьев;

Токен (USB-ключ, смарт–карта) – физическое компактное устройство, используемое двухфакторной аутентификации, часто используется для удалённого доступа;

Инсайдер – член какой-либо группы людей, который в силу своего служебного положения имеет доступ к конфиденциальной информации;

GitHub – веб-сервис для размещения IT–проектов, а также их совместной разработки с поддержкой системы контроля версий Git;

ОS Windows - семейство коммерческих операционных систем корпорации Microsoft;

Python – высокоуровневый язык программирования общего назначения, ориентированный на быструю разработку и читаемости кода;

Scikit-learn - это компьютерная библиотека для Python, которая предоставляет инструменты для анализа данных с заострением внимания на компьютерном обучении.

# **ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

OC – операционная система;

СКУД – система контроля и управления доступом;

ИC – информационная система;

ЭВМ – электронно-вычислительная машина;

ROC – кривая (receiver operating characteristic, рабочая характеристика приёмника) – график, позволяющий оценить качество бинарной классификации;

AUC – (Area Under Curve, площадь под ROC-кривой) – площадь, ограниченная ROC-кривой и осью доли ложных положительных классификаций;

FRR – ложного отказа в доступе (False Rejection Rate — доступ запрещён зарегистрированному в системе человеку);

FAR – коэффициентов ложного пропуска (False Acceptance Rate — система предоставляет доступ незарегистрированному пользователю).

**ВВЕДЕНИЕ**

В современном обществе главным ресурсом является информация.   
Во всем мире для любых действий давно уже используется компьютер. Поэтому вопрос защиты от несанкционированного доступа к рабочему месту, как никогда, актуален. Любое нарушение конфиденциальности или целостности информации наносит финансовый ущерб ее владельцу.

Подобные потери обходятся мировой экономике в миллиарды долларов. По данным компании InfoWatch в около 60,5% случаев виновниками утечек является внутренний нарушитель [1]. В связи с этим есть смысл проводить мониторинг действий пользователя за компьютером, осуществляя непрерывную аутентификацию и отслеживать опасные действия.

Стоит отметить, что аутентификация пользователя только при входе в информационную систему является недостаточной, т.к. во время работы пользователя может произойти его подмена. На данный момент основным средством аутентификации является использование пароля или токена. При использовании таких СКУД были выявлены их недостатки. Пароли похищаются, а смарт–карты теряются или подделываются. Очень часто встречается ситуация, когда пользователь оставляет свой пароль, написанный на листочке бумаги, на рабочем месте или просто передает свой пароль третьему лицу. Например, доверяя ему, просит произвести какую-то операцию.

Биометрический способ аутентификации используется реже, однако такой способ является более надежным (при правильной реализации), так как позволяет привязывать аутентификатор к личности человека. Таким образом исключается фактор потери, передачи или копирования средств аутентификации, что сильно усложняет задачу для мошенников.

Основная проблема создания биометрических систем аутентификации состоит в выделении информативных признаков. Наиболее популярными из них являются отпечатки пальцев, сетчатка и радужная оболочка. Однако, как уже было сказано ранее, существует множество способов хищения данных признаков незаметно для владельца, в особенности отпечатков пальцев. И стоимость реализации биометрической СКУД основанной на сканировании радужной оболочки или сетчатки глаза очень высока.

Способ биометрической аутентификации по анализу взаимодействия человека с компьютером посредствам движения курсора является достаточно новым. И на данный момент на рынке не представлено ни одной системы аутентификации личности, основанных на таком анализе взаимодействия.

Исходя из этого актуальность задачи непрерывной во времени (двухфакторной) аутентификации пользователя в информационную систему имеет место быть.

В рамках работ как отечественных, так и зарубежных ученых, например, Диденко С.М. и Бхатнагар М. [2–4], рассматривается подход для разграничения доступа в компьютерной системе на основе особенностей работы пользователя с мышью. Данные, полученные при работе пользователя с мышью обрабатываются нейронной сетью и используются для непрерывной во времени аутентификации пользователя (в процессе его работы за компьютером), получившего доступ к компьютерной системе. Это возможно в связи с тем, что у каждого пользователя существует собственная манера управления мышью.

В рамках данного исследования был рассмотрен аналогичный подход, основанный на особенностях работы пользователей с мышью, был применен метод используемый в работе [4], но с применением новых технологий машинного обучения. Учитывались наиболее перспективные стороны исследований [3,6,8,9] в целях аутентификации пользователя.

**Объектом** данного исследования является аутентификация пользователя.  **Предметом –** курсорный почерк (движение компьютерной мыши). **Целью** данного исследования является оценка эффективности применения курсорного почерка как второго фактора аутентификации пользователя. Цель определила следующие **задачи**:

1. Написать программный продукт для обработки данных;
2. Написать программный продукт для аутентификации пользователя;
3. Провести оценку эффективности модели аутентификации пользователя по курсорному почерку.

**Гипотеза** исследования состоит в том, что применение анализа курсорного почерка является эффективным методом вспомогательной аутентификации пользователя. В ходе работы применялись следующие **методы и виды деятельности:**

1. Анализ научной литературы;
2. Математические методы обработки данных;
3. Расчет параметров движения курсора;
4. Построение модели аутентификации пользователя по курсорному почерку при помощи методов машинного обучения.

# **ОСНОВНАЯ ЧАСТЬ**

1. **Состояние вопроса и обзор существующих разработок**
   1. **Биометрическая система поведения, основанная на взаимодействии человека с компьютером**

Х. Гамбоа и А. Фред [6] предложили проверить пользователя на основе его взаимодействия с игрой на память. Пользователю требуется определить соответствие одинаковых плиток. В ходе движения мыши между плитками выявлялись отдельные движения, называемые тактами. Такт определяется, как совокупность пройденных точек от одного клика к другому. Рассчитывались характеристики каждого такта, такие как кривизна и скорость движения. Набор из одного или нескольких тактов использовался для идентификации пользователя. Процедура обучения использовала максимальное правдоподобие с различными распределениями, такими как Вейбулла и Парзана [7]. Оценку проводили с использованием 50 пользователей с различным количеством выявленных тактов, имеющих среднюю продолжительность 1 секунды. Равные частоты ошибок (equal error rate) 0,007 и 0,002 были достигнуты в течение 100 и 200 ударов мыши, соответственно.

* 1. **Новая биометрическая технология, основанная   
     на динамике мыши**

А.А.Е. Ахмед и И. Траоре [8] предложили мониторинг активности мыши пользователей, когда они выполняют свои собственные повседневные задачи. Характеристики движений были извлечены и объединены в гистограммы, которые были использованы для идентификации пользователей.

Были определены четыре вида действий:

* Mouse-Move (ММ) - движение между двумя точками;
* Drag-and-drop (DD) - действие состоит из следующей последовательности событий: кнопки мыши вниз, движение, а затем кнопки мыши вверх;
* Point and Click (PC) - движение между двумя точками с последующим щелчком;
* Silence - нет движения.

Каждое действие описывается свойствами, такими как длительность, пройденное расстояние и направление движения. Направление движения определяется так: все возможные направления движения поделены на 8 секторов, из которых каждый занимает 45 градусов, как показано   
на рисунке 1.

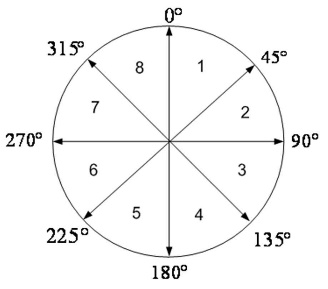


Рисунок 1 - 8 равных секторов окружности;

каждый сектор занимает 45 градусов

Примеры собранных действий представлены в таблице 1.

