

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
имени М.В. ЛОМОНОСОВА  
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

*На правах рукописи*

**Казачук Мария Андреевна**

**ДИНАМИЧЕСКАЯ АУТЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЕЙ  
НА ОСНОВЕ АНАЛИЗА РАБОТЫ С КЛАВИАТУРОЙ  
КОМПЬЮТЕРА**

Специальность 05.13.11 – Математическое и программное обеспечение  
вычислительных машин, комплексов и компьютерных сетей

**ДИССЕРТАЦИЯ**

на соискание ученой степени

кандидата физико-математических наук

Научный руководитель:  
кандидат физико-математических наук,  
доцент Петровский Михаил Игоревич

Москва – 2019

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	5
1 АНАЛИТИЧЕСКИЙ ОБЗОР СОВРЕМЕННЫХ РЕШЕНИЙ И АЛГОРИТМОВ ПО ДИНАМИЧЕСКОЙ АУТЕНТИФИКАЦИИ ПОЛЬЗОВАТЕЛЕЙ НА ОСНОВЕ КЛАВИАТУРНОГО ПОЧЕРКА .....	17
1.1 Обзор современных индустриальных решений по динамической аутентификации пользователей на основе клавиатурного почерка .....	17
1.2 Сбор и обработка данных .....	25
1.2.1 Особенности web-сбора данных .....	25
1.2.2 Особенности локального сбора данных .....	27
1.2.3 Описание собираемых данных .....	28
1.2.4 Фильтрация собранных данных .....	28
1.3 Выделение набора характеристических признаков .....	28
1.3.1 Разбиение потока событий на временные окна .....	29
1.3.2 Расчет характеристических признаков для каждого временного окна .....	30
1.4 Постобработка векторов признаков .....	32
1.5 Сокращение размерности пространства признаков .....	33
1.5.1 Методы, основанные на анализе используемых пользователем клавиш .....	33
1.5.2 Метод главных компонент .....	33
1.5.3 Эвристические методы сокращения размерности признакового пространства .....	35
1.6 Построение модели пользователя .....	36
1.6.1 Одноклассовый метод k-ближайших соседей .....	37
1.6.2 Одноклассовый метод опорных векторов (SVC, Single Class SVM) .....	37
1.6.3 Модель гауссовых смесей .....	39
1.6.4 Нейронные сети .....	39
1.6.5 Сравнение качества работы используемых подходов .....	40

1.7	Выводы.....	45
2	ПРЕДОБРАБОТКА ДАННЫХ, ХАРАКТЕРИЗУЮЩИХ ДИНАМИКУ РАБОТЫ ПОЛЬЗОВАТЕЛЕЙ С КЛАВИАТУРОЙ КОМПЬЮТЕРА.....	47
2.1	Описание используемых для исследования наборов данных.....	48
2.2	Фильтрация собранных данных.....	51
2.3	Разделение потока собираемых данных на временные окна.....	52
2.4	Расчет характеристических признаков для каждого временного окна.....	53
2.5	Выделение стабильных признаков, характеризующих динамику работы пользователя с клавиатурой компьютера.....	58
2.5.1	Формальная постановка задачи расчета уровня стабильности произвольного составного характеристического признака.....	60
2.5.2	Вычисление уровня стабильности составного характеристического признака с использованием критерия Колмогорова-Смирнова.....	61
2.6	Дискретизация признаков по квантилям .....	63
2.7	Экспериментальное исследование .....	64
2.7.1	Исследование параметров построения временных окон .....	65
2.7.2	Исследование признакового пространства модели представления и подбор соответствующих параметров .....	68
2.7.3	Исследование методов обработки признаков и подбор соответствующих параметров.....	70
2.7.4	Исследование устойчивости работы предложенных алгоритмов к смене используемого оборудования .....	75
2.8	Выводы.....	76
3	ПОСТРОЕНИЕ МОДЕЛИ КЛАВИАТУРНОГО ПОЧЕРКА ПОЛЬЗОВАТЕЛЯ.....	79
3.1	Задача поиска аномалий в данных. Определение исключения .....	81
3.2	Классические kernel-методы поиска исключений .....	83
3.3	Нечеткий метод выявления аномалий в данных на основе эллиптической кластеризации в RKHS.....	88
3.3.1	Формулировка и обоснование метода .....	89
3.3.2	Алгоритм поиска исключений с использованием процедуры блочного покоординатного спуска .....	92
3.3.3	Исследование сходимости метода .....	94
3.3.4	Исследование работы метода на простейших демонстрационных примерах .....	95

3.3.5	Оценка сложности работы алгоритма .....	96
3.4	Метод подбора оптимальных значений метапараметров алгоритмов одноклассовой классификации .....	98
3.5	Использование t-статистики Уэлша для оценки аномальности поведения пользователя на основе анализа целых сессий работы за компьютером.....	100
3.6	Экспериментальное исследование .....	103
3.7	Выводы.....	106
4	ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ЭКСПЕРИМЕНТАЛЬНОГО ОБРАЗЦА ПРОГРАММНОГО КОМПЛЕКСА .....	108
4.1	Сценарии функционирования.....	109
4.1.1	Сбор поведенческой информации о взаимодействии пользователей с клавиатурой компьютера .....	109
4.1.2	Построение индивидуальных моделей поведения пользователей .....	110
4.1.3	Применение индивидуальных моделей поведения пользователей .....	111
4.2	Программная реализация .....	113
4.2.1	Архитектура системы.....	113
4.2.2	Описание программных компонентов.....	117
4.2.3	Пример использования.....	126
4.3	Экспериментальное исследование .....	137
4.4	Оценка производительности .....	140
4.5	Выводы.....	143
5	ЗАКЛЮЧЕНИЕ .....	145
6	СПИСОК ЛИТЕРАТУРЫ .....	147

# ВВЕДЕНИЕ

## **Актуальность темы исследования**

Последние несколько десятилетий ознаменовали период стремительного развития информационных технологий и их внедрения в жизни людей. Вся важная информация теперь хранится в компьютерах пользователей, и вопросы, связанные с обеспечением безопасности данной информации, становятся наиболее критическими. Под информационной безопасностью понимается комплекс мер, направленный на обеспечение конфиденциальности, целостности и доступности информации. При этом, наиболее важным аспектом является обеспечение конфиденциальности – предотвращение утечки информации, поскольку в случае утери, риску подвергаются и другие факторы. Сценарии внутренних и внешних вторжений на компьютеры пользователей развиваются с каждым днем и необходимо оперативно уметь с ними бороться – разрабатывать системы, способные вовремя предотвращать попытки нарушения конфиденциальности информации.

Одной из ключевых задач при обеспечении конфиденциальности информации является задача аутентификации, или проверки принадлежности субъекту доступа предъявленного им идентификатора [1]. В качестве субъекта доступа рассматривается лицо или единица ресурса автоматизированной информационной системы, действия которой по доступу к ресурсам системы регламентируются правилами разграничения доступа, а его идентификатором называется уникальный признак объекта, позволяющий отличать его от других объектов. В зависимости от типа используемого идентификатора, существующие системы аутентификации можно подразделить на три категории:

- Системы аутентификации, использующие секретное знание (пароль, номер, ключ и т.д.).

Достоинством данных систем является легкость в использовании. Однако, данные системы ненадежны, поскольку используемая секретная информация может быть с легкостью украдена или передана. Также не стоит забывать, что пользователи часто используют короткие пароли, взломать которые с использованием простейших переборных алгоритмов не составляет трудности.

- Системы аутентификации, использующие в качестве идентификаторов физические объекты, принадлежащие пользователям (флэш-накопитель, магнитная карта и т.д.).

Заполучить злоумышленнику физический объект сложнее, чем пароль. Однако, в данных системах продолжает сохраняться возможность утери или компрометации. Также, для применения физических идентификаторов зачастую требуется дополнительное оборудование.

- Системы аутентификации, использующие в качестве идентификаторов биометрические данные пользователей.

Под биометрическими данными понимаются уникальные биологические и физиологические характеристики, которые позволяют установить личность человека. Примерами таких данных являются отпечатки пальцев, скан радужной оболочки глаза, геометрия лица, рукописный или клавиатурный почерк (набор динамических параметров работы пользователя с клавиатурой), голос. Отметим, что биометрические образцы, в отличие от рассмотренных ранее идентификаторов, невозможно забыть или потерять, а также их намного тяжелее скомпрометировать. Однако, биометрические системы аутентификации должны обладать большой чувствительностью, чтобы с высокой точностью аутентифицировать пользователей. Для того, чтобы быть использованными в системах аутентификации, биометрические образцы должны обладать свойствами уникальности и постоянства.

Существующие биометрические образцы подразделяются на два вида:

- Физиологические биометрические образцы.

К физиологическим биометрическим образцам относятся статические физиологические характеристики человека, находящиеся при нем в течение всей жизни (например, отпечатки пальцев, скан радужной оболочки глаза). Системы, основанные на данной биометрической информации, обладают высокой надежностью. Однако, данную информацию все-таки можно подделать [2]. Также, зачастую для использования физиологических биометрических образцов требуется дополнительное оборудование.

- Поведенческие биометрические образцы.

К поведенческим биометрическим образцам относятся поведенческие характеристики человека (например, голос, рукописный и клавиатурный почерк, походка). Поведенческие образцы подделать намного труднее, чем физиологические, поскольку для них не существует определенных шаблонов. Также, для их снятия редко требуется специальное оборудование. Поведенческую

биометрическую информацию можно изменить по воле человека. Поведенческие образцы обладают меньшей надежностью и стабильностью по сравнению с физиологическими биометрическими образцами, но являются высоко информативными и адаптируемыми, поэтому сегодня ведутся активные исследования в области их применения.

В зависимости от принципа осуществления процедуры проверки идентификатора, разделяют статическую и динамическую аутентификацию. Статическая аутентификация заключается в эпизодической проверке идентификатора пользователя. Системы статической аутентификации обладают достаточно простой реализацией. Однако, зачастую личность пользователя проверяется только при входе в систему и дальнейшую смену пользователя отследить невозможно. Динамическая аутентификация решает данную проблему: здесь личность пользователя проверяется постоянно на протяжении всей сессии работы пользователя за компьютером. Тем самым, злоумышленник не сможет завладеть компьютером после того, как легитимный пользователь (пользователь, для которого была построена модель) отойдет от своего рабочего места, забыв выйти из системы. Однако, системы динамической аутентификации потребляют большее количество программно-аппаратных ресурсов компьютера.

Таким образом, существующие системы аутентификации пользователей достаточно уязвимы и подвержены высокому риску вторжений. Их основными недостатками являются невозможность проконтролировать факт смены вошедшего в систему пользователя, а также ненадежность использования паролей и магнитных карт для входа в систему. Поэтому, необходимо разработать систему, позволяющую непрерывно проверять личность работающего за компьютером пользователя (тем самым способную своевременно обнаруживать различные попытки вторжений) и использующую при этом надежный идентификатор. Возможным путем решения данной проблемы является введение динамической аутентификации пользователя во время его взаимодействия со стандартными устройствами ввода, в частности на основе динамики его работы с компьютерной клавиатурой. К достоинствам данного подхода относятся:

- Непрерывная аутентификация и своевременное обнаружение вторжений;
- Высокая надежность за счет использования биометрических показателей;
- Отсутствие требований к наличию специализированного оборудования.

Современные стандарты по информационной безопасности регламентируют (декларируют) уникальность клавиатурного почерка и возможность создания

динамических систем информационной безопасности на основе анализа динамики работы пользователей с клавиатурой компьютера [3 – 5].

Дополнительно, динамическая аутентификация пользователей по клавиатурному почерку может широко применяться для обеспечения безопасности пользователей в Интернете (например, при осуществлении web-платежей). В данном случае, характеристики ввода пользователя, собираемые при помощи web-браузера и отправляемые на сервер, будут служить дополнительным или основным инструментом аутентификации и позволят защитить конфиденциальность данных пользователей в случае похищения их пароля от web-сайта злоумышленниками.

### **Степень разработанности темы**

Динамическая аутентификация пользователей на основе анализа динамики их работы с клавиатурой компьютера является достаточно перспективным направлением исследований и широко применяется для обеспечения безопасности как домашних компьютеров пользователей, так и компьютеров крупнейших корпораций, а также для предотвращения несанкционированного доступа злоумышленников к web-сайтам. Основу исследованию клавиатурного почерка положило исследование работы операторов телеграфа (середина XIX века). Однако, большинство исследований было направлено на изучение клавиатурного почерка пользователей в процессе ввода ими парольной фразы ограниченной длины – сегодня данные системы полноценно изучены и достигают высокого качества распознавания [6–13].

Исследования в области динамической аутентификации пользователей на основе непрерывной работы с клавиатурой компьютера проводятся сравнительно недавно. Существующие решения основаны на следующих концепциях:

- Для осуществления анализа клавиатурного почерка пользователя, необходимо фиксировать следующие характеристики его ввода: код используемой клавиши, тип события (нажатие / отжатие), а также временную метку, соответствующую произошедшему событию. Помимо этого, дополнительно можно фиксировать имя процесса, в рамках которого осуществляется взаимодействие пользователя с клавиатурой, а также адрес web-страницы, с которой работает пользователь.
- После этого, необходимо разбивать поступающий от клавиатуры поток событий на временные окна и вычислять характеристические признаки отдельно для каждого временного окна. Перспективным является использование комбинированного набора характеристических признаков,



включающего в себя как характеристики работы пользователя с одиночными клавишами, так и с комбинациями нескольких клавиш – N-граммами. Эффективно использовать N-граммы при  $N=2$  (диграфы). Также для повышения точности модели перспективно рассматривать скорость набора текста пользователем, а также разбивать все клавиши клавиатуры на группы и рассчитывать процент использования клавиш из каждой группы.

- Современные клавиатуры в среднем содержат порядка 100 клавиш, и при  $N=2$  мы получаем 10000 различных попарных комбинаций клавиш, для каждой из которых рассчитываются различные статистики – размерность полученного признакового пространства получается достаточно большой, при этом само пространство признаков (часть из которых являются шумовыми) получается разреженным, что свидетельствует об актуальности проблемы предобработки данных в данной задаче. Наиболее перспективным и показывающим высокое качество работы методом сокращения размерности признакового пространства является отбор признаков на основе анализа используемых пользователем клавиш (отбор наиболее часто используемых одиночных клавиш и диграфов). Для постобработки признаков в существующих работах предлагается использовать стандартизацию признаков, способствующую сведению к минимуму доминирования каких-либо признаков в строящейся модели.
- Основной сложностью задачи аутентификации является тот факт, что нам доступны данные только одного, легитимного, класса. А примеров нелегитимного целевого класса мало и зачастую их тяжело выделить путем «ручной» разметки. Данные задачи называются задачами одноклассовой классификации. В них легитимная модель строится без использования образцов других классов. При работе с пространством признаков большой размерности, где многие признаки оказываются нерелевантными с точки зрения выделения целевого класса, а также многие признаки являются взаимозависимыми, наиболее перспективными методами построения модели пользователя являются kernel-методы, основанные на переходе из исходного пространства признаков в пространство характеристик высокой размерности (Reproducing Kernel Hilbert Space – RKHS) с использованием потенциальной (kernel) функции и поиске зависимостей в новом результирующем пространстве. Но качество их работы сильно зависит от выбора значений метапараметров, которые тяжело подобрать в силу

отсутствия примеров нелегитимного класса. Наиболее объективной метрикой для оценки качества аутентификации является значение площади под ROC-кривой (ROC AUC), являющееся агрегированной характеристикой качества классификации, не зависящей от соотношения цен ошибок.