Таблица 1 - Пример собранных действий

|  |  |  |  |
| --- | --- | --- | --- |
| Тип действия | Расстояние  (в пикселях) | Время  (в секунды) | Направление |
| MM | 50 | 1 | 3 |
| PC | 237 | 3 | 4 |
| PC | 80 | 2 | 2 |
| Silence | - | 2 | - |

Сессия определяется как последовательность действий мыши, выполняемых пользователем. Последовательность ограничена в заранее определенным количеством действий и за определенный период времени. Пользователь характеризуется набором из 7 гистограмм, которые построены из данных сеанса пользователя. Для формирования гистограммы, данные усреднены по всей сессии.

Используемые виды гистограмм:

1. Пройденное расстояние - Распределение пройденного пути для каждого типа движения.

2. Тип действия - Относительная частота движений ММ, DD и PC в рамках сеанса.

3. Направление движения - Отношение расстояния ко времени, выполняемых в каждом из восьми направлений.

4. Средняя скорость движения - Средняя скорость по всему движению, выполняемом в каждом из восьми направлений.

5. Средняя скорость движения по отдельным видам движения - Средняя скорость выполнения MM, DD и PC.

6. Скорость движения по сравнению с пройденным расстоянием - Средняя скорости деленная на расстояние.

7. Время движения - Распределение времени для выполнения движений.

Гистограммы используются для построения вектора из 39 признаков, характеризующих каждый сеанс каждого пользователя.

Для каждого пользователя была построена нейронная сеть на основе векторов признаков, взятых из различных гистограмм. Нейронная сеть была обучена с помощью алгоритма обратного распространения ошибки. Обучение состояло из 5 сеансов - каждый из которых содержали 2000 движений (~ 13,55 минут). Этот эксперимент дает 2.4614% FRR и от 2.4649% FAR. Более короткое время (около 4 минут) получают результаты менее чем 24% FRR и 4,6% FAR. Таким образом, для построения точных гистограмм, требуется значительное количество движений мыши, контролируемое в течение относительно длительного периода времени.

* 1. **Повторная аутентификация пользователя с помощью движений мыши**

M. Пусара, и C. E. Бродлей [9] предложил схему проверки пользователя на основе движений мыши, в то время как участники просматривают определенный набор веб-страниц с помощью веб-браузера. Такие характеристики, как среднее и стандартное отклонение, третий момент расстояния, угла и скорости были рассчитаны из последовательности N движений. Были проведены три основных оценки: цель первой была проверить разницу в поведении между каждой парой пользователей. Результаты показали, что относительно большое количество пользователей можно отличить друг от друга. Во второй оценке, проводилась идентификация каждого пользователя х из множества оставшихся пользователей. Было построено решающее дерево для каждого пользователя х, в среднем были достигнуты 27,5% FAR и 3,06% FRR. Третья оценка была аналогична второй, но использовала только 11 (из 18) пользователей, а также был применен сглаживающий фильтр к данным. Были достигнуты 0,43% FAR и 1,75%.

* 1. **Разработка и исследование компьютерной модели динамики системы «пользователь-мышь»**

Диденко С.М. [3] предложил непрерывную во времени аутентификацию пользователя с помощью движения мыши. Выявлялись отдельные движения с кликом на конце. Для каждого движения рассчитывалось 7 параметров: время движения, расстояние, средняя скорость, угол начала движения, время остановки перед нажатием и время нажатия на клавишу, 7-ым параметром был идентификатор приложения (окна) в котором работал пользователь. Для каждого пользователя была построена радиальная нейросеть рисунок 2, входными значениями были параметры движений.

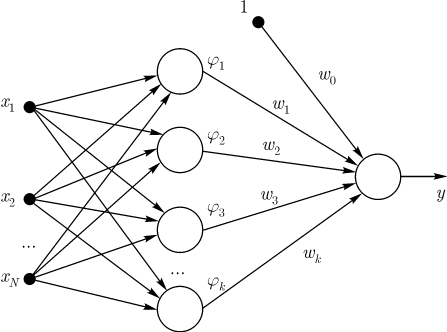


Рисунок 2 – Архитектура нейронной сети

Результаты показали, что разработанный метод идентификации системы «пользователь-мышь», основанный на применении нейронных сетей, в процессе экспериментов позволил распознать 93 пользователей по критерию «свой-чужой» с вероятностью от 0,86 (при количестве 1324 измерений) до 0,95 (при 2713 измерениях).

* 1. **Предлагаемый подход**

Все рассмотренные методы требуют большое количество агрегации данных, как например в методе основанном на гистограммах [8] или в работе Диденко С.М. [3] (более 1000 выявленных движений). А также большого количества времени для построения классификатора, что делаем невозможным работу системы в реальном применении.

Предлагаемых подход основан на проверке пользователя по каждому отдельному движению, что позволяет работать модели в реальном времени.

Для обучения используется 15 характеристик движения и метод одноклассовой классификации Isolation Forest, что позволяет хорошо отчищать выборку от выбросов, повышает точность модели, а также позволяется моментально обучить классификатор.

1. **Написание программного продукта для сбора данных**

**2.1 Написание программы для записи действий мыши**

Основной программный модуль для сбора данных mouse\_logger.py был написан на языке Python 3.7. Его задачей является сбор данных о движении мыши пользователя.

Для реализации программы с заданным функционалом потребовалось обращение к системным функциям OS Windows. Для этого была использована библиотека ctypes. Для получения координат (x, y) мыши использовалась функция windll.user32.GetCursorPos(), для получения состояния кнопок мыши использовалась функция windll.user32.GetKeyState(ID\_BUTTON), а для получения текущего момента времени использовалась стандартная библиотека time. Показания снимались каждые 20 мс. Такая частота дискретизации была подобранна эмпирическим путем, для того, чтобы не было потери информации и в тоже время не происходило быстрого засорения памяти. В начале работы программы формируется .txt файл в который происходит непрерывная запись событий. В результате работы получается файл с полями   
< action, x, y, ti>.

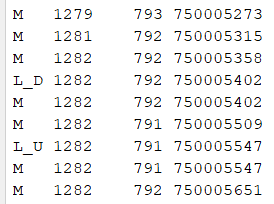


Рисунок 3 – Часть фала сбора данных программы

**2.2 Написание программного модуля для установки программы записи движения мыши**

Программный модуль install.py был написан на языке Python 3.7. Его задачей является установка основной программы на компьютер пользователя так, чтобы в дальнейшем запуск происходил автоматически.

В ходе исследования было выяснено, что реализовать программный модуль как windows сервис не представляется возможным, т.к. по соображениям безопасности обращение из сервиса к winAPI невозможно начиная с Windows Visa. Поэтому программный модуль install.py добавляет программу в автозагрузку. Пример работы модуля install.py представлен на рисунке 4.

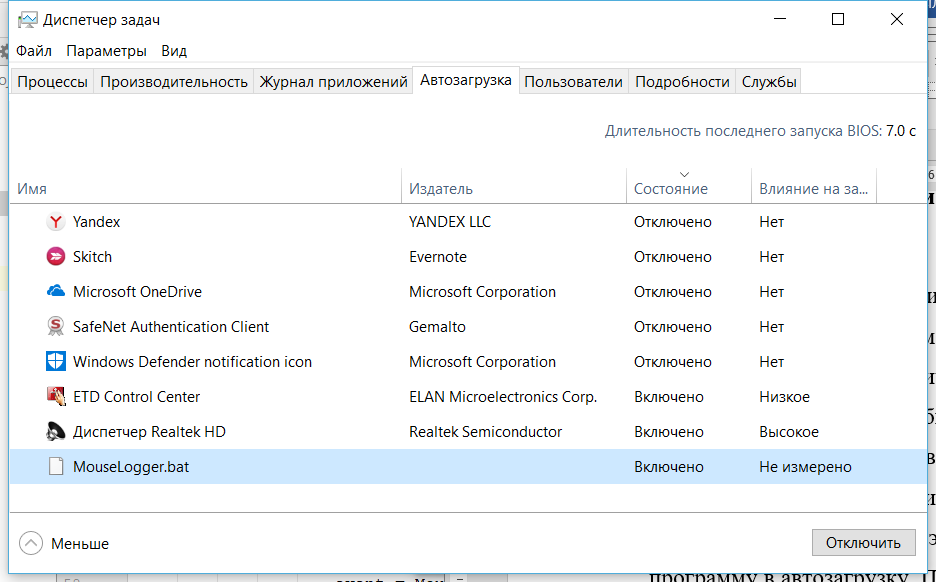


Рисунок 4 – Диспетчер задач, автозагрузка

1. **Написание программного продукта для обработки данных**

**3.1 Обработка данных**

Для анализа данных был взят открытый набор данных на GitHub [10], и ранее собранные данные.

Данные с GitHub представляют собой записанные RDP сессии клиентов в .csv файлы с полями рисунок 5:

record timestamp – прошедшее время (в секундах) с начала сеанса, записанное устройством мониторинга network;

client timestamp – прошедшее время (в секундах) с начала сеанса, записанное клиентом RDP;

button – текущее состояние кнопок мыши;

state – дополнительная информация о текущем состоянии мыши;

x – координата x курсора на экране;

y – координата y курсора на экране.

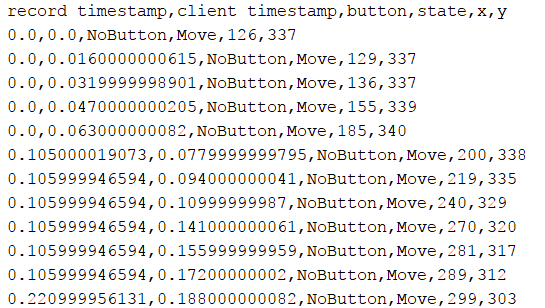


Рисунок 5 – Пример части файла с данными одного из пользователей

Набор данных содержит в себе данные о десяти пользователях, у каждого из которых имеется от пяти до семи сессий рисунок 6.

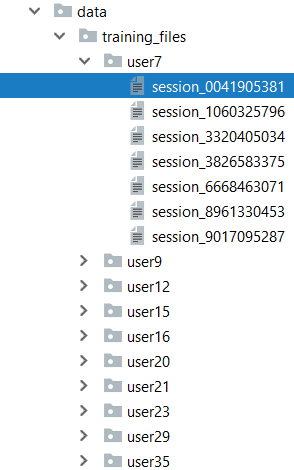


Рисунок 6 – Представление структуры данных

После визуального просмотра данных была произведена их отчистка и подготовка к обработке. Для этого была написана программа на языке Python 3.6 – *data\_preparation.py*.

В ходе обработки данные были удалены одинаковые строчки, лишняя информация, а также все данные были переведены в удобные для обработки единицы исчисления. В итоге был получен файл с данными <mouse state, xi, yi, ti> Рисунок 3, т.е. состояние мыши, xi и yi координаты, и момент времени ti в милисекундах. Данные были приведены к общему виду, к такому же как и при самостоятельном сборе данных.

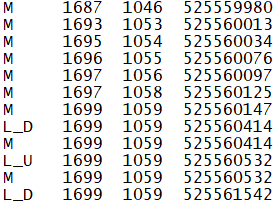


Рисунок 7 – Пример части файла с данными одного из пользователей

после обработки

### **3.2 Визуализация**

Далее данные были визуализированы для возможности упрощения анализа, для этого была написана программа *tsv\_to\_png.py*.

На рисунке 8 представлен пример визуализации сессий, где синими линиями обозначена траектория ведения мыши, а красными – нажатие левой клавиши мыши.

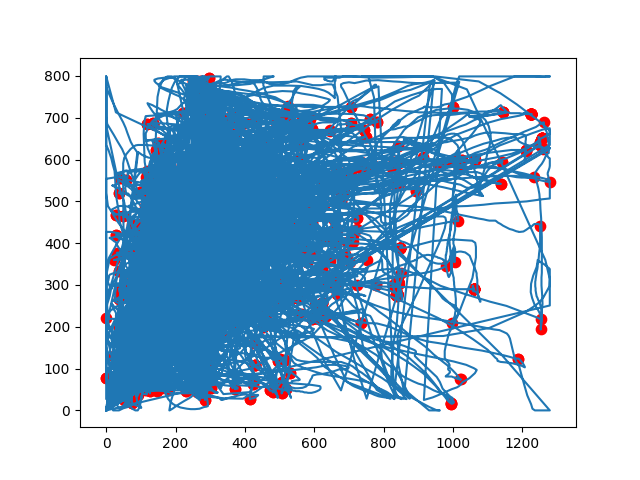


Рисунок 8 – Пример визуализации сессии пользователя user7,   
сессия session\_0041905381\_cl

Визуализация сессии не принесла никаких явных результатов. Из рисунка 8 можно сделать только такой вывод: что пользователь использует ОС Windows, т.к. большинство движений тянется к меню Пуск.

Для более глубокого анализа далее была проведена визаулизация всех движений, всех сессий пользователей рисунок 9.

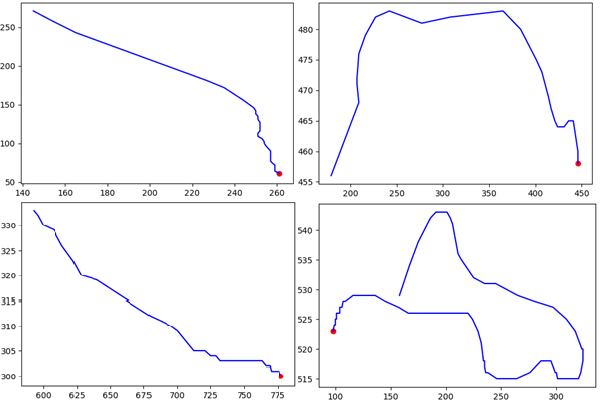


Рисунок 9 – Пример визуализации движений

Из рисунка 9 видно, что движения в основном имеют дугообразный вид, однако присутствуют и нестандартные траектории.

### **3.3 Отчистка данных и выявления особенностей движений**

За основу было взято два типа движения пользователей:

* отдельное движение с кликом на конце – move click (MC): остановка курсора, за тем движение и остановка с кликом в конце траектории;
* отдельное движение – mouse move (MM): остановка курсора, за тем движение и остановка.

Для выявления движений и их особенностей, нужно определиться, что мы считаем за отдельное движение. Ведь, если не ввести рамки, то за одно отдельное движение можно считать всю сессию. Для начала исследования было решено установить следующие параметры для получения широкой выборки движений:

1. перед началом движений была совершена остановка более чем   
   на 0.3 сек.;
2. после движения должна быть остановка больше чем на 0.3 сек. (только для движений типа MM);
3. движение должно содержать минимум 5 точек;
4. движение не может длиться меньше 0.2 сек. и более 5 сек.;
5. длина движения не более 2000 пикселей и не менее 50;
6. удержание левой кнопки мыши при клике не более 2 сек (только для движений типа MC).

Данные параметры были взяты исходя из здравого смысла для удаления из выборки самых больших выбросов и повышения информативности.

Было решено рассчитать следующие информационные параметры для MM и MC типов движений:

1. – время движения,

,

где *n* – количество точек движения;

1. – длина траектории,

L = ,

где разность между двумя строками в таблице данных;

1. – средняя горизонтальная скорость,

;

1. – средняя вертикальная скорость,

;

1. – средняя скорость,

;

1. – максимальная скорость,

;

1. – ускорение,

;

1. – рывок,

;

1. – время остановки перед нажатием на клавишу мыши, только для движения MC;
2. – время удержания левой кнопки мыши, только для   
   движения MC;
3. – угол начала движения, угол между направлением начального движения (до 3–й вершины траектории) и линией, связывающей начальную и конечную точки траектории,

;

1. – среднее квадратичное отклонение траектории,

,

где , - точка пересечения перпендикуляра опущенного из i-oй точки траектории с прямой кратчайшего пути траектории. Кратчайшим путем является отрезок, соединяющий первую и последнюю точки траектории.