Существующие решения обладают рядом серьезных недостатков [6–16]. В частности, точность аутентификации в них достигает порядка 85–90%, ROC AUC ниже 0.90 и качество работы классификатора сильно зависит от выбора значений метопараметров, которые тяжело подобрать в силу отсутствия примеров данных злоумышленника. Также, характер клавиатурного ввода пользователя может меняться во времени, что ведет к снижению качества распознавания (в том числе и при смене используемого оборудования) – современные подходы не способны выделять наиболее стабильные по времени признаковые характеристики. Дополнительно заметим, что зачастую необходимо решать задачу оценки аномальности поведения пользователя за длительный период (например, целую сессию) его работы за компьютером. С помощью классификатора мы сможем получить набор откликов для всех временных окон в рамках рассматриваемого временного интервала. Необходимо на основе полученной последовательности откликов уметь получать единое число – степень аномальности поведения пользователя за продолжительный промежуток времени (являясь агрегированной характеристикой, данная величина позволит более точно оценить аномальность действий пользователя). В существующих работах решение данной проблемы не предлагается. Поэтому необходимо разработать алгоритмы динамической аутентификации пользователей по клавиатурному почерку, обладающие высоким качеством работы, способные выделять наиболее стабильные по времени признаки и определять степень аномальности поведения пользователя как за короткий, так и за длинный промежуток времени. Для проверки предложенных алгоритмов необходимо разработать экспериментальный образец программного комплекса обнаружения аномалий в поведении пользователей.

### **Цели и задачи**

Целью диссертационной работы является исследование и разработка математического и программного обеспечения динамической аутентификации пользователей компьютеров на основе анализа их клавиатурного почерка.

Объектом исследования диссертационной работы является поведенческая информация пользователей при работе с персональным компьютером / ноутбуком (при взаимодействии с клавиатурой компьютера). Под поведенческой информацией

пользователя будем понимать данные о специфике нажатий и отжатий пользователем клавиш клавиатуры.

Предметом исследования диссертационной работы являются численные характеристики поведенческой информации пользователей, использование которых позволит с высокой точностью определять легитимность рассматриваемого пользователя по динамике его клавиатурного ввода.

Для достижения поставленной цели необходимо решение следующих задач:

- Разработка алгоритмов предобработки данных, характеризующих динамику работы пользователей с клавиатурой компьютера, включающих в себя выбор используемого признакового пространства, а также исследование и разработку методов постобработки признаков и сокращения размерности признакового пространства, совместное использование которых позволит рассчитать векторы информативных и стабильных по времени характеристических признаков рассматриваемого пользователя;
- Разработка методов построения модели пользователя, позволяющих достичь высокого качества распознавания (более 0.90 ROC AUC) в данной задаче, разработка метода подбора оптимальных значений метапараметров алгоритмов одноклассовой классификации, а также разработка методов оценки аномальности поведения пользователя на основе анализа целых сессий работы за компьютером;
- Разработка архитектуры и реализация экспериментального образца программного комплекса, выполняющего сбор поведенческой информации, построение и применение индивидуальных моделей поведения пользователей на основе разработанного комплекса алгоритмов для обнаружения аномалий в поведении пользователей.

### **Научная новизна**

Научная новизна диссертационной работы заключается в:

- Предложенном новом подходе к сокращению размерности пространства признаков путем анализа характера изменения распределения признаков с течением времени на основе расчета статистики Колмогорова-Смирнова и выделения наиболее стабильных из них;
- Предложенном новом нечетком методе выявления аномалий в данных на основе эллиптической кластеризации в RKHS, строящем эллиптические контуры с оптимальным центром для выделения аномалий в пространстве

признаков высокой размерности (RKHS). Поиск оптимальных значений метапараметров данного алгоритма осуществляется собственно разработанным методом без использования информации о данных нелегитимного класса;

- Предложенном новом методе оценки аномальности поведения пользователя на основе анализа целых сессий работы за компьютером, использующем t-статистику Уэлша для сравнения работы классификатора на валидационном наборе данных легитимного пользователя и рассматриваемых данных тестового пользователя.

### **Теоретическая и практическая значимость работы**

Теоретическая значимость исследования заключается в том, что разработанные автором методы являются достаточно перспективными и могут использоваться для решения различных прикладных задач интеллектуального анализа данных на этапах уменьшения размерности признакового пространства, а также построения и применения одноклассовых моделей.

Практическая значимость диссертационной работы состоит в разработке и реализации экспериментального образца программного комплекса обнаружения аномального поведения пользователей на основе анализа их клавиатурного почерка. Полученные результаты диссертационной работы могут послужить основой для построения перспективных современных систем информационной безопасности, которые будут включать в себя средства анализа динамики работы пользователей с клавиатурой компьютера. При этом, могут использоваться как все разработанные модули, так и отдельные из них (например, модули сбора данных о динамике работы пользователей с клавиатурой персонального компьютера / ноутбука).

### **Методология и методы исследования**

При получении основных результатов диссертационной работы использовались методы теории вероятностей, математической статистики и теории машинного обучения. При разработке модулей экспериментального образца программного комплекса обнаружения аномального поведения пользователей на основе анализа их клавиатурного почерка использовались методы объектно-ориентированного анализа и проектирования, а также методы низкоуровневого API-перехвата целевых событий от клавиатуры, поступающих в операционную систему.

### **Положения, выносимые на защиту**

1. Предложенный подход к подготовке данных, описывающих клавиатурный почерк пользователя, включающий в себя способ построения признакового пространства и подход к дальнейшей обработке признаков на основе дискретизации их по квантилям совместно с сокращением размерности признакового пространства путем отбора наиболее значимых признаков по уровню их стабильности с использованием критерия Колмогорова-Смирнова, позволяют рассчитать векторы информативных и стабильных по времени характеристических признаков рассматриваемого пользователя;
2. Нечеткий метод выявления аномалий в данных на основе эллиптической кластеризации (ESFC) в RKHS позволяет стоять в пространстве высокой размерности эллиптические области с оптимальным центром для выявления аномалий в данных и тем самым позволяет достичь высокого качества распознавания (более 0.90 ROC AUC) в данной задаче. Подбор оптимальных значений метапараметров данного алгоритма на основе валидационного набора данных позволяет строить стабильные к смене тестового набора данных одноклассовые модели без использования информации о данных нелегитимного класса. Использование t-статистики Уэлша позволяет оценить аномальность поведения пользователя как за короткий, так и за продолжительный период его работы за компьютером;
3. Предложенный комплекс алгоритмов может быть использован в экспериментальном образце мультиагентного программного комплекса для обнаружения аномального поведения пользователей по особенностям работы с клавиатурой компьютера.

### **Личный вклад**

Личный вклад автора заключается в выполнении основного объема теоретических и экспериментальных исследований, а также в разработке архитектуры и реализации экспериментального образца программного комплекса динамической аутентификации пользователей по динамике их работы с клавиатурой компьютера. Полученные результаты диссертационной работы были оформлены автором в виде научных публикаций, а также представлены на научных конференциях. Подготовка части материалов к публикации проводилась совместно с соавторами, причем вклад диссертанта был определяющим.

### **Степень достоверности и апробация результатов**

Результаты, представленные в работе, докладывались:

- на конференции «Intelligent Data Engineering and Automated Learning-IDEAL 2016», Янчжоу, Китай, 14 октября 2016;
- на конференции «Intelligent Data Engineering and Automated Learning-IDEAL 2018», Мадрид, Испания, 22 ноября 2018;
- на конференции «Ломоносовские чтения – 2017», МГУ имени М.В. Ломоносова, Москва, Россия, 26 апреля 2017;
- на конференции «Тихоновские чтения – 2017», МГУ имени М.В. Ломоносова, Москва, Россия, 25 октября 2017;
- на конференции «Тихоновские чтения – 2018», МГУ имени М.В. Ломоносова, Москва, Россия, 1 ноября 2018;
- на конференции «Ломоносовские чтения – 2019», МГУ имени М.В. Ломоносова, Москва, Россия, 24 апреля 2019;
- а также на научном семинаре кафедры автоматизации систем вычислительных комплексов имени члена-корреспондента РАН, профессора Льва Николаевича Королёва весной 2016, весной 2017 и осенью 2017 г.

Основные результаты по теме диссертации изложены в 7 публикациях, 2 из которых опубликованы в изданиях, входящих в систему цитирования Scopus – [14, 15] (из них [14] также входит в систему цитирования Web of Science), [16] опубликована в издании, входящем в систему цитирования RSCI, 4 публикации – [17 – 20] являются тезисами докладов.

Результаты диссертационной работы использовались в НИР «Разработка технологий биометрической идентификации пользователя по признакам, проявляющимся при использовании устройств ввода данных персональных ЭВМ» (Номер договора №01-04/15 от 08 апреля 2015 г), 2015–2017 гг.

### **Объем и структура работы**

Диссертационная работа состоит из введения, четырех глав, заключения и списка литературы. Далее излагается краткое содержание работы.

Первая глава посвящена исследованию существующих подходов к решению задачи динамической аутентификации пользователей на основе анализа их клавиатурного почерка. Рассматриваются как современные промышленные решения, так и существующие научно-технические публикации. Исследуются как системы, основанные только на анализе динамики работы пользователей с клавиатурой компьютера, так и многофакторные решения. Рассматриваются достоинства и недостатки существующих

систем. На основе проведенного аналитического обзора формулируются направления дальнейших исследований в части выбора модели представления поведенческой информации, характеризующей клавиатурный почерк пользователей, и методов обнаружения аномалий.

Вторая глава посвящена исследованию и разработке алгоритмов предобработки данных, характеризующих динамику работы пользователей с клавиатурой компьютера, включающих в себя выбор используемого признакового пространства, а также исследованию и разработке методов постобработки признаков и сокращения размерности признакового пространства. Предложен новый подход к подготовке данных, описывающих клавиатурный почерк пользователя, включающий в себя способ построения признакового пространства и подход к дальнейшей обработке признаков на основе дискретизации их по квантилям, позволивший решить проблему мультимодального распределения характеристических признаков. Разработан подход к сокращению размерности признакового пространства путем отбора наиболее значимых признаков по уровню их стабильности с использованием критерия Колмогорова-Смирнова, анализирующий постоянство распределения характеристических признаков с течением времени. Данный подход позволил построить пространство стабильных по времени признаковых характеристик и тем самым решить проблему падения качества распознавания пользователей с течением времени (в том числе и при смене используемого оборудования).

В третьей главе проводится исследование и разработка методов построения модели пользователя, позволяющих достичь высокого качества распознавания (более 0.90 ROC AUC) в данной задаче, а также разработка методов подбора оптимальных значений метапараметров алгоритмов одноклассовой классификации и оценки аномальности целых сессий работы пользователей за компьютером. Разработан новый эффективный нечеткий метод поиска исключений в данных, строящий в пространстве высокой размерности эллиптические области с оптимальным центром для выявления аномалий. Доказана сходимость данного метода. Разработан метод подбора оптимальных значений метапараметров алгоритмов одноклассовой классификации, позволяющий строить стабильные к смене тестового набора данных модели без использования информации о данных нелегитимного класса. Оценка аномальности поведения пользователя производится как за короткий, так и за продолжительный период работы – с использованием разработанного метода на основе анализа целых сессий работы за компьютером с использованием t-статистики Уэлша. Проводится экспериментальное исследование разработанных алгоритмов.

В четвертой главе проводятся разработка и реализация экспериментального образца мультиагентного программного комплекса (ЭО ПК), использующего предложенный комплекс алгоритмов для обнаружения аномального поведения пользователей по особенностям работы с клавиатурой компьютера. Приводится детальное описание архитектуры и программной реализации разработанного ЭО ПК. Проводится экспериментальное исследование данного ЭО ПК, а также исследуется производительность его основных программных модулей.



# 1 АНАЛИТИЧЕСКИЙ ОБЗОР СОВРЕМЕННЫХ РЕШЕНИЙ И АЛГОРИТМОВ ПО ДИНАМИЧЕСКОЙ АУТЕНТИФИКАЦИИ ПОЛЬЗОВАТЕЛЕЙ НА ОСНОВЕ КЛАВИАТУРНОГО ПОЧЕРКА

На сегодняшний день проблема динамической аутентификации пользователей по динамике их работы с клавиатурой персонального компьютера (ноутбука) является актуальной и широко разрабатывается.

Как и для любой задачи одноклассовой классификации, решение данной задачи состоит из следующих этапов:

- 1) подготовка данных;
- 2) определение и расчет набора признаков, которые будут использоваться при аутентификации пользователя;
- 3) применение методов постобработки признаков;
- 4) применение методов уменьшения размерности данных;
- 5) применение методов построения модели пользователя.

Рассмотрим современные промышленные решения, существующие научно-технические публикации, достигнутые результаты по динамической аутентификации пользователей на основе динамики их работы с клавиатурой персонального компьютера (ноутбука) более подробно.

## **1.1 Обзор современных промышленных решений по динамической аутентификации пользователей на основе клавиатурного почерка**

На сегодняшний день количество существующих на рынке промышленных решений по аутентификации пользователей на основе динамики их работы с клавиатурой персонального компьютера постоянно увеличивается. И если раньше данные системы ограничивались только анализом ввода пары логин / пароль (рассматривалась исключительно статическая аутентификация), то сейчас активно развиваются системы, способные анализировать поведение пользователя за компьютером непрерывно.

Одним из наиболее известных коммерческих решений в области непрерывной фоновой аутентификации пользователей по клавиатурному почерку является продукт BehavioWeb компании BehavioSec [8, 11, 21]. Для анализа поведения пользователя в нем используются ритм и скорость набора текста, а также сила нажатия на клавиши. Программное обеспечение встраивается в web-сайт или web-приложение. Для этого используются поставляемая JavaScript-библиотека, а также J2EE-модуль, встраиваемый в web-сервер для осуществления процедуры аутентификации. Разработчики обращают внимание на то, что их решение анализирует изменение характеристик ввода пользователя с течением времени и периодически обновляет модель пользователя. Однако, используемые для этого алгоритмы не называются.

### **KeyTrac**

Не менее популярным решением является продукт KeyTrac [8, 11, 22], позволяющий осуществлять в фоновом режиме как аутентификацию, так и идентификацию пользователей компьютера, основываясь на динамике их клавиатурного ввода. Данные пользователей (продолжительности нажатия на клавиши клавиатуры, а также продолжительности перескока между клавишами) записываются с помощью компонента KeyTrac Recorder и отправляются на сервер компании, где происходит их сравнение с построенной ранее моделью. При этом построение модели пользователя способно осуществляться на любом произвольно вводимом тексте, а не только при многократном вводе одних и тех же фраз. Для передачи данных используется предоставляемый KeyTrac API. Далее сервер возвращает свой вердикт в виде булевой величины true / false – соответствуют ли присланные тестовые данные рассматриваемому легитимному профилю либо нет. Для встраивания данного решения в web-сайт также предлагается использовать предоставляемую JavaScript-библиотеку. Разработчики системы утверждают, что их решение нечувствительно к смене используемого языка ввода, а также к смене используемого оборудования. Используемые для этого алгоритмы, а также методы построения модели и дальнейшей классификации не называются. Поскольку все данные динамики работы пользователей с клавиатурой анонимизируются, разработчики рекомендуют использовать свой продукт в том числе и в приложениях электронной коммерции.

### **KeystrokeID (ID Control)**

Другим известным решением, осуществляющим непрерывный анализ динамики работы пользователей с клавиатурой компьютера, является продукт KeystrokeID компании ID Control [8, 23]. Для дальнейшего анализа здесь используются такие характеристики

набора, как промежутки времени между нажатием и отпусканием одной клавиши и промежутки времени между двумя последовательными нажатиями разных клавиш. Сбор данных, характеризующих динамику работы пользователей с клавиатурой, реализуется при помощи Java-апплета. Вначале модель обучается при вводе пользователем логина и пароля, далее происходит дообучение на данных фоновой работы пользователя. Используемые при этом алгоритмы не раскрываются. Данный продукт прост в использовании и используется для защиты компьютеров пользователей от сетевых атак.

### **Scout Analytics**

Интересным решением является продукт компании Scout Analytics [8, 24], используемый для корректной аутентификации пользователей системы хранения и просмотра электронных публикаций. Он позволяет предотвратить написание отзывов о публикациях разными людьми с одного аккаунта, тем самым помогая составить более честную и объективную оценку публикаций. Алгоритмы, используемые в данном решении, запатентованы [25]. Для анализа используются код нажатой клавиши, тип события (нажатие / отжатие), а также время произошедшего события. Далее, в качестве рассчитываемых признаков выступают продолжительности нажатий на отдельные клавиши клавиатуры, продолжительности перескоков при нажатии двух или трех клавиш, а также различные статистики от данных величин (среднее значение, медиана, среднеквадратичное отклонение, максимум, минимум и т.д.). В качестве классификатора используется нейронная сеть. Модель поведения пользователя строится на основании его поведения за последние 60 дней (тем самым, каждый день происходит обновление модели). Также, для дополнительного статистического анализа используется информация об устройстве, с которого был осуществлен вход на сайт, и используемый при этом IP-адрес.

### **TypingDNA**

Одним из последних появившихся на рынке решений в области непрерывной аутентификации пользователей на основе клавиатурного почерка является продукт TypingDNA [26]. Разработчики предлагают использовать данное решение в учебном процессе: в течение семестра студенты работают за компьютерами во время занятий, в то время как в фоновом режиме происходит сбор данных динамики их работы с клавиатурой компьютера и обучение персональных моделей на этих данных. Во время контрольных мероприятий также производятся сбор пришедших от клавиатуры событий и сопоставление этих данных с построенными ранее моделями. Таким образом, удастся выявить недобросовестных студентов, выполняющих контрольные работы

несамостоятельно. Вторая область применения данного продукта – защита клиентов онлайн банковских систем от несанкционированного доступа посторонних лиц к их учетным записям. В качестве характеристических признаков выступают продолжительности нажатий, а также продолжительности перескоков для 44 наиболее часто используемых пользователем клавиш. Разработчики гарантируют высокое качество работы системы при условии, что для построения модели пользователем было напечатано на клавиатуре без продолжительных пауз не менее 100 символов. Сбор данных осуществляется при помощи Java-апплета. Разработчики предоставляют TypingDNA API и заявляют, что данное решение совместимо с большинством современных языков программирования. Используемые в данном продукте алгоритмы машинного обучения не называются.

### **KeystrokeDNA**

Также набирает популярность коммерческое решение KeystrokeDNA [27], разработчики которого позволяют встраивать его в любое web-приложение и советуют использовать данный продукт для защиты компьютеров пользователей от сетевых вторжений. При этом, для анализа поведения пользователей используются ритм и скорость набора текста. Разработчики данного решения заявляют о стабильности его работы как при смене языка ввода, так и при смене используемой клавиатуры. Для установки и встраивания KeystrokeDNA пользователям предоставляется KeystrokeDNA API, а также собственная JavaScript-библиотека и подробная инструкция по их использованию. Более никакой информации о данном продукте в свободном доступе не предоставляется.

Отметим, что помимо решений, основанных только на непрерывной аутентификации пользователей по динамике их работы с клавиатурой персонального компьютера (ноутбука), на рынке также распространены коммерческие продукты, осуществляющие многофакторную динамическую аутентификацию пользователей. Данные решения помимо клавиатурного почерка также анализируют динамику работы пользователей с мышью, с файловой системой и другие различные биометрические показатели. Разработчики данных продуктов (как анализирующих только клавиатурный почерк, так и решений, основанных на многофакторной аутентификации) заявляют, что точность аутентификации их решений достигает порядка 90%.

Рассмотрим наиболее известные решения по непрерывной многофакторной аутентификации пользователей, использующие в качестве одной из своих анализируемых характеристик клавиатурный почерк пользователей.

## **BioCatch**

Одним из решений, осуществляющих динамическую многофакторную аутентификацию пользователей, является широко известный продукт Biocatch [8, 28], анализирующий в режиме онлайн динамику работы пользователей с клавиатурой, мышью и web-ресурсами. При анализе клавиатурного почерка, помимо прочего данное решение позволяет определять, правой или левой является рассматриваемый пользователь, вычисляет размер его руки и также учитывает эти характеристики для определения легитимности пользователя. Всего данное решение вычисляет порядка 2000 характеристических признаков, из которых алгоритмами машинного обучения отбираются 20 наиболее значимых. Но какие именно это признаки и что за алгоритмы используются для их отбора – разработчики умалчивают. Данный продукт предлагается использовать для защиты web-приложений от несанкционированного доступа, а также для защиты пользователей онлайн банковских систем от доступа злоумышленников к их счетам. Аналитикам предоставляется широкий инструмент мониторинга, отображающий на различных графиках возможные риски и аномалии, а также формирующий высокоинформативные отчеты о работе системы.

## **BioTracker (Plurilock)**

Одним из наиболее известных решений по непрерывной многофакторной аутентификации пользователей является продукт BioTracker компании Plurilock [8, 11, 29]. Для того, чтобы с высокой точностью аутентифицировать пользователя, приложению необходимо обучиться на порядка 30 минутах его непрерывной работы с клавиатурой персонального компьютера. Дополнительно для улучшения качества построенной модели также анализируется работа пользователя с мышью. Необходимо заметить, что BioTracker периодически перестраивает модель пользователя, что позволяет учитывать изменение динамики его работы с клавиатурой и мышью в течение длительного периода времени. Разработчики рекомендуют использовать данный продукт как для защиты персональных компьютеров от сетевых атак, так и в приложениях электронной коммерции и государственных структурах – BioTracker может работать и как самостоятельное решение, так и встраиваться в готовые приложения пользователей. Продукт поддерживается всеми популярными сегодня операционными системами (Windows, Linux, Mac OS). Алгоритмы, используемые в данном решении, не называются.

## **CVMetrics, Tickstream (Intensity Analytics)**

Продукты CVMetrics и Tickstream компании Intensity Analytics [8, 30] также широко используются для непрерывной аутентификации пользователей по динамике их

работы с клавиатурой и мышью. Данные решения нашли свое применение в сфере электронной коммерции, энергетике, судебно-медицинской экспертизе, государственных структурах – местах, где защита компьютеров от посторонних вторжений является одной из важнейших задач. Помимо анализа взаимодействия пользователей с мышью и клавиатурой, продукт Tickstream также дополнительно анализирует текстовые документы, с которыми работает пользователь. Для анализа клавиатурного почерка, и CVMetrics, и Tickstream для каждой нажатой клавиши записывают ее номер, соответствующее событие (нажатие / отжатие), а также временную метку, когда данное событие произошло. Разработчики замечают, что основным в их методике является именно анализ динамики работы пользователей с клавиатурой, потому что, в отличие от динамики работы с мышью, клавиатурный почерк является более стабильной характеристикой. Механизм аутентификации основан на вычислении различных статистик, определяющих степень похожести тестируемых данных с данными легитимного профиля. Однако, какие именно статистики вычисляются для этого в данных продуктах – разработчики умалчивают.

### **Symantec VIP**

Высокое качество работы также демонстрирует продукт Symantec VIP [31], предлагающий свои решения как для защиты домашних компьютеров пользователей от несанкционированного доступа, так и для защиты систем электронной коммерции от сетевых вторжений. Динамическая многофакторная аутентификация пользователя осуществляется на основе анализа динамики его работы с клавиатурой и мышью. Процедура аутентификации производится спустя каждые 125 клавиатурных нажатий, при этом характеристики взаимодействия пользователя с мышью используются для уточнения построенной модели. Для принятия решения о легитимности рассматриваемого пользователя используется статистический подход. В случае обнаружения злоумышленника происходит блокировка системы, также система может сфотографировать злоумышленника. Symantec VIP поддерживается операционными системами Windows и Mac OS. В случае встраивания Symantec VIP в web-сайт, он будет корректно работать во всех популярных сегодня браузерах: Internet Explorer, Firefox, Chrome, Safari. Конкретные алгоритмы, используемые в данном решении, разработчиками не называются.

### **NuData Security (Mastercard)**

Интересным решением является продукт NuData Security всемирно известной платежной корпорации Mastercard [32]. Решение NuData Security используется для обеспечения безопасности электронных переводов посредством данной платежной

системы. Аутентификация ведется на основании непрерывного анализа динамики работы пользователя с клавиатурой и мышью. Также при построении модели пользователя учитываются характеристики устройства, с которого осуществляется вход на сайт платежной компании, и время, проведенное пользователем на данном сайте. В случае обнаружения аномалии, все операции по счету блокируются. Разработчики отмечают, что получить доступ к их системе, копируя поведение легитимного пользователя, практически невозможно. Для обучения модели требуется достаточно небольшой промежуток времени – порядка одного посещения пользователем web-сайта компании с осуществлением платежной операции. Данное решение поддерживается операционными системами Windows и Mac OS. Разработчики NuData Security отмечают высокое качество работы их системы и низкий процент ложных срабатываний. При этом, используемые в данном решении алгоритмы машинного обучения не раскрываются.

### **NoPassword**

Также набирает популярность недавно появившееся на мировом рынке коммерческое решение NoPassword [33], осуществляющее непрерывную многофакторную аутентификацию пользователей по динамике их работы с клавиатурой и мышью, а также с файловой системой компьютера или планшета. При этом, и для аутентификации на компьютере, и для аутентификации на планшете используются единые алгоритмы, которые демонстрируют высокое качество работы системы в том числе и при использовании сессий удаленного рабочего стола. Для анализа клавиатурного почерка данное решение вычисляет продолжительности нажатий и продолжительности перескоков при нажатии клавиш клавиатуры. Разработчики предлагают использовать данный продукт для защиты как домашних, так и служебных компьютеров пользователей от несанкционированного доступа злоумышленников. Для встраивания данного решения пользователям предоставляются удобные в использовании NoPassword API и NoPassword SDK. Также администраторам системы предлагаются обладающий широкими возможностями инструмент мониторинга активности пользователей в режиме онлайн и возможность гибкой настройки политик доступа. Данное решение поддерживается всеми наиболее популярными сегодня браузерами и операционными системами. Используемые в данном продукте алгоритмы машинного обучения не называются.

### **Выводы**

В заключение данного подраздела, сформулируем основные выводы по проведенному обзору индустриальных решений.

Динамическая аутентификация пользователей по динамике их работы с клавиатурой персонального компьютера (ноутбука) является достаточно перспективным направлением исследований, и в первую очередь используется для защиты компьютеров пользователей от доступа злоумышленников. На рынке распространены как решения, анализирующие только клавиатурный почерк пользователей, так и продукты многофакторной аутентификации, помимо прочего также учитывающие динамику работы пользователей с мышью, файловой системой, web-ресурсами и многие другие биометрические показатели. Точность работы данных систем (как однофакторных, так и многофакторных) составляет порядка 90%. Достоинством использования многофакторных систем является уточнение модели пользователя, недостатком – увеличение количества затрачиваемых ресурсов системы. Системы аутентификации пользователей по компьютерному почерку используются как для защиты домашних компьютеров пользователей от посторонних вторжений, так и в системах электронной коммерции, государственных структурах, образовании, энергетике. При обнаружении аномалий данные решения способны не только заблокировать дальнейший доступ к системе, но также и сфотографировать злоумышленника.

Механизмы динамической аутентификации пользователей могут как использоваться для предотвращения несанкционированного доступа к web-сайтам (в данном случае будет осуществляться web-сбор данных), так и встраиваться в приложения пользователей или использоваться как самостоятельные решения (при этом данные будут собираться при помощи локального сборщика). Для этого разработчиками предоставляются удобные в использовании JavaScript-библиотеки, J2EE-модули, Java-апплеты, а также собственно разработанные компаниями API и SDK. В качестве собираемых характеристик выступают коды нажатых клавиш, типы произошедших событий (нажатие / отжатие), а также соответствующие им временные метки. Для дальнейшего анализа в коммерческих решениях вычисляются продолжительности нажатия на данные клавиши, а также продолжительности перескоков между клавишами. В ряде решений данные характеристики вычисляются не для всех, а только для наиболее часто используемых пользователем клавиш. В некоторых продуктах дополнительно анализируется скорость набора текста. Используемые алгоритмы уменьшения размерности признакового пространства, а также алгоритмы машинного обучения разработчиками коммерческих систем не раскрываются. В среднем, для построения высококачественной модели, коммерческим решениям требуется обучиться примерно на 15–20 минутах непрерывной работы пользователя с клавиатурой компьютера.



Разработчики отмечают, что со временем модель пользователя устаревает, поэтому ее периодически необходимо перестраивать. Также стоит обратить внимание на то, что при смене используемого языка ввода, а также при смене используемой аппаратуры качество распознавания может падать, что также необходимо учитывать. В некоторых продуктах системным администраторам предоставляются многофункциональные инструменты мониторинга активности пользователей в режиме онлайн, включающие в себя просмотр различных графиков рисков и аномалий, автоматическое составление отчетов, а также предоставляется возможность гибкой настройки и быстрой корректировки политик доступа. Рассмотренные коммерческие решения поддерживаются всеми наиболее распространенными сегодня браузерами и операционными системами.

## **1.2 Сбор и обработка данных**

Поскольку динамическая аутентификация пользователей по клавиатурному почерку может широко применяться и на web-сайтах, и в самостоятельных локальных приложениях, в существующих научно-исследовательских работах, как и в рассмотренных выше коммерческих решениях, рассматривается как осуществляемый web-браузером web-сбор данных, характеризующих динамику работы пользователей с клавиатурой [34 – 42], так и локальный сбор данных средствами операционной системы компьютера [43 – 50].

### **1.2.1 Особенности web-сбора данных**

Технологии, используемые для web-сбора данных, характеризующих динамику работы пользователя с клавиатурой, можно подразделить на две категории в зависимости от использования плагинов [11, 51] – независимых программных модулей, динамически подключаемых к web-браузеру и предназначенных для расширения его функциональных возможностей. К программным средствам, не использующим плагины при web-сборе, относятся встраиваемые в web-страницы JavaScript-программы, а также расширения для web-браузера. К программным средствам, работа которых основывается на использовании в системе специальных плагинов, относятся такие широко известные технологии как Flash и Java-апплеты. Взаимодействие web-браузера с плагином осуществляется через API интерфейс прикладного программирования [52]. Среди наиболее популярных API интерфейсов выделяют NPAPI (Netscape Plugin Application Programming Interface, [53]), PPAPI (Pepper Plugin Application Programming Interface, [54]) и ActiveX [55]. Однако, постепенно Internet Explorer отказывается от использования NPAPI и ActiveX, а интерфейс

PPAPI поддерживается только web-браузерами Google Chrome и Opera, что свидетельствует о том, что не существует универсального решения. Отметим, что наиболее часто используемой для web-сбора и перспективной сегодня является технология JavaScript.

### **JavaScript-программы**

Встраиваемые в web-страницы JavaScript-программы [11, 42, 56] выполняются на стороне клиента и тем самым могут взаимодействовать с внешними ресурсами, собирая информацию об их использовании без использования дополнительного ПО [35, 39]. Однако, стоит иметь в виду, что в разных web-браузерах могут использоваться разные версии JavaScript-интерпретатора, что необходимо учитывать при разработке. Также, авторы [39] говорят о больших задержках при обработке событий нажатия на клавиши клавиатуры при использовании технологий web-сбора, длительность которых существенно зависит от степени загруженности компьютера. Время, проходящее от момента нажатия на клавишу до вызова обработчика данного события, может составлять от десятков до сотен миллисекунд, что значительно ниже скорости обработки событий в локальных сборщиках данных средствами операционной системы – порядка двухсот микросекунд [57]. Также стоит отметить, что при использовании JavaScript-технологий в большинстве web-браузеров нет возможности определить, левые или правые функциональные клавиши (Shift, Ctrl, Alt и т.д.) были нажаты [39], что является серьезным недостатком, поскольку данная информация может существенно помочь аутентифицировать пользователя.