1. – средняя угловая скорость,

;

1. – кривизна движения,

;

1. – изменение искривления движения,

.

Каждая сессия пользователя была обработана, и в ней выявлялись движения move\_click и mouse\_move, и для каждого движения производились расчеты всех информационных параметров. Все результаты были записаны в файл *FulloutPutParamClick.tsv* рисунок 10.

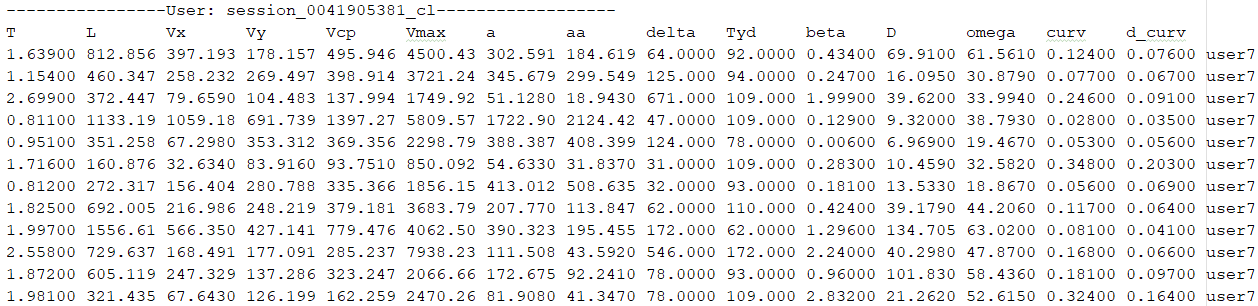


Рисунок 10 – Пример вычислений показателей сессии

### **3.3.1 Отчистка данных для движения типа MC**

Далее была проведена более строгая отчистки данных от выбросов. Для этого в начале была построена гистограмма всех параметров для движения типа MC рисунок 11.

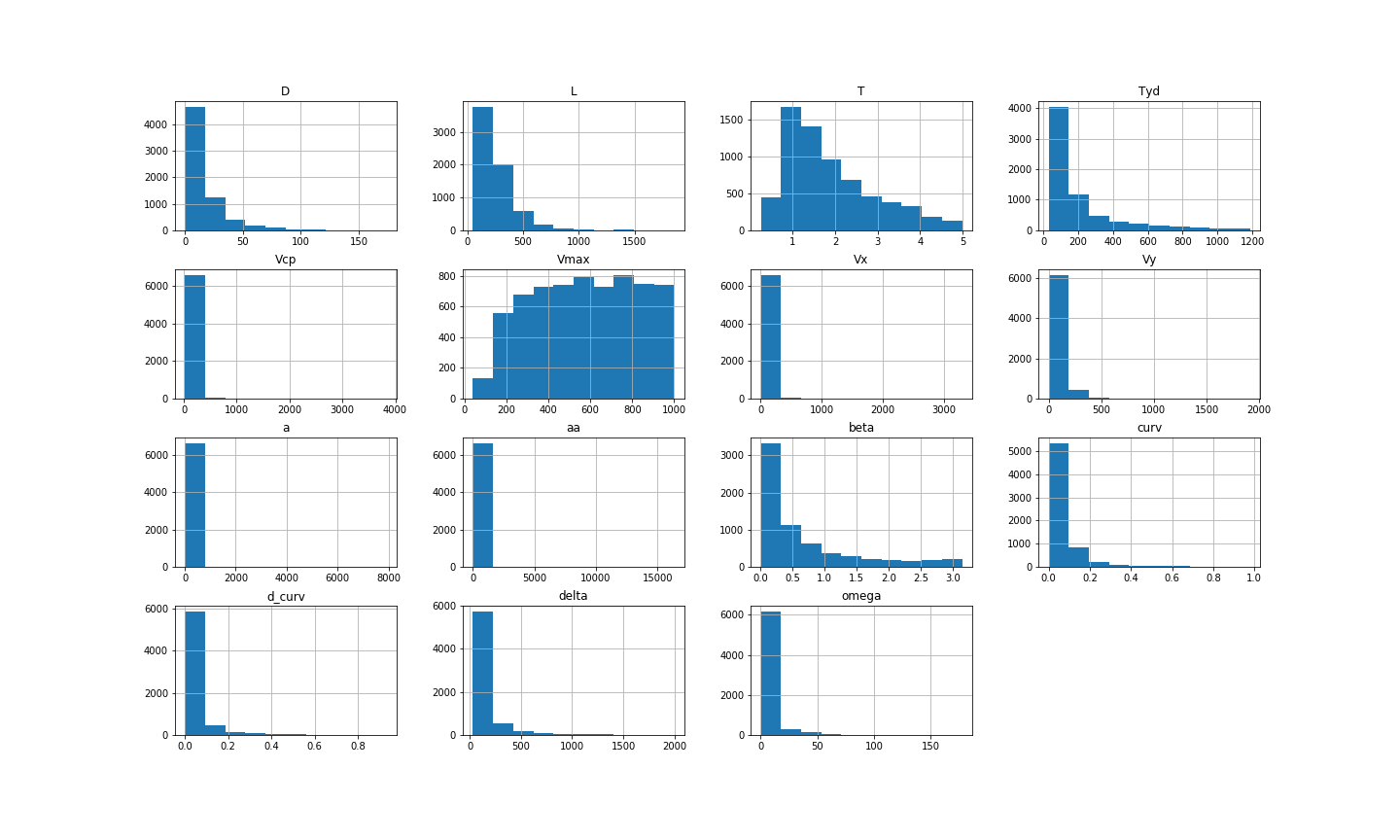


Рисунок 11 – Гистограммы всех параметров движения типа MC

Для наглядности можно рассмотреть более детально один из параметров рисунок 12.

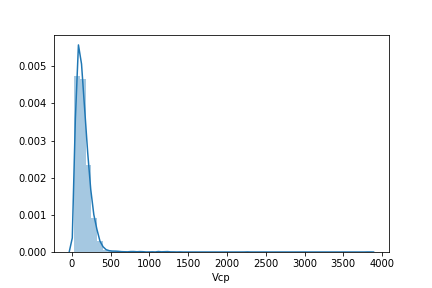


Рисунок 12 – Гистограмма параметра Vcp для движения типа MC

Из гистограмм всех параметров (рисунок 11) видно, что распределение далеко от нормального, а также видно, что существуют сильные выбросы.

Далее экспериментальным методом были подобранны границы значений всех параметров:

1. : 500 < < 3000 мс;
2. : 100 < < 1000 пикселей;
3. : 100 < < 1000 пикселей/мс;
4. : 800 < < 6500 пикселей/мс;
5. : 500 < < 3000 пикселей/мс2;
6. : 1200 < < 20 мс;
7. : 1200 < < 20 мс;
8. : 500 < < 3000 рад;
9. : 500 < < 3000 пикселей;
10. : 500 < < 3000 рад/мс;
11. : 500 < < 3000 рад/пиксель;
12. :

500 < < 3000 рад/(пиксель\*мс).

После введенных ограничений на параметры, были снова построены гистограммы, которые отражены на рисунке 13.