### **Расширения для web-браузера**

Расширения для web-браузера представляют собой программы, расширяющие его функциональные возможности [11, 58]. В отличие от рассмотренных выше JavaScript-программ, встраиваемых в коды web-страниц, расширения для web-браузера предоставляют возможность сбора данных, характеризующих динамику работы пользователя с клавиатурой, при просмотре любых web-страниц, а не только тех, в коды которых заранее вшиты JavaScript-сборщики данных [40]. Это обосновывается способностью расширений модифицировать код просматриваемых web-страниц. В этом заключается их главное отличие от плагинов. Однако, стоит иметь в виду, что разные браузеры предоставляют разные программные интерфейсы для написания расширений (в том числе и требующие реализации на разных языках программирования), что значительно затрудняет создание универсальных решений.

## **Flash**

Flash [11, 59] – это мультимедийная платформа компании Adobe Systems, предназначенная для создания интерактивных web-приложений с богатой векторной, растровой, трехмерной компьютерной графикой и мультимедиа, работающих как внутри, так и вне web-браузера. Для создания приложений используется собственно разработанный язык ActionScript. Adobe Flash используется в web-браузерах Opera и Google Chrome (посредством интерфейса PPAPI), а также в web-браузере Firefox (посредством интерфейса NPAPI). Стоит отметить, что использование Flash сильно замедляет работу браузера, в следствие чего спрос на данный продукт постепенно падает. Заметим, что при использовании технологии Flash собрать данные динамики работы пользователя с клавиатурой получится только внутри самого Flash-объекта.

## **Java-апплет**

Java-апплет [11, 60] является прикладной программой, чаще всего написанной на языке Java в формате байт-кода и выполняющейся в виртуальной машине JVM (Java Virtual Machine, [61]) как внутри web-браузера, так и локально в операционной системе [36 – 38]. Данное решение является кроссплатформенным и поддерживается большинством современных браузеров и операционных систем, его основное назначение – предоставление интерактивных возможностей web-приложениям. Для встраивания Java-апплетов в web-браузер используются интерфейсы NPAPI и ActiveX. Однако, при использовании Java-апплетов не предоставляется возможности сбора данных, пришедших от клавиатуры, в рамках всей просматриваемой пользователем web-страницы: данные можно получить лишь внутри того ее объекта, в котором данный Java-апплет располагается.

### **1.2.2 Особенности локального сбора данных**

Отличительной чертой локального сбора данных, характеризующих динамику работы пользователя с клавиатурой компьютера, является очень быстрая обработка получаемых событий – задержки между приходом сигнала о нажатии клавиши клавиатуры и его обработкой в данном случае значительно меньше аналогичных задержек, возникающих при web-сборе [62]. Однако, стоит иметь в виду, что в отличие от web-подхода, в данной ситуации в силу различных особенностей операционных систем кроссплатформенное решение создать невозможно, в следствие чего для каждой операционной системы необходимо реализовывать свой собственный сборщик.

В случае операционной системы Windows, для перехвата событий от клавиатуры используются специальные hook-и (ловушки, [63]). Они подключаются к операционной системе средствами WinAPI (функциями SetWindowsHookEx и UnhookWindowsHookEx). Простейшее решение для локального сбора данных в Windows будет состоять из двух файлов: динамической библиотеки-перехватчика (.dll), предназначенной для встраивания в процессы Windows для перехвата целевых событий, и исполняемого файла-инжектора (.exe), предназначенного для управления данной библиотекой [45 – 47].

### **1.2.3 Описание собираемых данных**

Анализ существующих работ показал, что как при использовании локального сбора данных, так и при использовании технологий web-сбора, для осуществления дальнейшего анализа поведения пользователя необходимо фиксировать следующие показатели, характеризующие динамику его работы с клавиатурой:

- Код используемой клавиши;
- Тип события (нажатие / отжатие);
- Временная метка, соответствующая произошедшему событию.

Авторы работы [37] также отмечают, что помимо сбора вышеописанных характеристик, существует возможность средствами операционной системы получить имя процесса, в рамках которого осуществляется взаимодействие пользователя с клавиатурой, либо же адрес web-страницы, с которой работает пользователь, и использовать данные характеристики для дополнительного анализа.

### **1.2.4 Фильтрация собранных данных**

Авторы работ [37, 38], а также разработчики ряда коммерческих решений [21, 23, 28] отмечают, что в силу определенных особенностей операционных систем, в системе могут фиксироваться дубликаты пришедших от клавиатуры событий, которые необходимо отфильтровывать для повышения точности строящейся модели. Для этого необходимо для каждой используемой клавиши анализировать все моменты ее нажатий и отпусков.

## **1.3 Выделение набора характеристических признаков**

Процесс выделения признаков, характеризующих динамику работы пользователя с клавиатурой, состоит из двух последовательных этапов [37, 38, 45 – 47, 62]:

- 1) Разбиение потока событий от клавиатуры на временные окна;
- 2) Расчет характеристических признаков для каждого временного окна.

Рассмотрим каждый из данных этапов более подробно.

### **1.3.1 Разбиение потока событий на временные окна**

Под временным окном понимается последовательный фрагмент рассматриваемого потока событий. Размер временного окна является важным параметром, поскольку окна большого размера позволяют передать больше информации о поведении пользователя, но для их построения требуется значительное количество событий, чем может воспользоваться злоумышленник и успеть похитить данные из компьютера до того, как система выявит внутреннее либо внешнее вторжение. Для того, чтобы преодолеть эту проблему, авторы работ [37, 38, 45 – 47, 62] предлагают использовать окна, перекрывающиеся во времени на некоторое число событий. Тем самым, мы получим больше временных окон и большее количество прогнозов для одного и того же потока событий соответственно, что позволит более точно аутентифицировать пользователя. Процент перекрытия окон также является параметром системы, подбираемым экспериментально.

Для разбиения потока событий на временные окна можно использовать следующие методики:

- 1) Разбиение потока событий на временные окна по количеству событий в окне (данную методику предпочтительнее использовать при умеренной работе пользователя с клавиатурой [47, 62]);
- 2) Разбиение потока событий на временные окна по длительности работы пользователя в рамках данного окна (данную методику предпочтительнее использовать, если пользователь активно взаимодействует с клавиатурой на протяжении всего времени своей работы за компьютером). Существующие исследования показывают, что в данном случае оптимальный размер временного окна – 2 минуты [37, 47].

Как в первом, так и во втором случае, можно строить окна фиксированной (по количеству событий либо времени) либо плавающей длины (задавая интервалы для возможных размеров окон).

Также возможно использовать принудительное разбиение на временные окна при возникновении следующих ситуаций [37, 45, 47]:

- Длительная пауза в работе пользователя (в данном случае, события, пришедшие до и после паузы, будут принадлежать разным временным окнам);
- Смена пользователем рабочего процесса либо просматриваемой web-страницы (в данном случае, события из разных процессов или web-страниц будут принадлежать разным временным окнам).

При этом, если после принудительного разбиения возникли окна небольшого размера, их следует убирать из дальнейшего анализа. Принудительное разбиение на окна позволяет избавиться от скачков в поведении пользователя при резкой смене деятельности, тем самым повышая качество аутентификации.

### **1.3.2 Расчет характеристических признаков для каждого временного окна**

Расчет признаков, характеризующих динамику работы пользователя с клавиатурой, может осуществляться как для всех клавиш клавиатуры целиком, так и независимо внутри отдельных групп клавиш (формируемых исходя из физического расположения клавиш на клавиатуре, а также их функционального предназначения, [37, 38, 43, 64]).

Существует несколько методик для выделения признаков, характеризующих клавиатурный почерк пользователя:

- Анализ одиночных нажатий на клавиши клавиатуры;
- Анализ последовательных нажатий на клавиши клавиатуры (то есть анализ нажатий пользователем комбинаций нескольких клавиш – диграфов, триграфов, N-грамм);
- Комбинированный анализ динамики работы пользователя как с одиночными клавишами клавиатуры, так и с их комбинациями.

Подход на основе анализа одиночных нажатий [11, 22, 41, 46, 47, 50, 65, 66] основывается на вычислении продолжительностей нажатий пользователем отдельных клавиш клавиатуры (hold time), а также продолжительностей перескоков между двумя последовательными нажатиями клавиш (промежутков времени между отпусканием предыдущей и нажатием текущей клавиши, latency). Демонстрационный пример, отражающий вычисление данных характеристик, приведен на Рисунке 1. В данном случае, будут вычисляться следующие величины:

- 1)  $t_1^{up} - t_1^{down}$ ,  $t_2^{up} - t_2^{down}$  (промежутки времени между нажатием и отжатием каждой клавиши);
- 2)  $t_2^{down} - t_1^{up}$  (продолжительности перескока между двумя последовательными нажатиями клавиш).

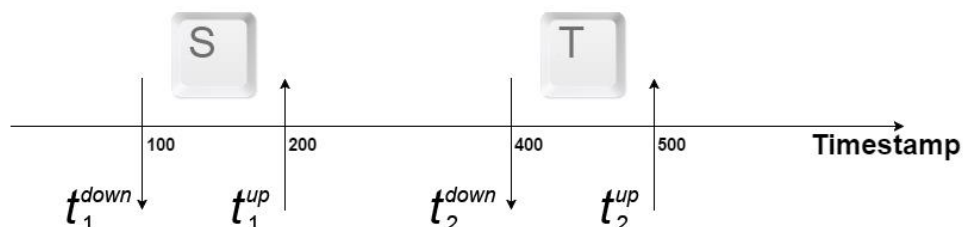


Рисунок 1 — Вычисление признаковых характеристик

Отметим, что авторы работ [37, 38] дополнительно предлагают рассчитывать скорость набора текста пользователем, а также процент использования клавиш из каждой группы.

В рамках подхода на основе анализа последовательных нажатий на клавиши клавиатуры вводится понятие N-грамм – комбинаций N последовательно нажатых пользователем клавиш ( $N > 1$ ). На практике часто используются частные случаи N-грамм при  $N = 2$  и  $N = 3$  (диграфы и триграфы соответственно, [11, 34, 35, 39, 43, 48, 67, 68]). Использование N-грамм, состоящих более чем из трех клавиш, является вычислительно затруднительным и нецелесообразным, поскольку, если на стандартной клавиатуре порядка 105 клавиш, то при  $N = 3$  количество триграфов будет равно  $105^3$ , в следствие чего процессы вычисления признаков, построения модели пользователя и классификации будут занимать длительное время, что является недопустимым в системах онлайн аутентификации пользователей. Рассмотрение одиночных нажатий на клавиши клавиатуры является частным случаем N-грамм (в данном случае  $N = 1$ ).

Для N-грамм возможен расчет следующих характеристик:

- 1)  $t_{i+N-1}^{down} - t_i^{down}$  (промежуток времени между нажатием первой и нажатием последней клавиши);
- 2)  $t_{i+N-1}^{up} - t_i^{down}$  (промежуток времени между нажатием первой и отжатием последней клавиши);
- 3)  $t_{i+N-1}^{down} - t_i^{up}$  (промежуток времени между отжатием первой и нажатием последней клавиши);
- 4)  $t_{i+N-1}^{up} - t_i^{up}$  (промежуток времени между отжатием первой и отжатием последней клавиши),

где  $i$  – порядковый номер N-граммы в рассматриваемом временном окне.

В различных работах вычисляются как все данные характеристики, так и некоторые из них. Например, в работе [43] для диграфов вычисляются только величины  $t_{i+N-1}^{down} - t_i^{down}$ , а в работах [38, 45] рассматриваются величины  $t_{i+N-1}^{down} - t_i^{down}$  и  $t_{i+N-1}^{up} - t_i^{down}$  при  $N = 2$ .

В рамках комбинированного подхода анализируются как одиночные нажатия клавиш клавиатуры пользователем, так и нажатия N-грамм [11, 36 – 38, 40, 42, 45, 49, 62, 64, 69 – 71]. Наиболее перспективным является именно данный подход, поскольку он позволяет учесть больше индивидуальных особенностей пользователя при работе с клавиатурой, анализируя как его работу с отдельными клавишами, так и с их комбинациями.

Во всех рассмотренных методиках в итоговый вектор признаков входят вычисленные для каждой одиночной клавиши или N-граммы средние значения вышеописанных признаков, а также дополнительно могут рассчитываться различные статистики от вышеописанных величин (минимальное, максимальное значения, медиана, среднее отклонение и т.д., [37, 38]).

## **1.4 Постобработка векторов признаков**

Большинство современных методов машинного обучения сильно чувствительны к шкалированию входных данных. Для того, чтобы решить данную проблему, в существующих работах вводится шаг нормализации [12, 36, 66, 71, 72] (или ее подвиды – стандартизации [37, 38]) данных. Нормализация данных выполняет отображение исходного пространства признаков таким образом, что их значения после этого преобразования принадлежат отрезку  $[0, 1]$ . После же процедуры стандартизации каждый признак имеет среднее значение, равное нулю, и дисперсию, равную единице. Благодаря этому, ни один из используемых признаков не будет доминировать в строящейся модели, и классификатор корректно учтет каждый из них.

Суть стандартизации заключается в осуществлении следующего преобразования:

$$x \rightarrow \frac{x - Ex}{\sqrt{Dx}}, \quad (1)$$

где  $x$  – значение рассматриваемого признака, а  $Ex$  и  $Dx$  – математическое ожидание и дисперсия данного признака соответственно, полученные на обучающем наборе.



## **1.5 Сокращение размерности пространства признаков**

Современные клавиатуры в среднем содержат порядка 100 клавиш, и при  $N=2$  мы получаем 10000 различных попарных комбинаций клавиш, для каждой из которых рассчитываются различные статистики – размерность полученного признакового пространства получается достаточно большой, при этом само пространство признаков получается разреженным и не все используемые признаки оказывают значимое влияние на итоговый результат классификации. Более того, некоторые признаки могут являться шумовыми. Для решения данной проблемы применяются методы сокращения размерности признакового пространства.

Наиболее часто используемыми методами уменьшения размерности пространства признаков в задаче распознавания пользователя по клавиатурному почерку являются:

- Методы, основанные на анализе используемых пользователем клавиш;
- Метод главных компонент;
- Эвристические методы сокращения размерности признакового пространства.

Рассмотрим данные методы более подробно.

### **1.5.1 Методы, основанные на анализе используемых пользователем клавиш**

Методы, основанные на анализе используемых пользователем клавиш [34, 35, 39, 43], производят отбор наиболее часто используемых пользователем клавиш и  $N$ -грамм, для которых далее рассчитываются описанные выше признаковые характеристики. Количество отбираемых для дальнейшего анализа клавиш клавиатуры является параметром системы, подбираемым экспериментально. Авторы работ [34, 35] утверждают, что для успешного распознавания пользователей достаточно использовать характеристические признаки для 50 наиболее часто используемых пользователем диграфов. Поскольку каждый пользователь имеет свой собственный набор наиболее часто используемых клавиш, данный метод позволяет учесть больше индивидуальных особенностей пользователей и тем самым повысить качество аутентификации.