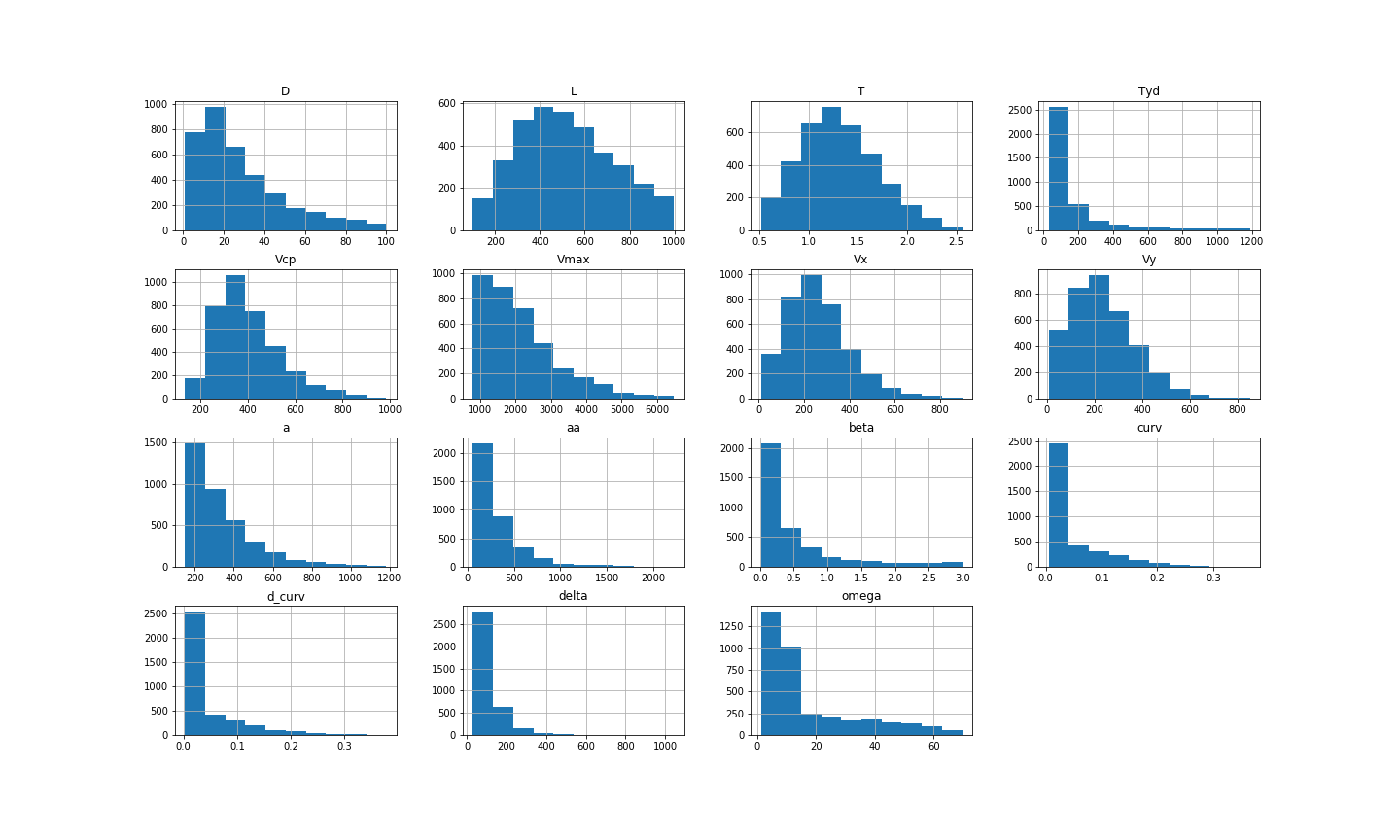


Рисунок 13 – Гистограмма всех параметров движения типа MC   
после отчистки от выбросов

Из гистограмм видно, что распределения параметров стали более нормальными и наглядно видно, что выбросов стало меньше.

### **3.3.2 Отчистка данных для движения типа MM**

Далее была проведена отчистки данных от выбросов для движения типа MM. Для этого в начале была построена гистограмма всех параметров рисунок 14.

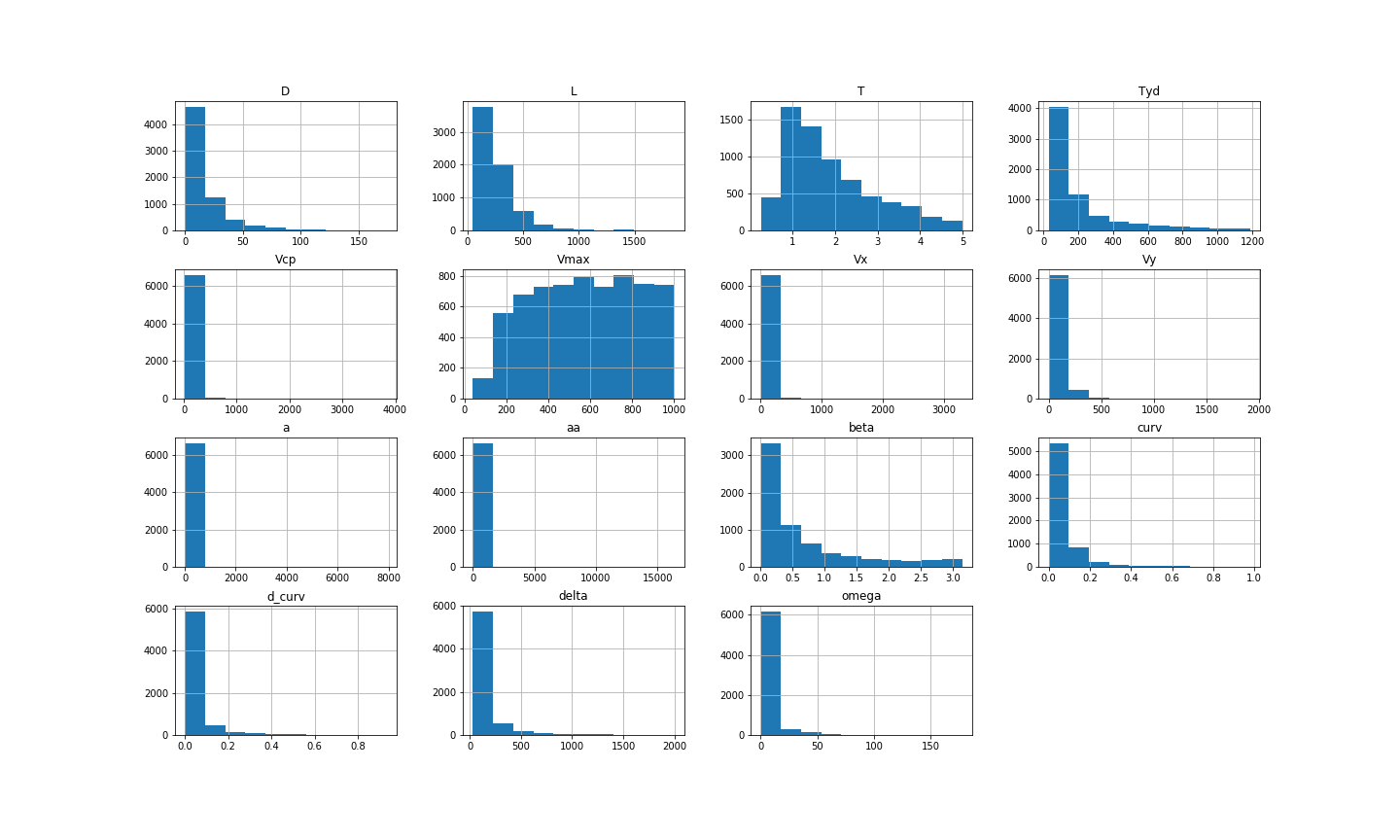


Рисунок 14 – Гистограмма всех параметров движения типа MC

Для наглядности можно рассмотреть более детально один из параметров рисунок 15.

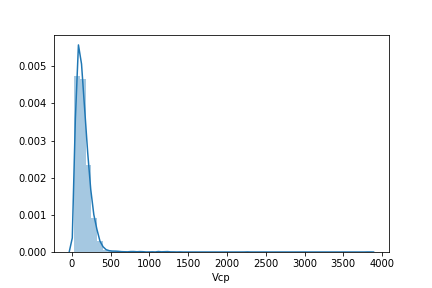


Рисунок 15 – Гистограмма параметра Vcp для движения типа MC

Из гистограмм всех параметров (рисунок 14) видно, что распределение далеко от нормального, а также видно, что существуют сильные выбросы.

Далее экспериментальным методом были подобранны границы значений всех параметров:

1. : 500 < < 3000 мс;
2. : 100 < < 1000 пикселей;
3. : 100 < < 1000 пикселей/мс;
4. : 800 < < 6500 пикселей/мс;
5. : 500 < < 3000 пикселей/мс2;
6. : 1200 < < 20 мс;
7. : 1200 < < 20 мс;
8. : 500 < < 3000 рад;
9. : 500 < < 3000 пикселей;
10. : 500 < < 3000 рад/мс;
11. : 500 < < 3000 рад/пиксель;
12. :

500 < < 3000 рад/(пиксель\*мс).

После введенных ограничений на параметры, были снова построены гистограммы, которые отражены на рисунке 16.

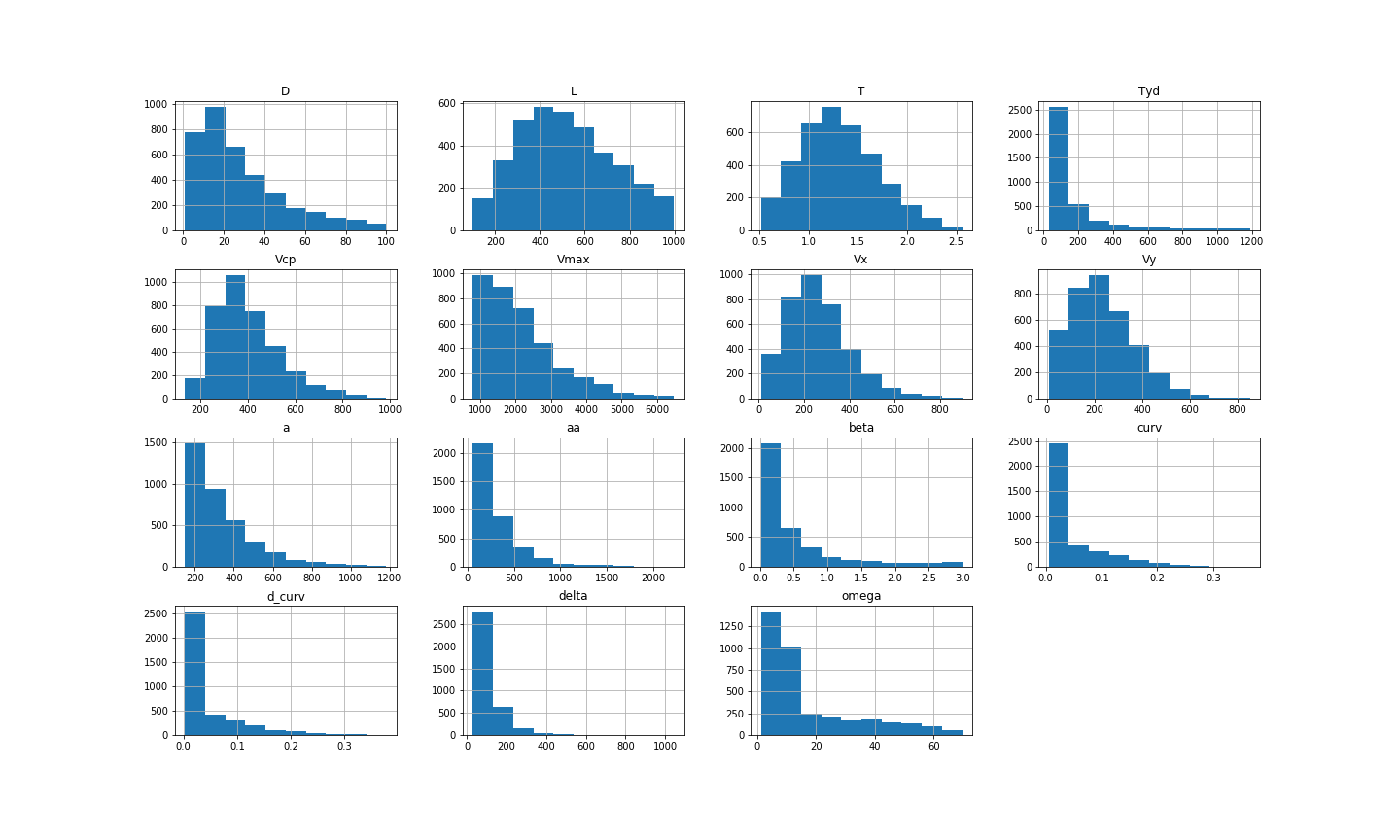


Рисунок 16 – Гистограмма всех параметров движения типа MC   
после отчистки от выбросов

Из гистограмм видно, что распределения параметров стали более нормальными и наглядно видно, что выбросов стало меньше.

Для того чтобы избежать влияние выбросов на конечный результат, каждые 10 движений усреднялись, т.е. брались медианы значений информационных параметров по 10-ти движениям. Таким образом был получен файл, состоящий только из информативных векторов. Все результаты были записаны в файл *ParamClick.tsv* рисунок 17.

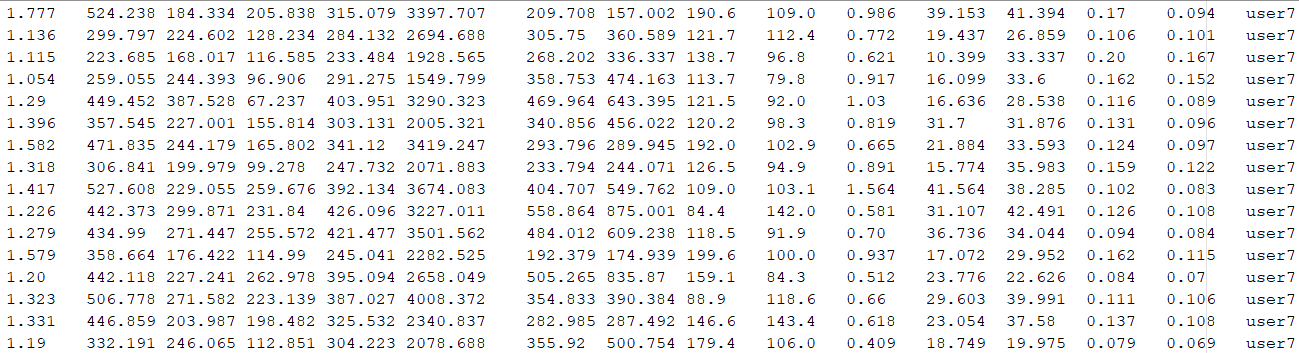


Рисунок 17 – Пример информативных векторов сессии

из файла ParamClick.tsv

Таким образом по результатам этих трех этапов (отчистка, визуализация, обработка данных) можно сделать выводы о том, что необходимо производить отчистку данных для получения более точных результатов; графический анализ данных сессий пользователей не принес никаких результатов в связи с беспорядочностью передвижения и кликов мыши. Однако из визуализации отдельных движений стало понятно, что в основном у всех пользователей движения имеют дугообразный вид. Из этого можно сделать вывод, что у разные пользователи совершают похожие движения, но по-разному, с разными информационными параметрами; расчет средних показателей параметров по 10-ти движениям показал, что у одного и того же пользователя значения примерно одинаковые.

1. **Построение модели многоклассовой классификации пользователей и оценка ее эффективности**

За курсорный почерк пользователя были взяты вектора усреднённых значений параметров по всем сессиям полученные в первой части исследования из файла *ParamClick.tsv*. Для получения более точной оценки модели была использована перекрестная проверка с разделением полученной выборки на обучающую и тестовую в пропорции 85% на 15%. Классификация пользователей сводится к задаче классификации информационных векторов.

Для этой задачи было решено рассмотреть два метода машинного обучения: «k-ближайших соседей» и «Случайный лес». Была написана программа *Classifier.py*.

В начале, с помощью библиотеки *scikit-learn*, был построен классификатор на основе метода «k-ближайших соседей» рисунок 18.