### **1.5.2 Метод главных компонент**

Метод главных компонент (Principal Component Analysis, PCA) является одним из наиболее распространенных методов сокращения размерности входных данных [11, 12,

45, 46, 73, 74]. Задача данного метода – поиск подпространств меньшей размерности, в ортогональной проекции на которые разброс данных будет максимальным. Таким образом, PCA проецирует данные на новую координатную систему меньшей размерности, определяемую собственными векторами и собственными числами матрицы ковариации (заметим, что ковариация двух случайных величин является мерой их линейной зависимости). После того, как собственные векторы и собственные числа найдены, собственные числа сортируются в порядке убывания. Благодаря этому можно получить компоненты в порядке уменьшения их значимости. Собственный вектор, соответствующий наибольшему собственному числу – самая главная компонента набора данных. Он выражает самые существенные отношения между координатами. Поэтому главные компоненты получаются умножением строк из собственных векторов на отсортированные собственные значения матрицы ковариации. Количество отбираемых главных компонент является параметром алгоритма. В существующих работах метод главных компонент обычно применяется в комбинации с методом машинного обучения SVM.

Демонстрационный пример, отражающий принцип работы данного метода, представлен на Рисунке 2.

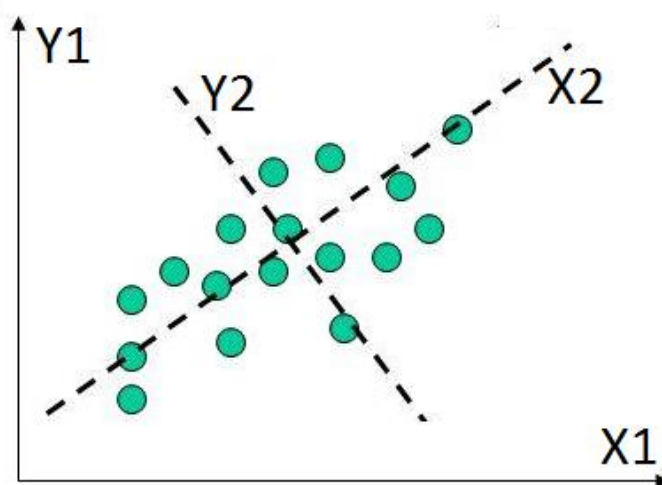


Рисунок 2 — Демонстрация принципа работы метода главных компонент

Отметим, что одним из основных недостатков данного метода является прямо пропорциональная зависимость размера ковариационной матрицы и размерности входных данных, вследствие чего поиск собственных векторов может быть достаточно затруднительным для данных высокой размерности. Стоит помнить, что данный метод не всегда эффективно снижает размерность входных данных при заданных ограничениях на

точность. Также заметим, что прямые и плоскости не всегда обеспечивают хорошую аппроксимацию.

### **1.5.3 Эвристические методы сокращения размерности признакового пространства**

Наиболее известными эвристическими методами уменьшения размерности данных, применяющимися в задаче аутентификации пользователей по клавиатурному почерку, являются генетический алгоритм (Genetic Algorithm, GA [75]), метод роя частиц (Particle Swarm Optimization, PSO, [76]), а также муравьиный алгоритм (Ant Colony Optimization, ACO, [77]).

#### **Генетический алгоритм**

Генетический алгоритм [36, 44, 72, 78 – 80] – это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путем последовательного подбора, комбинирования и вариации искомых параметров при изучении влияния на итоговый отклик с использованием механизмов, аналогичных естественному отбору в природе. В случае решения задачи отбора признаков, каждая хромосома в данном алгоритме представляет собой набор признаков, где каждый ген характеризует наличие признака: 0 – отсутствует, 1 – присутствует. Главным элементом данного алгоритма является функция приспособленности гена (фитнес-функция). Существует несколько способов выбора фитнес-функции. Например, один из них основан на алгоритме классификации k-ближайших соседей. Фитнес-функция формируется в предположении, что после отбрасывания неинформативного признака набор ближайших соседей изменяется минимально. Другой способ задания фитнес-функции – вычисление ее как точности классификации. Следует учитывать, что некоторые параметры данного алгоритма выбираются произвольным образом, поэтому его необходимо запускать много раз для нахождения оптимального решения.

#### **Метод роя частиц**

Метод роя частиц [44, 72, 81] является методом численной оптимизации, для использования которого не требуется знать точного градиента оптимизируемой функции. Данный метод оптимизирует решающую функцию, поддерживая популяцию возможных решений, называемых частицами, и перемещает эти частицы в пространстве согласно простой формуле. При этом запоминаются наилучшие места из уже посещенных. Наилучшее место напрямую определяется качеством классификации при данном наборе признаков. Поиск наилучших положений осуществляется всей колонией пчел (частиц) на

исходном пространстве признаков. Параметрами данного алгоритма являются количество пчел-агентов, максимальное количество итераций, общее количество признаков и количество признаков, которое необходимо оставить. Достоинствами данного метода являются отсутствие склонности к заикливанию в локальных оптимумах (поскольку алгоритм основан на случайном поиске), а также тот факт, что поиск наилучшего решения основывается на решениях агентов всей колонии пчел.

### **Муравьиный алгоритм**

Суть муравьиного алгоритма заключается в моделировании поведения муравьиной колонии [64, 72, 78, 79]. В реальном мире муравьи ходят в случайном порядке и по нахождению продовольствия возвращаются в свою колонию, прокладывая феромонами тропы. Если другие муравьи находят такие тропы, то они, вероятнее всего, пойдут по ним. Они также откладывают феромоны, в следствие чего на коротких тропах концентрация феромонов будет большей и все муравьи будут выбирать данный путь. В случае решения задачи выбора информативных признаков, набор признаков представляется в виде графа, в котором каждый узел – это признак. Параметром алгоритма, задаваемым в начале работы, является количество информативных признаков, которые необходимо найти. Муравей останавливается тогда, когда пройдено необходимое количество признаков. На каждом шаге происходит испарение феромона. На каждой итерации алгоритма выбирается набор признаков (путь муравья) с минимальной ошибкой. Критериями окончания работы алгоритма являются пройденное необходимое количество итераций, а также достижение порога минимальной ошибки. Таким образом, количество феромона на каждой грани обратно пропорционально количеству ошибок, полученных при классификации объектов по этому признаку.

Данные эвристические алгоритмы позволяют выявлять как отдельные информативные признаки, так и группы признаков, совместное использование которых дает уменьшение ошибки классификации. Основными недостатками данных алгоритмов являются их вычислительная трудоемкость и низкая масштабируемость, поскольку они основаны на методах случайного поиска. На практике вышеперечисленные эвристические алгоритмы обычно применяются в комбинации с методом опорных векторов (SVM).

## **1.6 Построение модели пользователя**

Следующим этапом решения задачи аутентификации пользователя по динамике его работы с клавиатурой компьютера является построение модели его поведения. Отметим

основную сложность данной задачи: она является задачей одноклассовой классификации, поскольку нам доступны данные динамики работы с клавиатурой только легитимного пользователя. Данные клавиатурного почерка нелегитимных пользователей нам недоступны. При этом, в обучающей выборке может также присутствовать определенный небольшой процент исключений – объектов или событий в выборке, чьи признаки или их комбинации не соответствуют зависимостям, характерным для остальных объектов или событий в данной выборке. Для разных алгоритмов машинного обучения формальное понятие исключения может быть разным.

Наиболее часто применяющимися в данной задаче методами построения модели пользователя являются:

- одноклассовый метод  $k$ -ближайших соседей;
- одноклассовый метод опорных векторов (SVC, Single Class SVM);
- модель гауссовых смесей;
- нейронные сети.

Рассмотрим данные методы более подробно.

#### **1.6.1 Одноклассовый метод $k$ -ближайших соседей**

В одноклассовом методе  $k$ -ближайших соседей (One-Class KNN, [37, 38, 41, 43, 47, 66, 68]) рассматриваются расстояния между объектами в базе данных. Объект  $O$  считается исключением, если как минимум  $p$ -я часть всех объектов базы находится на расстоянии, большем  $D$ , от данного объекта  $O$ . Параметры  $p$  и  $D$  подбираются экспериментально с целью достижения наилучшего качества распознавания.

Данный алгоритм обладает достаточно простой реализацией и устойчив к аномальным выбросам. Однако стоит иметь в виду, что набор данных, используемый для алгоритма, должен быть репрезентативным.

#### **1.6.2 Одноклассовый метод опорных векторов (SVC, Single Class SVM)**

Одноклассовый метод опорных векторов (SVM, [16, 41 – 43, 45, 46, 48 – 50, 64, 69, 80]) является одним из наиболее часто используемых классификаторов и стабильно показывает высокие результаты (ROC-индекс) в задаче распознавания пользователей по динамике их работы с клавиатурой. Данный алгоритм имеет две широко известные разновидности: SVC и Single Class SVM.

В методе SVC (Support Vector Clustering) объекты из исходного множества неявно отображаются с помощью потенциальной (kernel) функции в пространство характеристик высокой размерности (Reproducing Kernel Hilbert Space, RKHS), где далее происходит поиск гиперсферы минимального радиуса, содержащей внутри себя «основную часть» образов объектов из исходного множества. Исключениями считаются объекты, чей образ лежит за пределами найденной гиперсферы. Таким образом, в данном методе решается следующая задача оптимизации:

$$\min_{\xi \in \mathbb{R}^N, R \in \mathbb{R}, a \in H} \left[ R^2 + \frac{1}{vN} \sum_{i=1}^N \xi_i \right], \quad (2)$$

$$\|\varphi(x_i) - a\|_H^2 \leq R^2 + \xi_i, \forall i \in [1, N], \quad (3)$$

где  $R$  – радиус построенной гиперсферы;  $a$  – центр гиперсферы;  $N$  – число объектов в выборке  $X$ ,  $0 < v \leq 1$  – процент исключений;  $\xi_i$  – дополнительные переменные. Для решения данной задачи предлагается использовать метод множителей Лагранжа. В результате бинарная решающая функция будет иметь следующий вид:

$$f(z) = \text{sgn} \left( R^2 - \sum_{i,j=1}^N \beta_i \beta_j K(x_i, x_j) + 2 \sum_{i=1}^N \beta_i K(x_i, z) - K(z, z) \right), \quad (4)$$

где  $\beta_i$  – множители Лагранжа:  $\beta_i = \frac{1}{vN}$  для исключений,  $0 < \beta_i < \frac{1}{vN}$  для граничных объектов и  $\beta_i = 0$  для остальных объектов;  $z$  – тестируемый вектор признаков;  $K$  – используемая потенциальная функция.

Метод Single Class SVM аналогичен методу SVC: он находит гиперплоскость, отделяющую «основную часть» образов объектов от начала координат. Исключениями считаются объекты, чей образ лежит ближе, чем найденная гиперплоскость, к началу координат. При использовании потенциальной функции RBF результаты работы методов Single Class SVM и SVC получаются достаточно схожими.

Достоинствами данных алгоритмов являются возможность использования даже небольшого набора данных для обучения, а также возможность получить функцию классификации с минимальной верхней оценкой ожидаемого риска. Основным недостатком данных методов является чрезмерная простота найденных границ для определения исключений в RKHS, которые не учитывают в полной мере зависимости между признаками. Также стоит отметить, что как и у любого метода опорных векторов, результирующая модель в методах SVC и Single Class SVM зависит только от наблюдений, лежащих на границе, и не дает возможности корректно оценить степень аномальности для наблюдений внутри границы.

### 1.6.3 Модель гауссовых смесей

Гауссовы смеси (Gaussian Mixture Models, GMM, [11, 35]) – это совокупность распределений нормальной величины. В работах [35, 41, 43, 65] утверждается, что распределение данных динамики работы с клавиатурой для всех пользователей – нормальное, но для каждого пользователя это распределение свое. Для каждого тестируемого набора данных его распределение сравнивается с известными нормальными компонентами распределения данных легитимного пользователя.

Плотность вероятности  $P(x)$  случайной величины  $x$  при нормальном распределении вычисляется по формуле:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad (5)$$

где  $\sigma^2$  – дисперсия случайной величины;  $\mu$  – математическое ожидание случайной величины.

Полностью модель гауссовой смеси определяется векторами математического ожидания, ковариационными матрицами и весами смесей для каждого компонента модели. Смеси гауссовых распределений способны очень точно аппроксимировать произвольные распределения. Однако данный метод обладает достаточно сложной реализацией и является вычислительно трудным. Также не стоит забывать, что выбор количества используемых в модели смесей сильно влияет на результат работы данного алгоритма.

### 1.6.4 Нейронные сети

Нейронная сеть [12, 36, 66, 67, 69, 71, 82 – 84] является математической моделью биологических нейронных сетей, состоящих из набора взаимодействующих между собой нейронов, каждый из которых получает на вход некий набор сигналов (функции выхода предшествующих нейронов) и вырабатывает на выходе результирующий сигнал. Первым нейронам сети на вход подаются элементы векторов признаков рассматриваемых объектов. В результате работы последнего нейрона мы получаем число, являющееся результатом классификации. По его значению мы можем определить, принадлежит ли рассматриваемый объект легитимному классу или нет.

Нейронные сети позволяют эффективно строить нелинейные зависимости, более точно описывающие наборы данных, а также позволяют эффективнее сжимать данные. Также одним из главных достоинств нейронных сетей является их способность к дообучению.

Существует большое количество видов нейронных сетей и их модификаций. Наиболее известными и часто использующимися из них являются ограниченная машина Больцмана [71, 82], RBF-нейронная сеть [66], FF-MLP нейронная сеть [66, 67], а также рекуррентная нейронная сеть [70].

### **1.6.5 Сравнение качества работы используемых подходов**

Для оценки качества распознавания пользователей по динамике их работы с клавиатурой в существующих работах используются следующие метрики [12]:

- Точность классификации (Accuracy) – доля верных срабатываний классификатора;
- FAR (False Acceptance Rate) – отношение количества случаев, когда злоумышленник был распознан системой как зарегистрированный пользователь, к общему числу рассматриваемых наблюдений (величина ошибки второго рода);
- FRR (False Rejection Rate) – доля ситуаций, когда зарегистрированный пользователь был распознан системой как злоумышленник (величина ошибки первого рода);
- EER (Equal Error Rate) – процентное количество ошибок, при котором FAR совпадает с FRR (при варьировании порога для принятия решения классификатором);
- ROC-кривая (Receiver Operating Characteristic) – графическая интерпретация качества работы бинарного классификатора. Данная характеристика отражает зависимость величин TPR (доли верно распознанных легитимных пользователей) и FPR (доли верно отвергнутых злоумышленников);
- AUC (Area Under Curve) – количественная интерпретация ROC-кривой, обозначающая площадь под ней.

Достигнутые в рассмотренных выше научных публикациях результаты в области непрерывной аутентификации пользователей по клавиатурному почерку представлены в Таблице 1. Отметим, что тестовые наборы, используемые в данных работах, различны.