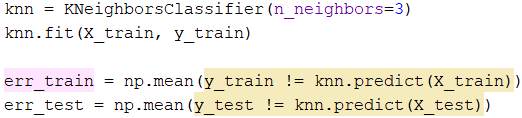


Рисунок 18 – Листинг построения классификатора

«k-ближайших соседей»

Классификатор показал точность 54%, что недопустимо.

Далее был построен классификатор на основе метода «Случайный лес» рисунок 19.

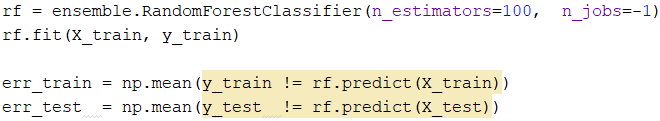


Рисунок 19 – Листинг построения классификатора

«Случайный лес»

Классификатор показал точность 74%, что уже лучше, но недостаточно.

Далее был произведен отбор значащих признаков с помощью метода rf.feature\_importances\_ рисунок 20.

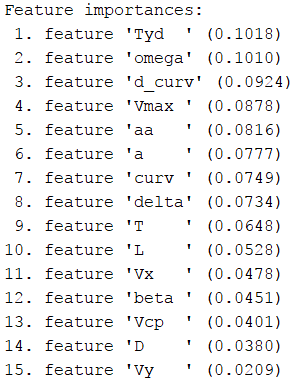


Рисунок 20 – Вывод значений важности признаков

Отбор признаков показал, что [], оказались наименее значащие. Эти признаки было решено убрать, а также ], для того, чтобы убрать прямые зависимости средней длины и времени движения.

Далее были заново построены информационные вектора пользователей без наименее значащих признаков и записаны в файл *ParamClick\_ImpFeat.tsv* рисунок 21.

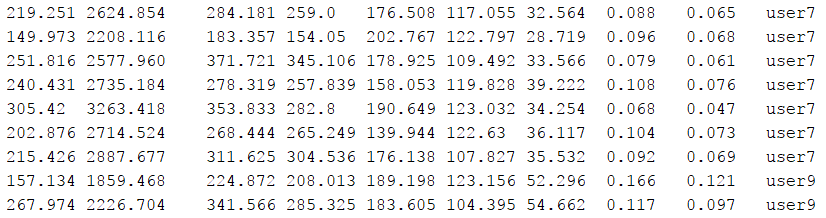


Рисунок 21 – Пример информативных векторов сессии

из файла ParamClick\_ImpFeat.tsv

Классификатор показал точность в 81%, т.е. ошибка определения принадлежности тестового вектора к классу была 19%.

Поскольку первые результаты испытания модели, примененной в данном исследовании, показали точность более 80%, то на основе приведенных данных можно сделать вывод об эффективности применения метода анализа курсорного почерка в целях классификации пользователей. Однако для решения задачи аутентификации пользователя такой метод не подходит, т.к. с увеличением количества пользователей будет падать точность модели. В высоконагруженной системе с миллионами пользователей модель будет малоэффективной.

Эффективность модели можно повысить за счет следующих действий: сбора большего количества данных; повышения качества отчистки данных от выбросов; более глубокого анализа особенностей поведения; использования большего количества типов движения; применения более подходящего классификатора, а также настроек самого классификатора.

1. **Построение модели одноклассовой классификации пользователей и оценка ее эффективности**

Для построения модели аутентификации пользователя по курсорному почерку был использован метод одноклассовой классификации Isolation Forest. Этот классификатор построен на методе Случайный лес, но в данном случай происходит не многоклассовая классификация, а «один против всех». Выбор данного классификатора также связан с тем, что он имеет много тонких настроек и умеет хорошо обрабатывать выбросы, тем самым мы сможешь повысить эффективность нашей модели.

Для каждого пользователя был построен отдельный классификатор. Для оценки модели была использована проверка с разделением полученной выборки на обучающую и тестовую, где обучающей выборкой были все сессия, кроме последней, а тестовой последняя сессия *legitim* и все сессии других пользователей *fraud*.

С помощью библиотеки *scikit-learn*, были построены классификаторы для двух типов движения *move\_click* и *mouse\_move* на основе метода Isolation Forest. Были написаны программы *IF\_move\_click.py* и *IF\_move.py*.

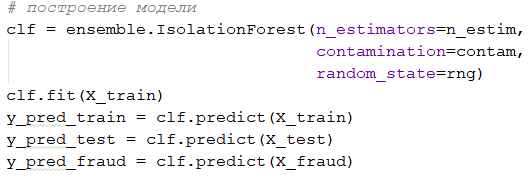


Рисунок 22 – Листинг построения классификатора

«Isolation Forest»

Далее происходила настройка классификатора. Важнейшим параметром является *Contamination*, он отвечает за загрязнённость обучающей выборки. Параметр был подобран экспериментальным методом. Далее для оценки точности была построена ROC кривая и вычислена AUC метрика для классификатора движения *mouse\_click*. Результаты представлены на рисунке 23.

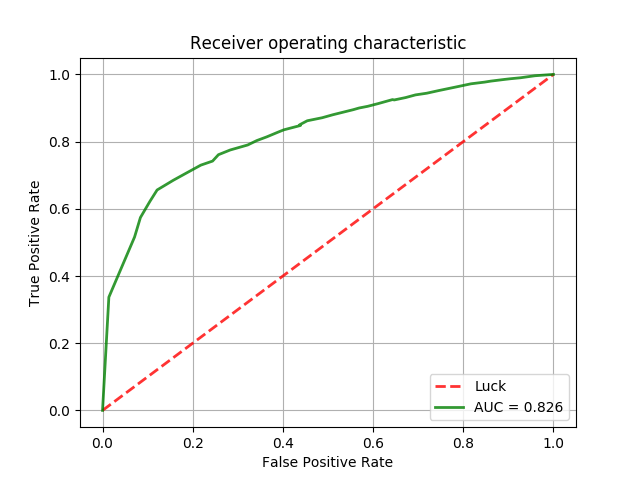


Рисунок 23 – ROC кривая для движения mouse\_click

Классификатор показал точность 98.59% на legitim выборке, где этой выборкой служили информационные вектора, полученные из последней сессии пользователя. И 33.69% на fraud выборке, которая состояла из всех информационных векторов других пользователей.

Далее для оценки точности была построена ROC кривая и вычислена AUC метрика для классификатора движения *mouse\_move*. Результаты представлены на рисунке 24.

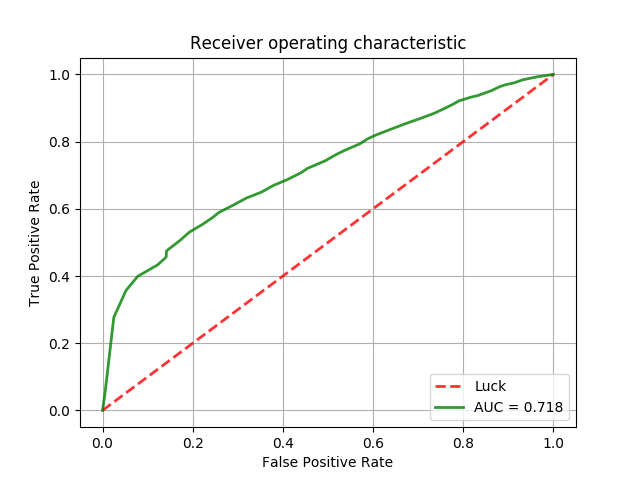


Рисунок 24 – ROC кривая для движения mouse\_move

Классификатор показал точность 97.79% на legitim выборке, где этой выборкой служили информационные вектора, полученные из последней сессии пользователя. И 25.96% на fraud выборке, которая состояла из всех информационных векторов других пользователей.