Таблица 1 — Сравнение качества работы используемых подходов

Работа	Используемые признаки	Используемый метод сокращения размерности признакового пространства	Используемый метод машинного обучения	Качество аутентификации
[35]	Характеристики диграфов	Отбор 50 наиболее часто используемых диграфов	GMM	Accuracy = 0.94 EER = 5.88
[37]	Продолжительности нажатий и перескоков для одиночных клавиш; Характеристики диграфов (PP); Статистика нажатий по группам клавиш; Скорость набора текста	—	One-Class KNN	Accuracy = 0.96
[38]	Продолжительности нажатий и перескоков для одиночных клавиш; Характеристики диграфов (RP, PP); Статистика нажатий по группам клавиш; Скорость набора текста	—	One-Class KNN	Accuracy = 0.96 (без смены оборудования) Accuracy = 0.76 (со сменой оборудования)
[41]	Продолжительности нажатий и перескоков для одиночных клавиш	—	One-Class SVM	EER = 7.67
			GMM	EER = 3.68
			One-Class KNN	EER = 10.36
[42]	Продолжительности нажатий и перескоков для одиночных клавиш; Характеристики диграфов	—	One-Class SVM	EER = 2.94

Работа	Используемые признаки	Используемый метод сокращения размерности признакового пространства	Используемый метод машинного обучения	Качество аутентификации
[43]	Характеристики диграфов (PP); Статистика нажатий по группам клавиш	Отбор наиболее часто используемых диграфов	GMM	EER = 7.96
			One-Class KNN	EER = 5.97
			One-Class SVM	EER = 7.00
[45]	Продолжительности нажатий и перескоков для одиночных клавиш; Характеристики диграфов (RP, PP)	Метод главных компонент	One-Class SVM	FAR = 7.46 FRR = 14.99 ROC AUC = 0.87
[46]	Продолжительности нажатий и перескоков для одиночных клавиш	Метод главных компонент	One-Class SVM	FAR = 15.52 FRR = 22.94
[47]	Продолжительности нажатий и перескоков для одиночных клавиш	—	One-Class KNN	EER = 0.04
[48]	Характеристики диграфов и триграфов	—	One-Class SVM	EER = 9.0 ROC AUC = 0.85
[49]	Продолжительности нажатий и перескоков для одиночных клавиш; Характеристики диграфов	—	One-Class SVM	EER = 10.25
[50]	Продолжительности нажатий и перескоков для одиночных клавиш	—	One-Class SVM	Accuracy = 0.85
[64]	Продолжительности нажатий и перескоков для одиночных клавиш;	Муравьиный алгоритм	One-Class SVM	FAR = 0.02 FRR = 0.375

Работа	Используемые признаки	Используемый метод сокращения размерности признакового пространства	Используемый метод машинного обучения	Качество аутентификации
	Характеристики диграфов; Статистика нажатий по группам клавиш			
[65]	Продолжительности нажатий и перескоков для одиночных клавиш	—	GMM	EER = 0.09
[66]	Продолжительности нажатий и перескоков для одиночных клавиш	—	One-Class KNN	EER = 30.0
			RBF-нейронная сеть	EER = 17.7
			FF-MLP нейронная сеть	EER = 12.8
[67]	Характеристики диграфов	—	FF-MLP нейронная сеть	FAR = 0.02 FRR = 4.82
[68]	Характеристики диграфов	—	One-Class KNN	EER = 4.53
[69]	Продолжительности нажатий и перескоков для одиночных клавиш; Характеристики диграфов	—	One-Class SVM	Accuracy = 0.84
[70]	Продолжительности нажатий и перескоков для одиночных клавиш; Характеристики диграфов	—	RNN нейронная сеть	Accuracy = 0.91
[71]	Продолжительности нажатий и перескоков	—	Ограниченная машина	Accuracy = 0.90

<b>Работа</b>	<b>Используемые признаки</b>	<b>Используемый метод сокращения размерности признакового пространства</b>	<b>Используемый метод машинного обучения</b>	<b>Качество аутентификации</b>
	для одиночных клавиш; Характеристики диграфов		Больцмана	

Среди основных проблем, проявляющихся при аутентификации пользователей по динамике их работы с клавиатурой компьютера и сильно влияющих на итоговый результат распознавания, авторы работ [36, 38, 62, 85 – 87] выделяют следующие:

- Современные решения не предлагают способа построения стабильного по времени признакового пространства, в следствие чего наблюдается падение качества распознавания пользователей с течением времени, в том числе и при смене используемой аппаратуры (Поскольку клавиатуры разных производителей имеют различное расположение клавиш, а также различные технические характеристики, авторы работ [38, 62, 85 – 87] отмечают резкое падение качества распознавания (на 20–25%) при смене используемой аппаратуры). Для частичного решения данной проблемы авторы работ [62, 86] предлагают периодически обновлять построенную модель клавиатурного почерка пользователя;
- Поскольку в реальных ситуациях нам доступны данные только легитимного класса, а примеры целевого нелегитимного класса либо отсутствуют, либо не отмечены в обучающей выборке, обычные методы подбора метапараметров алгоритмов построения одноклассовой модели пользователя с использованием валидационного набора, содержащего размеченные примеры обоих классов (как для методов обучения с учителем) использовать невозможно. Для частичного решения данной проблемы авторы работы [36] предлагают использовать искусственную генерацию объектов нелегитимного класса.

## 1.7 Выводы

Из проведенного обзора существующих подходов к анализу клавиатурного почерка пользователей, применяемых в современных научных работах и программных системах, можно сделать следующие выводы:

- Динамическая аутентификация пользователей на основе анализа динамики их работы с клавиатурой компьютера является достаточно перспективным направлением исследований и широко применяется для обеспечения безопасности как домашних компьютеров пользователей, так и компьютеров крупнейших корпораций, а также для предотвращения несанкционированного доступа злоумышленников к web-сайтам.
- Для осуществления анализа клавиатурного почерка пользователя, необходимо фиксировать следующие характеристики его ввода: код используемой клавиши, тип события (нажатие / отжатие), а также временную метку, соответствующую произошедшему событию. Помимо этого, дополнительно можно фиксировать имя процесса, в рамках которого осуществляется взаимодействие пользователя с клавиатурой, а также адрес web-страницы, с которой работает пользователь.
- После этого, необходимо разбивать поступающий от клавиатуры поток событий на временные окна и вычислять характеристические признаки отдельно для каждого временного окна. Перспективным является использование комбинированного набора характеристических признаков, включающего в себя как характеристики работы пользователя с одиночными клавишами, так и с комбинациями нескольких клавиш – N-граммами. Эффективно использовать N-граммы при  $N=2$  (диграфы). Также для повышения точности модели перспективно рассматривать скорость набора текста пользователем, а также разбивать все клавиши клавиатуры на группы и рассчитывать процент использования клавиш из каждой группы.
- Поскольку размерность признакового пространства для данной задачи является слишком большой, необходимо использовать методы сокращения размерности признакового пространства. Наиболее перспективным из них и показывающим высокое качество работы является отбор признаков на основе анализа используемых пользователем клавиш (отбор наиболее часто используемых одиночных клавиш и диграфов).

- Для постобработки признаков в существующих работах предлагается использовать стандартизацию признаков, способствующую сведению к минимуму доминирования каких-либо признаков в строящейся модели.
- В качестве метода построения модели пользователя наивысшее качество работы показывают методы One-Class SVM (SVC), One-Class KNN, а также нейронные сети. Однако, качество аутентификации не является достаточно высоким (точность составляет порядка 85–90%, ROC AUC ниже 0.90). Дополнительно, качество работы одноклассовых классификаторов сильно зависит от выбора значений метапараметров, которые тяжело подобрать в силу отсутствия примеров нелегитимного класса.
- Зачастую необходимо уметь определять степень аномальности поведения пользователя не за короткий промежуток времени, соответствующий одному вектору признаков, а за длительный период его работы за компьютером (например, за целую сессию работы пользователя за компьютером). В существующих работах решение данной проблемы не предлагается.
- Современные подходы не способны выделять наиболее стабильные по времени признаки, в следствие чего наблюдается проблема падения качества распознавания пользователей с течением времени (в том числе и при смене используемого оборудования).

Сформулированные выводы являются обоснованием направлений дальнейших исследований:

- *Выбор признакового пространства, исследование и разработка методов постобработки признаков и сокращения размерности признакового пространства с целью выделения наиболее информативных и стабильных по времени признаков.*
- *Исследование и разработка методов построения модели пользователя, позволяющих достичь высокого качества распознавания (более 0.90 ROC AUC) в данной задаче, а также разработка метода подбора значений метапараметров алгоритмов одноклассовой классификации, способного выявлять оптимальные значения параметров используемых алгоритмов.*
- *Исследование и разработка методов оценки аномальности поведения пользователя на основе анализа целых сессий работы за компьютером.*

## 2 ПРЕДОБРАБОТКА ДАННЫХ, ХАРАКТЕРИЗУЮЩИХ ДИНАМИКУ РАБОТЫ ПОЛЬЗОВАТЕЛЕЙ С КЛАВИАТУРОЙ КОМПЬЮТЕРА

*При работе (при подготовке) над данным разделом диссертации использованы следующие публикации автора, в которых, согласно Положению о присуждении ученых степеней в МГУ, отражены основные результаты, положения и выводы исследования:*

- *Методы поиска исключений в потоках сложноструктурированных данных / М. А. Казачук, М. И. Петровский, И. В. Машечкин, О. Е. Горохов // Вестник Московского университета. Серия 15: Вычислительная математика и кибернетика. – 2019. – № 3. – С. 17-28.*

В данной диссертационной работе проводится исследование и разработка методов обнаружения аномального поведения пользователей на основе анализа динамики их работы с клавиатурой персонального компьютера (ноутбука).

Формально, анализируемые в данной задаче данные  $H(U_i)$  представляют собой последовательности событий  $A_{ij}$ , пришедших в операционную систему / web-браузер от клавиатуры в течение работы рассматриваемого пользователя  $U_i$  за компьютером (отметим, что в рамках поставленной задачи мы рассматриваем открытое множество пользователей  $U$ ):

$$H(U_i) = (A_{i1}, A_{i2}, \dots, A_{iN}), \quad (6)$$

где каждое событие  $A_{ij}, j \in [1, N]$  представляется тройкой характеристик:

$$A_{ij} = \langle key_{ij}, type_{ij}, time_{ij} \rangle, \quad (7)$$

в которой  $key_{ij}$  – код используемой клавиши;  $type_{ij}$  – тип произошедшего события (нажатие / отжатие),  $time_{ij}$  – временная метка, соответствующая данному событию.

Необходимо на основе имеющейся последовательности событий  $H(U_i)$  сформировать набор векторов признаков  $F(U_i)$ , характеризующих динамику работы пользователя  $U_i$  с клавиатурой компьютера:

$$H(U_i) = (A_{i1}, A_{i2}, \dots, A_{iN}) \rightarrow F(U_i) = (V_{i1}, V_{i2}, \dots, V_{iM}), \quad (8)$$

где каждый вектор признаков  $V_{ij}$  представляет собой числовой вектор, состоящий из вещественных чисел. Предлагается разбивать поток собираемых данных на временные окна (последовательные фрагменты следующих друг за другом событий) и строить свой

вектор признаков для каждого окна:  $(A_{ij_1}, \dots, A_{ij_G}) \rightarrow V_{ij}$ , где  $A_{ij_1}, \dots, A_{ij_G}$  – последовательность произошедших в рамках данного временного окна событий.

Стоит обратить внимание на то, что с течением времени характер ввода пользователя может меняться, что влечет за собой снижение качества аутентификации (в том числе и при смене используемого оборудования). Поэтому необходимо построить пространство стабильных по времени признаковых характеристик.

*Задачей данного раздела является исследование и разработка алгоритмов предобработки данных, характеризующих динамику работы пользователей с клавиатурой компьютера, включающих в себя выбор используемого признакового пространства, а также исследование и разработку методов постобработки признаков и сокращения размерности признакового пространства, совместное использование которых позволит рассчитать векторы информативных и стабильных по времени характеристических признаков  $V_{ij}$  для рассматриваемого пользователя  $U_i$ .*

*К предлагаемому решению поставленной задачи предъявляются следующие требования:*

- полученное результирующее признаковое пространство должно с высокой точностью описывать клавиатурный почерк пользователя (при его использовании должно достигаться достаточно высокое качество распознавания);
- полученное результирующее признаковое пространство также должно быть стабильно по времени.

## **2.1 Описание используемых для исследования наборов данных**

Поскольку динамическая аутентификация пользователей по динамике их работы с клавиатурой компьютера широко применяется как в web-приложениях [34 – 42], так и в локальных приложениях пользователей [43 – 50], необходимо разработать алгоритмы, показывающие высокое качество работы в обеих данных областях. Поэтому, для экспериментальной проверки предлагаемых подходов к решению поставленной задачи, необходимо использовать как наборы данных, характеризующие динамику фоновой работы пользователя с локальными приложениями, так и наборы данных, характеризующие клавиатурный почерк пользователя при работе в web-браузере.



В рамках данного исследования использовались четыре набора тестовых данных:

### **Набор данных 1 (локальные данные)**

Данный набор «Villani» [38] представляет собой данные динамики работы 144 пользователей с клавиатурой, собранные в фоновом режиме во время повседневной работы за компьютером (дополнительно, пользователи писали эссе в свободной форме на заданные темы). В среднем, каждый пользователь проработал за компьютером порядка одного рабочего дня. Помимо информации о динамике клавиатурного ввода, по каждому пользователю также собиралась следующая информация: тип используемого компьютера (персональный компьютер / ноутбук), пол, возраст пользователя и наличие осведомленности о сборе данных. Информация по типу используемых клавиатур владельцами набора данных не предоставляется. По каждому пользователю было собрано порядка 27000 событий клавиатурного ввода и построено в среднем по 50 векторов характеристических признаков (каждый вектор признаков описывал динамику работы пользователя с клавиатурой в рамках порядка 300 последовательных событий клавиатурного ввода).

### **Набор данных 2 (локальные данные)**

Данный набор представляет собой данные динамики работы 20 пользователей с клавиатурой, собранные в фоновом режиме во время повседневной работы за компьютером. В среднем, каждый пользователь проработал за компьютером порядка 10 часов: по одному часу в течение 10 дней. Использовались мембранные клавиатуры с цифровым блоком и цилиндрическими клавишами. По каждому пользователю было собрано порядка 65500 событий клавиатурного ввода и построено в среднем по 97 векторов характеристических признаков (каждый вектор признаков описывал динамику работы пользователя с клавиатурой в рамках порядка 300 последовательных событий клавиатурного ввода).

### **Набор данных 3 (web-данные)**

Данный набор представляет собой данные динамики работы 20 пользователей с клавиатурой, собранные при помощи технологии JavaScript в фоновом режиме работы пользователей на форуме в web-браузере. В среднем, каждый пользователь проработал за компьютером порядка 10 часов: по одному часу в течение 10 дней. Использовались мембранные клавиатуры с цифровым блоком и цилиндрическими клавишами. По каждому пользователю было собрано порядка 65000 событий клавиатурного ввода и построено в среднем по 95 векторов характеристических признаков (каждый вектор

признаков описывал динамику работы пользователя с клавиатурой в рамках порядка 300 последовательных событий клавиатурного ввода).

#### **Набор данных 4 (локальные данные)**

Данный набор данных представляет собой данные динамики работы 10 пользователей с клавиатурой компьютера двух видов (в обоих случаях данные собирались в фоновом режиме во время повседневной работы пользователей за компьютером):

- Данные динамики работы пользователей за общим компьютером в течение трех дней. Использовалась мембранная клавиатура с цифровым блоком и плоскими клавишами. По каждому пользователю было собрано порядка 43000 событий клавиатурного ввода и построено в среднем по 78 векторов характеристических признаков (каждый вектор признаков описывал динамику работы пользователя с клавиатурой в рамках порядка 300 последовательных событий клавиатурного ввода);
- Данные динамики работы пользователей за разными компьютерами (каждый пользователь работал за своим собственным компьютером) в течение трех дней. Использовались мембранные клавиатуры с цифровым блоком и цилиндрическими клавишами. По каждому пользователю было собрано порядка 37000 событий клавиатурного ввода и построено в среднем по 70 векторов характеристических признаков (каждый вектор признаков описывал динамику работы пользователя с клавиатурой в рамках порядка 300 последовательных событий клавиатурного ввода).