На основе приведенных данных можно сделать вывод об эффективности применения метода анализа курсорного почерка в целях одноклассовой классификации пользователей. Однако для решения задачи аутентификации пользователя требуется расставить границы чувствительности модели. Выбор данной границы обуславливается тем, что создаваемая система работает, как дополнительная аутентификация, а значит влияние на правомерного пользователя должно быть минимальным.

1. **Построение модели одноклассовой классификации пользователей и оценка ее эффективности**

В предыдущей главе мы получили классификаторы способные выдавать верный вердикт по одному информационному вектору с точностью 98.59% и 97.79% для legitim выборки, 33.69% и 25.96% для fraud выборки для движений *mouse\_click* и *mouse\_move* соответственно. Однако пользователь за сессию делает много движений, что позволяет получить несколько информационных векторов, что повышает точность результата. Расставив границы таким образом, что если более 60% информационных векторов классификатор относит к правомерному пользователю, то пользователь считается аутентифицированным и выдается вердикт *Legitimate user*, если менее 40%, то пользователь не аутентифицируется и выдается вердикт *Fraud*, иначе классификатор не принимает никакого решения и выдается вердикт *Unknown*.

Данные настройки границ модели позволили достичь результатов:

Таблица 2 – Результаты для движения mouse\_click

|  |  |
| --- | --- |
| Для движения *mouse\_click* | |
| Истинное значение | Вердикт системы |
| Legitimate user | Legitimate user: 90% |
| Unknown: 10% |
| Fraud: 0% |
| Fraud | Legitimate user: 10% |
| Unknown: 50% |
| Fraud: 40% |

Таблица 3 – Результаты для движения mouse\_ move

|  |  |
| --- | --- |
| Для движения *mouse\_move* | |
| Истинное значение | Вердикт системы |
| Legitimate user | Legitimate user: 90% |
| Unknown: 10% |
| Fraud: 0% |
| Fraud | Legitimate user: 20% |
| Unknown: 60% |
| Fraud: 20% |

Результаты показали:

1. 0% ошибку I рода и 10% ошибку II рода для движения *mouse\_click***;**
2. 0% ошибку I рода и 20% ошибку II рода для движения *mouse\_move*.

Испытания модели, примененной в данном исследовании, показали высокую точность. На основе приведенных данных можно сделать вывод об эффективности применения метода анализа курсорного почерка в целях дополнительной аутентификации пользователей.

Эффективность модели можно повысить за счет следующих действий: сбора большего количества данных; повышения качества отчистки данных от выбросов; более глубокого анализа особенностей поведения; использования большего количества типов движения; применения более точных настроек классификатора; определить корреляцию между движениями и объединить их в одну единую решающую систему.

# **ЗАКЛЮЧЕНИЕ**

В работе был написан программный продукт для отчистки данных и переформатирование их в удобную для анализа форму. Был проведен анализ особенностей движений пользователей, и выявлено один наиболее явный вид движения: перемещение мыши с кликом в конце траектории и отдельное движение. Движения такого типа были выявлены из сессий пользователя, и для них были посчитаны параметры. Усредненное значение этих параметров по сессии были приняты как портрет пользователя и разделены на обучающую и тестовую выборки. Было произведено обучение классификатора методом «k-ближайших соседей» и «Случайный лес». «Случайный лес» показал более высокую точность в 74%, после чего был произведен отбор значащих признаков, после чего классификатор показал точность правильной идентификации пользователя 81%. Однако для решения задачи аутентификации пользователя такой метод не подходит, т.к. с увеличением количества пользователей будет падать точность модели. В высоконагруженной системе с миллионами пользователей модель будет малоэффективной. Далее был построен одноклассовый классификатор с помощью метода «Isolation Forest», который показал способность выдавать верный вердикт по одному информационному вектору с точностью 98.59% и 97.79% для legitim выборки, 33.69% и 25.96% для fraud выборки для движений *mouse\_click* и *mouse\_move* соответственно. Это дало возможность построить модель аутентификации пользователя по курсорному почерку с 0% ошибку I рода и 10% ошибку II рода для движения *mouse\_click* и 0% ошибку I рода и 20% ошибку II рода для движения *mouse\_move*. Соответственно гипотеза исследования подтвердилась: на основе полученных данных можно сделать вывод об эффективности применения метода анализа курсорного почерка в целях аутентификации пользователя. В целях повышения эффективности модели можно применить следующие методы: сбор большего количества данных; повышение качества отчистки данных от выбросов; использования большего количества типов движения; более глубокий анализ особенностей поведения; применение более точных настроек классификатора; использование ансамбля решений вынесения вердикта для разных типов движений.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

* 1. Глобальное исследование утечек конфиденциальной информации в 2017 году. Аналитический центр InfoWatch. 2017 г. – Режим доступа: https://www.infowatch.ru/analytics/report2017 (дата обращения: 01.08.2018).
  2. Диденко С.М. Исследование динамики работы пользователя с манипулятором мышь/ С.М. Диденко, В.А. Шапцев // Математическое и информационное моделирование. – Тюмень: Изд–во Тюм. ун–та, 2004. – С. 295–304.
  3. Диденко С.М. Разработка и исследование компьютерной модели динамики системы «пользователь – мышь»: автореф. дис. ... канд техн. наук: 05.13.19 / С.М. Диденко. – Тюмень, ТюмГУ, 2007. – 18 с.
  4. Monika Bhatnagar, Raina K Jain and Nilam S Khairnar. Article: A Survey on Behavioral Biometric Techniques: Mouse vs Keyboard Dynamics // IJCA Proceedings on International Conference on Recent Trends in Engineering and Technology 2013 ICRTET(5), May 2016. – P. 22–25.
  5. Jain–Shing Wu, Chih–Ta Lin, Yuh–Jye Lee, Song–Kong Chong. Keystroke and Mouse Movement Profiling for Data Loss Prevention // Journal of Information Science and Engineering, 2015, no 31, P. 23–42.
  6. H. Gamboa, and A. Fred, “Behavioural Biometric System Based on Human Computer Interaction”, Proc. of SPIE, vol. 5404, pp. 381–392, 2004.
  7. R. B. Abernethy, “The New Weibull Handbook”, 2001.
  8. A. A. E. Ahmed, and I. Traore, “A New Biometric Technology Based on Mouse Dynamics”, IEEE Transactions on Dependable and Secure Computing, Vol. 4, No. 3, pp. 165–179, July-September 2017.
  9. M. Pusara, and C. E. Brodley, “User Re-Authentication via Mouse-movements”, Proc. of ACM Workshop Visualization and Data Mining for Computer Security, 2004.
  10. Fülöp, Á., Kovács, L., Kurics, T., Windhager–Pokol, E. (2016). Balabit Mouse Dynamics Challenge data set. Available at: https://github.com/balabit/Mouse–Dynamics–Challenge.
  11. Salil P. Banerjee, Damon L. Woodard. Biometric Authentication and Identification using Keystroke Dynamics: A Survey // Journal of Pattern Recognition Research, 2012. no 7. P. 116-139.
  12. Еременко А.В.. Разграничение доступа к информации на основе скрытого мониторинга действий пользователей в информационных системах: скрытая идентификация / А.В. Еременко, Е.А. Левитская, А.Е. Сулавко, А.Е. Самотуга // Вестник СибАДИ. – 2014. – № 6(40). – С. 92-102.
  13. Сулавко А.Е. Разграничение доступа к информации на основе скрытого мониторинга действий пользователей в информационных системах: портрет нелояльного сотрудника / А.Е. Сулавко, А.В. Еременко, Е.А. Левитская // Известия Трансиба / ОмГУПС. – 2015. – № 1(21). – С. 80-89.
  14. Margit Antal, László Zsolt Szabó, Izabella László. Keystroke Dynamics on Android Platform // 8th International Conference Interdisciplinarity in Engineering, INTER-ENG 2014, 9-10 October 2014, Tirgu Mures, Romania. - P. 1-7.
  15. P. Grother, and E. Tabassi, “Performance of Biometr ic Sample Quality Measures”, National Institute of Standards and Technology, 2016.