Подробное описание формата данных наборов данных приведено в Разделе 4 данной работы. Все данные наборы содержат такие характеристики ввода пользователя, как код нажатой клавиши, тип события (нажатие / отжатие), а также временную метку, соответствующую произошедшему событию. Также дополнительно содержится информация о названии процесса или адресе web-страницы, в рамках которых происходило взаимодействие пользователя с клавиатурой компьютера. В среднем, по каждому пользователю было собрано порядка 47500 событий клавиатурного ввода. Отметим, что все используемые тестовые наборы содержат введенный пользователями текст как на русском, так и на английском языках. С целью проверки стабильности по времени разработанного решения, сбор данных происходил в разное время суток (дополнительно, для проверки работы системы при смене используемого оборудования использовался Набор данных 4). При использовании первого, второго и третьего тестовых наборов модель по каждому пользователю строилась на первой половине его собранных

данных. Для тестирования использовалась вторая половина данных динамики его работы с клавиатурой, а также данные всех других пользователей. При использовании четвертого тестового набора построение модели пользователя осуществлялось на первой части набора (данные, собранные на общем компьютере) и тестировалось на второй части набора (данные, собранные за собственными компьютерами пользователей).

## **2.2 Фильтрация собранных данных**

В ходе анализа имеющихся наборов данных, характеризующих динамику работы пользователей с клавиатурой компьютера, а также по результатам проведенного обзора существующих решений, был выявлен ряд проблем, возникающих на стадии сбора данных при использовании различных операционных систем, браузеров, локального программного обеспечения и различного оборудования. В частности, было установлено, что использование клавиатурных тренажеров (в частности, клавиатурного тренажера Stamina), негативно влияет на качество распознавания, поскольку заставляет работать пользователя в нехарактерном ему темпе клавиатурного ввода [9]. Поэтому, все события, зарегистрированные в рамках данного процесса Stamina и аналогичных процессов, должны удаляться из дальнейшего рассмотрения.

Также было установлено, что в собранных данных могут присутствовать непарные события нажатия или отжатия клавиш (для одной клавиши может присутствовать событие нажатия, но отсутствовать парное ему событие отжатия и наоборот). Данные события являются шумовыми и также могут негативно влиять на итоговый результат распознавания. Поэтому все непарные события для каждой клавиши также необходимо удалять из дальнейшего рассмотрения.

Отдельно стоит рассмотреть ситуацию длительного удерживания клавиш клавиатуры. В такой ситуации система последовательно генерирует события нажатия клавиши, при этом устанавливая специальный флаг события о повторении нажатия. Время удерживания клавиши, после которого фиксируется событие повторения, зависит от настроек операционной системы и обычно варьируется от 250 до 1000 миллисекунд, при этом частота генерации события повторения обычно варьируется от 2.5 повторений в секунду до 30. Такую цепочку событий стоит исключать из рассмотрения, так как она не дает никакой информации об особенностях работы пользователя с клавиатурой, а может лишь дать данные о конкретных настройках операционной системы. Удаление данных цепочек также позволяет решить проблему фиксирования в системе дубликатов

пришедших от клавиатуры событий, которая может наблюдаться с редкой периодичностью в силу ряда особенностей некоторых операционных систем [37, 38].

### **2.3 Разделение потока собираемых данных на временные окна**

Следующим этапом решения поставленной задачи является разделение потока поступающих от клавиатуры событий на временные окна (на последовательные фрагменты следующих друг за другом событий) и расчет характеристических признаков для каждого временного окна.

Одной из основных проблем, возникающих на этапе разбиения потока собираемых данных на временные окна [11, 37, 38, 45 – 47, 62], является выбор оптимального размера временного окна. Поскольку окна большого размера полноценно характеризуют динамику работу пользователя с клавиатурой компьютера в течение длительного промежутка времени, они являются более информативными. Однако, чем больше размер временного окна, тем меньше временных окон мы можем получить из ограниченного потока событий. С целью увеличения количества используемых для анализа временных окон необходимо использовать окна с перекрытиями (см. Рисунок 3). Важными параметрами, сильно влияющими на итоговый результат классификации, являются размер временного окна, а также процент перекрытия между временными окнами. Отметим, что при использовании большого процента перекрытия окон мы получим большое количество практически одинаковых векторов признаков, что повысит сложность, но не позволит повысить информативность строящейся модели. С другой стороны, при использовании низкого процента перекрытия мы можем потерять большую часть полезной информации.

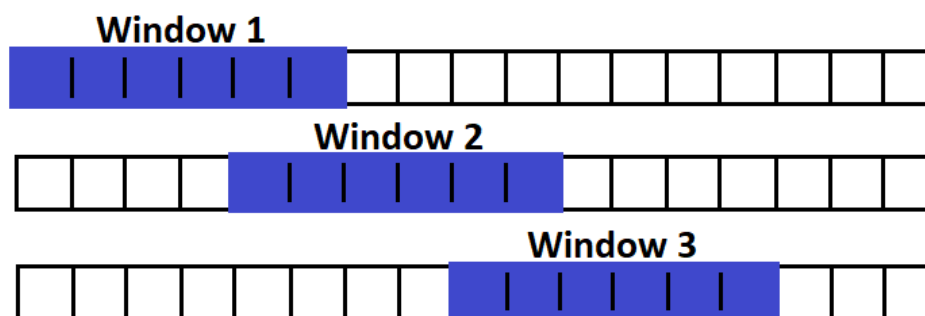


Рисунок 3 — Временные окна с перекрытиями

Поскольку существуют методики разделения потока событий на временные окна как по количеству событий в окне, так и по длительности работы пользователя в рамках данного временного окна, был проведен анализ активности пользователей в рамках

предоставленных тестовых наборов данных. Разбиение по количеству событий в окне предпочтительнее использовать при умеренной работе пользователя с клавиатурой, разбиение по длительности работы пользователя в рамках временного окна эффективнее использовать при активном темпе работы пользователя с клавиатурой. Проведенный анализ показал, что в среднем пользователи обладают умеренным темпом клавиатурного ввода, в связи с чем, эффективнее разделять поток собранных данных на временные окна, основываясь на количестве событий во временном окне. Тем самым гарантируется, что каждое временное окно будет содержать достаточную информацию о клавиатурном почерке пользователя.

Также было проведено исследование возможности использования принудительного разбиения потока клавиатурных событий на временные окна при возникновении длительных пауз в работе пользователя за клавиатурой, а также при смене пользователем активного процесса или просматриваемой web-страницы. Разбиение потока событий на временные окна при возникновении паузы в клавиатурном вводе пользователя (и отнесение событий, пришедших в систему до и после паузы, к разным временным окнам) позволило значительно улучшить качество аутентификации. Обработка же смены деятельности при переходе между приложениями или web-сайтами (отнесение к разным временным окнам событий, пришедших в систему до и после перехода между любыми двумя приложениями или web-сайтами), как показали эксперименты, не даёт улучшения точности распознавания. Это может быть связано с тем, что во время перехода между приложениями или web-страницами пользователь совершает характерные действия, позволяющие классификатору фиксировать определённую специфику его поведения.

Было получено, что эффективнее разделять поток собранных данных на окна размером в 500 событий для каждого окна, и осуществлять принудительное разбиение на окна при возникновении длительных пауз в деятельности пользователя. При этом, если при разбиении получаются окна меньшего размера, то их можно оставлять для дальнейшего рассмотрения при условии, что они содержат не менее 300 событий. Если после принудительного разбиения на временные окна возникли окна меньшего размера, их следует убирать из дальнейшего анализа с целью улучшения качества распознавания.

## ***2.4 Расчет характеристических признаков для каждого временного окна***

На основании вышеприведенного аналитического обзора, а также проведенной серии экспериментальных исследований, было принято решение использовать

комбинированный набор характеристических признаков, характеризующих динамику работы пользователей с клавиатурой компьютера [36 – 38, 40, 45, 49, 62, 69], так как он позволяет учесть больше индивидуальных особенностей пользователя при работе с клавиатурой, анализируя как его работу с отдельными клавишами, так и с их комбинациями (N-граммами, где N – число клавиш, последовательно нажатых в рамках рассматриваемой комбинации). Поскольку в существующих научных работах расчет признаков не для всех, а только для наиболее часто используемых легитимным пользователем клавиш, значительно улучшает качество классификации, а также увеличивает скорость работы программы, сформированный комбинированный набор признаков рассчитывался для 50 наиболее часто используемых пользователем одиночных клавиш клавиатуры и 100 наиболее часто используемых пользователем диграфов. Данные пороговые значения были подобраны экспериментально. Отметим, что при выборе рассматриваемых пороговых значений стоит не забывать, что перед нами ставится задача построения точной, быстро строящейся и быстро применимой модели, поскольку данные алгоритмы должны работать в режиме, близком к режиму реального времени. N-граммы при  $N > 3$  в предлагаемом подходе не рассматриваются, поскольку расчет характеристических признаков для них является вычислительно трудным и не дает значительного улучшения итогового качества аутентификации.

Таким образом, для каждого пользователя будет использоваться собственное признаковое пространство, что также позволит улучшить качество аутентификации, поскольку позволит учесть больше его индивидуальных особенностей при работе с клавиатурой компьютера. Наиболее часто используемые легитимным пользователем одиночные клавиши и диграфы определяются на обучающей выборке. Отметим, что поскольку мы рассматриваем непосредственно сами клавиши клавиатуры, без использования сведений о текущей языковой раскладке, строящиеся модели клавиатурного почерка пользователей не будут чувствительны к используемому языку ввода, что также повысит качество распознавания.

Дополнительно, был проведен анализ физического расположения и функционального предназначения клавиш на клавиатуре, в результате которого были выделены следующие 17 групп клавиш (разные группы клавиш выделены разными цветами, см. Рисунок 4), характеристики работы с каждой из которых также вошли в результирующий набор признаков:

- Esc, F1, F2, F3, F4, F5, F6;
- F7, F8, F9, F10, F11, F12;
- Клавиша Windows;
- Стрелки;

- Клавиша «`~», A, C, D, E, F, G, Q, R, S, T, V, W, X, Y, Z, Клавиша «Пробел»;
- Клавиша «'/"», Клавиша «[ {», Клавиша «\|/»», Клавиша «] }», Клавиша «;:», Клавиша «>,», Клавиша «.>», Клавиша «/?», B, H, I, J, K, L, M, N, O, P, U;
- Shift, Ctrl, Alt;
- Левый Shift, Левый Ctrl, Левая клавиша «Windows», Левая клавиша «Контекстное меню»;
- Правый Shift, Правый Ctrl, Правая клавиша «Windows», Правая клавиша «Контекстное меню»;
- Tab, Caps Lock;
- Backspace, Enter;
- Scroll Lock, Select, Print, Execute, Print Screen, Insert, Delete, Help, Page Up, Page Down, Home, End, Pause;
- 1, 2, 3, 4, 5, 6;
- 7, 8, 9, 0, Клавиша «-», Клавиша «+»;
- Все клавиши Num Pad;
- Клавиша Cancel (остановка процесса);
- Медиа-клавиши (сон, вперед-назад, браузер, звук, управление музыкальными треками, почта и т.д.).



Рисунок 4 — Выделенные группы клавиш клавиатуры

Помимо данных характеристик, для каждого пользователя также рассчитывается частота его клавиатурного ввода в рамках каждого временного окна.

Подробное описание вычисляемых для каждого временного окна характеристических признаков, описывающих взаимодействие пользователя с клавиатурой компьютера [12], приведено в Таблице 2.

Таблица 2 — Описание набора вычисляемых характеристических признаков

Номер признака	Описание
1	<p><i>Название признака</i></p> <p><b>Среднее время удержания клавиши во временном окне</b></p> <p><i>Алгоритм вычисления признака</i></p>

Номер признака	Описание
	<p>Сначала для всех клавиш вычисляем все продолжительности их удерживаний в рамках данного временного окна:</p> $t_i^{up} - t_i^{down},$ <p>где <math>i</math> – порядковый номер клавиши в рассматриваемом временном окне.</p> <p>Далее усредняем данные значения по каждой клавише.</p> <p><i>Сущность признака и область его значений</i></p> <p>Среднее время удержания рассматриваемой клавиши во временном окне.</p> <p>Данный признак вычисляется для одиночных клавиш.</p> <p>Положительные вещественные значения.</p> <p><i>Формат</i></p> <p>Миллисекунды</p>
2	<p><i>Название признака</i></p> <p><b>Среднее время между отпусканием первой и отпусканием второй клавиши во временном окне</b></p> <p><i>Алгоритм вычисления признака</i></p> <p>Сначала для всех диграфов вычисляем все промежутки времени между отпусканием первой и отпусканием второй клавиши в рамках данного временного окна:</p> $t_{i+1}^{up} - t_i^{up},$ <p>где <math>i</math> – порядковый номер клавиши (первой клавиши диграфа) в рассматриваемом временном окне.</p> <p>Далее усредняем данные значения по каждому диграфу.</p> <p><i>Сущность признака и область его значений</i></p> <p>Среднее время между отпусканием первой и отпусканием второй рассматриваемых клавиш во временном окне.</p> <p>Данный признак вычисляется для диграфов.</p> <p>Положительные вещественные значения.</p> <p><i>Формат</i></p> <p>Миллисекунды</p>
3	<p><i>Название признака</i></p> <p><b>Среднее время между нажатием первой и отпусканием второй клавиши во временном окне</b></p>



Номер признака	Описание
	<p><i>Алгоритм вычисления признака</i></p> <p>Сначала для всех диграфов вычисляем все промежутки времени между нажатием первой и отпусканием второй клавиши в рамках данного временного окна:</p> $t_{i+1}^{up} - t_i^{down},$ <p>где <math>i</math> – порядковый номер клавиши (первой клавиши диграфа) в рассматриваемом временном окне.</p> <p>Далее усредняем данные значения по каждому диграфу.</p> <p><i>Сущность признака и область его значений</i></p> <p>Среднее время между нажатием первой и отпусканием второй рассматриваемых клавиш во временном окне.</p> <p>Данный признак вычисляется для диграфов.</p> <p>Положительные вещественные значения.</p> <p><i>Формат</i></p> <p>Миллисекунды</p>
4	<p><i>Название признака</i></p> <p><b>Среднее время удержания группы клавиш во временном окне</b></p> <p><i>Алгоритм вычисления признака</i></p> <p>Сначала для всех клавиш, принадлежащих рассматриваемой группе, вычисляем все продолжительности их удерживаний в рамках данного временного окна:</p> $t_i^{up} - t_i^{down},$ <p>где <math>i</math> – порядковый номер клавиши в рассматриваемом временном окне.</p> <p>Далее усредняем данные значения по каждой группе.</p> <p><i>Сущность признака и область его значений</i></p> <p>Среднее время удержания клавиш, принадлежащих рассматриваемой группе, во временном окне.</p> <p>Данный признак вычисляется для групп клавиш.</p> <p>Положительные вещественные значения.</p> <p><i>Формат</i></p> <p>Миллисекунды</p>
5	<p><i>Название признака</i></p>

Номер признака	Описание
	<p><b>Частота набора текста пользователем во временном окне</b></p> <p><i>Алгоритм вычисления признака</i></p> <p>Рассчитываем величину</p> $\frac{t}{N},$ <p>где <math>t</math> – продолжительность данного временного окна; <math>N</math> – количество нажатых в рамках данного временного окна клавиш.</p> <p><i>Сущность признака и область его значений</i></p> <p>Частота набора текста для данного временного окна.</p> <p>Данный признак вычисляется для всех клавиш (всего вводимого текста в рамках данного временного окна).</p> <p>Положительные вещественные значения.</p> <p><i>Формат</i></p> <p>Секунды/нажатие</p>

Отметим, что по результатам проведенного экспериментального исследования в результирующий набор признаков для диграфов вошли усредненные значения признаков  $t_{i+1}^{up} - t_i^{up}$  и  $t_{i+1}^{up} - t_i^{down}$ . Величины  $t_{i+1}^{down} - t_i^{up}$  и  $t_{i+1}^{down} - t_i^{down}$  были убраны из дальнейшего рассмотрения, поскольку их включение в итоговый вектор признаков не дает улучшения качества строящейся модели.

## **2.5 Выделение стабильных признаков, характеризующих динамику работы пользователя с клавиатурой компьютера**

В результате обзора существующих подходов к решению задачи динамической аутентификации пользователя на основе анализа его клавиатурного почерка было выявлено, что размерность результирующего признакового пространства в данной задаче получается достаточно высокой. Это связано с тем, что помимо расчета признаков для одиночных клавиш, мы также рассматриваем характеристические признаки для N-грамм, количество которых экспоненциально зависит от количества клавиш, расположенных на клавиатуре. Помимо того, что большое количество признаков увеличивает сложность модели и время ее построения, часть признаков могут являться шумовыми и негативно влиять на итоговый результат распознавания. Для решения данной проблемы было

принято решение отбирать 50 наиболее часто используемых пользователем одиночных клавиш клавиатуры и 100 наиболее часто используемых пользователем диграфов и дальнейший расчет признаков осуществлять только для них. Данный подход позволил улучшить качество построенных моделей.

Однако, в результате проведенной серии экспериментов было выявлено, что даже после отбора наиболее часто используемых одиночных клавиш и диграфов, качество распознавания пользователей с течением времени продолжает резко ухудшаться (в том числе и при смене используемой аппаратуры) – необходимо научиться отбирать наиболее стабильные по времени признаковые характеристики. Для решения данной проблемы в данной диссертационной работе предлагается рассматривать распределение характеристических признаков, описывающих клавиатурный почерк пользователя (для отобранных одиночных клавиш и диграфов) в рамках каждого временного окна, анализировать характер изменения данного распределения с течением времени и оставлять для дальнейшего рассмотрения только те признаки, распределение которых постоянно на протяжении всего времени работы рассматриваемого пользователя за компьютером (назовем данные признаки *стабильными*) [12]. Данную методику можно применять для характеристик, многократно проявляющихся в рамках каждого временного окна (так называемых *составных* признаков). В данной задаче, к таким характеристикам относятся:

- Время удержания клавиши во временном окне;
- Время между отпусканием первой и отпусканием второй клавиши во временном окне;
- Время между нажатием первой и отпусканием второй клавиши во временном окне;
- Время удержания группы клавиш во временном окне.

Для данных характеристик ранее было предложено вычислять среднее значение по каждой клавише, диграфу или группе клавиш соответственно в рамках каждого временного окна и вносить полученное значение в результирующий вектор признаков. Предварительный анализ распределения каждой из этих характеристик и удаление из дальнейшего рассмотрения тех характеристик, характер распределения которых с течением времени значительно изменяется, позволит не только улучшить качество аутентификации, но с высокой вероятностью также позволит и решить проблему падения качества распознавания пользователей во времени, в том числе и при смене используемого оборудования (поскольку если без смены оборудования характер

распределения признака резко меняется с течением времени, то со сменой оборудования он будет еще более непостоянным, что резко снизит качество аутентификации). Таким образом, в результирующий вектор признаков будут входить средние значения отобранных стабильных составных признаков для каждого временного окна (характер распределения которых будет сохраняться с течением времени).

Определять характер изменения распределения рассматриваемых характеристик с течением времени предлагается с использованием *критерия Колмогорова-Смирнова*. Для каждой характеристики на основе данного критерия мы сможем определить *степень ее стабильности* (постоянства ее распределения). Для отбора признаков можно отсортировать признаки по уровню их стабильности и отобрать некоторое их число, либо же задать порог для уровня стабильности, при пересечении которого признак будет добавлен к дальнейшему рассмотрению. При этом, количество отбираемых для дальнейшего рассмотрения стабильных признаков (или порог для уровня стабильности во втором случае соответственно) будет являться параметром системы.

В данной задаче единственным несоставным признаком (проявляющимся однократно в каждом временном окне) является частота набора текста пользователем в рамках каждого временного окна. Как показали проведенные эксперименты, данный признак является высоко информативным, и его использование повышает качество аутентификации пользователя, в том числе и при смене используемого оборудования, поэтому его не стоит убирать из дальнейшего рассмотрения.

### **2.5.1 Формальная постановка задачи расчета уровня стабильности произвольного составного характеристического признака**

Рассмотрим формальную постановку задачи расчета уровня стабильности произвольного составного характеристического признака.

Для каждого пользователя  $U_i$  имеется последовательность построенных для него временных окон событий:

$$B(U_i) = (W_{i1}, W_{i2}, \dots, W_{iM}), \quad (9)$$

в которой каждое временное окно  $W_{ij}$ ,  $j = \overline{1, M}$  содержит последовательность неусредненных характеристических признаков для одиночных клавиш или их комбинаций, рассчитанных на основе поступивших в рамках данного временного окна событий:

$$W_{ij} = (W_{ij0}, W_{ij1}, \dots, W_{ijs}), \quad (10)$$

где  $W_{ijk}$ ,  $k = \overline{1, S}$  –  $k$ -ая рассчитанная для одиночной клавиши или диграфа характеристика;  $S$  – размер временного окна  $W_{ij}$ ;  $M$  – количество временных окон, построенных для рассматриваемого пользователя  $U_i$ .

Необходимо для каждого составного признака оценить характер постоянства его распределения с течением времени. Для этого необходимо сравнивать распределения данного признака в рамках каждого временного окна  $W_{ij}$  и в рамках всего набора временных окон целиком  $B(U_i)$ .

Для решения поставленной задачи можно воспользоваться статистическими критериями, опирающимися на проверку гипотезы  $H_0$  о том, что распределение в двух рассматриваемых выборках совпадает:

$$H_0: P_1 = P_2 \quad (11)$$

против альтернативной гипотезы о том, что распределение в данных выборках различно:

$$H_1: P_1 \neq P_2. \quad (12)$$

При выборе используемого статистического критерия необходимо иметь в виду, что количество наблюдений в рассматриваемых выборках различно.

Отметим, что при проверке статистических гипотез рассчитывается уровень их значимости (p-value), значения которого лежат в диапазоне  $[0,1]$ . После проверки всех статистических гипотез для рассматриваемого признака (по каждому временному окну), мы сможем объединить полученные значения p-value и на выходе получим конкретную величину – уровень стабильности рассматриваемого признака, принимающую значения из фиксированного диапазона. Дальнейший отбор признаков будет осуществляться на основе данной величины: в итоговый набор признаков войдут усредненные значения составных признаков с наибольшим уровнем стабильности. В качестве используемого статистического критерия предлагается использовать критерий Колмогорова-Смирнова.

Рассмотрим процедуру вычисления уровня стабильности составных характеристических признаков с использованием данного критерия более подробно.

### **2.5.2 Вычисление уровня стабильности составного характеристического признака с использованием критерия Колмогорова-Смирнова**

Одним из наиболее часто используемых на практике статистических критериев является критерий Колмогорова-Смирнова [13, 88, 89]. Данный критерий также способен оценить уровень стабильности произвольного составного признака. В этом критерии распределение признака описывается множеством принимаемых им значений.

Суть критерия Колмогорова-Смирнова заключается в проверке гипотезы о том, что рассматриваемая случайная величина подчиняется некоторому теоретическому закону распределения. Здесь, в качестве случайной величины рассматриваются значения составного признака перед их усреднением в конкретном временном окне, а в качестве теоретического распределения – все значения признака, проявившиеся на некоторой выборке (возможно, на всем обучающем наборе).

Пусть  $F_{1,n}$  – функция распределения рассматриваемого составного признака в данном временном окне,  $n$  – количество раз, которое признак проявился в данном временном окне,  $F_{2,m}$  – теоретическая функция распределения признака,  $m$  – количество раз, которое признак проявился в выборке, по которой рассматривается теоретическое распределение.

Основной задачей является нахождение такого числа  $\lambda$ , что

$$\sqrt{\frac{nm}{n+m}} D_{n,m} \geq \lambda, \quad (13)$$

$$\text{где } D_{n,m} = \sup_x |F_{1,n}(x) - F_{2,m}(x)|. \quad (14)$$

Далее, исходя из таблицы квантилей распределения Колмогорова, ищется соответствующее найденному  $\lambda$  значение p-value – вероятность ошибки при отклонении от нулевой гипотезы. Данная процедура выполняется для каждого рассматриваемого составного признака в каждом имеющемся временном окне (векторе признаков).

Таким образом, для конкретного признака в каждом временном окне будет получено значение p-value, показывающее, насколько значения признака в данном окне соответствуют значениям данного признака, полученным на выборке, из которой берется теоретическое распределение (возможно, на всем обучающем наборе). Для объединения p-value, полученных для всех имеющихся окон, предлагается использовать метод Фишера. Для этого, необходимо осуществить расчет статистики:

$$\chi^2_{2k} \approx -2 \cdot \sum_{i=1}^k \ln(p_i), \quad (15)$$

где  $k$  – число комбинируемых значений,  $p_i$  –  $i$ -ое значение p-value. Далее, по таблице распределения Хи-квадрат находится значение p-value, соответствующее данной статистике с числом степеней свободы  $2k$ .

Полученные значения p-value для множества составных признаков могут использоваться для их ранжирования и селекции по степени стабильности. Одним из возможных вариантов является фиксирование порога уровня стабильности (задание уровня значимости), при превышении которого признак будет считаться стабильным и будет использован при построении модели. Также, возможно отбирать некоторое предопределенное количество признаков с наибольшим значением p-value.

## 2.6 Дискретизация признаков по квантилям

По результатам предварительной серии экспериментов было выявлено, что распределение признаков, характеризующих клавиатурный почерк пользователей, является мультимодальным. Для решения данной проблемы в данной диссертационной работе предлагается использовать дискретизацию признаков по квантилям, ранее не применявшуюся в задаче динамической аутентификации пользователей на основе анализа их работы с клавиатурой компьютера.

Дискретизация признака на основе значений квантилей осуществляет преобразование непрерывных значений признака в дискретные [12]. Данное преобразование позволяет избавиться от шумов в данных и в целом сократить их объем. Для его осуществления множество значений признака на обучающем наборе разбивается на непересекающиеся интервалы по квантилям порядка  $\frac{1}{k}, \frac{2}{k}, \dots, \frac{k-1}{k}$ , где  $k$  – число интервалов разбиения. При этом рассматриваются только уникальные значения полученных квантилей. Далее, осуществляется отображение значения признака в номер соответствующего ему интервала. Оптимальное количество квантилей подбирается экспериментально. Помимо борьбы с шумами в данных, дискретизация также помогает бороться со случайными действиями пользователя, тем самым повышая качество работы алгоритмов классификации. Пример дискретизации признаков по квантилям представлен на Рисунке 5.

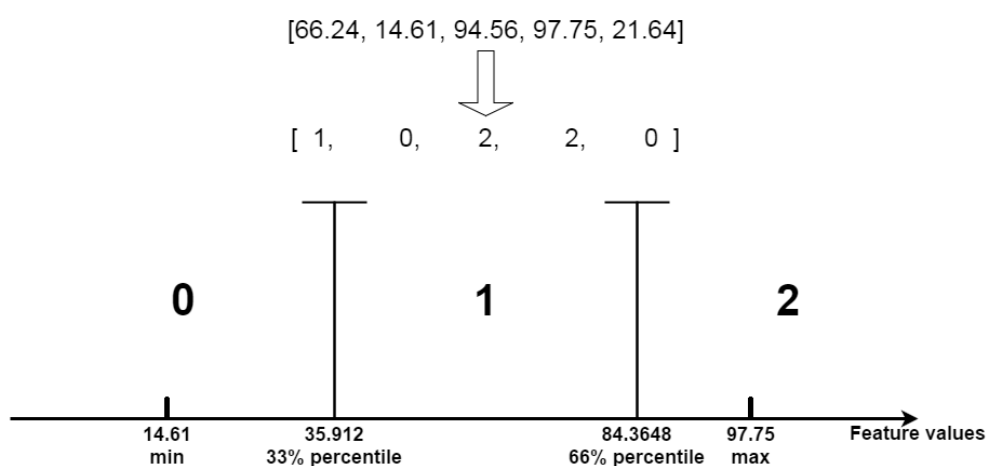
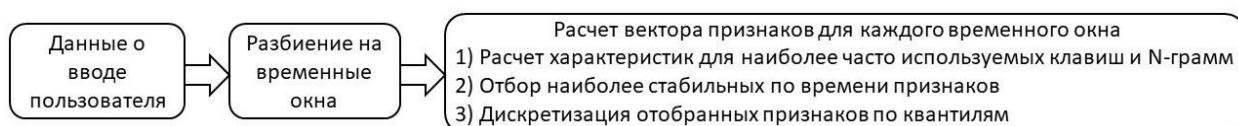


Рисунок 5 — Пример дискретизации признаков по квантилям

Общая схема предложенного подхода к обработке данных представлена на Рисунке 6.



## 2.7 Экспериментальное исследование

С целью проверки предложенных методов и подходов к предобработке данных, характеризующих динамику работы пользователей с клавиатурой компьютера, была проведена серия экспериментальных исследований, состоящая из следующих этапов:

- 1) Исследование параметров построения временных окон;
- 2) Исследование признакового пространства модели представления и подбор соответствующих параметров;
- 3) Исследование методов обработки признаков и подбор соответствующих параметров;
- 4) Исследование устойчивости работы предложенных алгоритмов к смене используемого оборудования.

Отметим, что экспериментальные исследования проходили как на данных клавиатурного почерка пользователей, собранных локально в фоновом режиме при обычной работе пользователей за компьютером, так и на данных, собранных web-сборщиком при работе пользователей в web-браузере. Также, с целью проверки устойчивости работы предложенных алгоритмов были проведены дополнительные эксперименты со сменой используемого оборудования. Эксперименты проводились с использованием классификатора SVC (использовалось ядро RBF, ширина ядра подбиралась автоматически, процент возможных исключений составлял 10%), как наиболее перспективного по результатам проведенного обзора. Данный метод показывает наилучшее качество распознавания и основан на переходе в пространство характеристик высокой размерности (RKHS), целью которого является эффективное использование более простых геометрических структур для описания зависимостей во входных данных. Являясь kernel-методом [15], SVC позволяет бороться с проблемой проклятия размерности. Для оценки качества работы рассматриваемых алгоритмов использовалось усредненное по всем пользователям значение площади под ROC-кривой (ROC AUC), являющееся агрегированной характеристикой качества классификации, не зависящей от соотношения цен ошибок.



### 2.7.1 Исследование параметров построения временных окон

Основная задача первого этапа экспериментов заключалась в определении степени влияния на точность динамической аутентификации пользователей по динамике их работы с клавиатурой компьютера таких параметров системы, как размер временного окна, процент перекрытия временных окон, количество квантилей при дискретизации и ограничение на минимальный размер обучающего набора. На основе анализа результатов соответствующих экспериментов можно составить представление о том, при каких значениях данных параметров достигается наилучшая точность классификации. Далее, за счет фиксации данных оптимальных значений параметров можно получить возможность сократить количество экспериментов на последующих этапах исследований. Ограничение на минимальный размер обучающего набора вводится в виду того, что в имеющихся наборах данных присутствуют данные пользователей, уровень активности работы с клавиатурой которых не позволил рассчитать по ним векторы признаков в количестве, необходимом для построения высокоточных поведенческих моделей.

При вычислениях на данном этапе был зафиксирован следующий набор характеристических признаков, описывающих клавиатурный почерк пользователей:

- Среднее время удержания клавиши во временном окне;
- Характеристики диграфов (среднее время между отпусканием первой и отпусканием второй клавиши во временном окне, среднее время между нажатием первой и отпусканием второй клавиши во временном окне);
- Среднее время удержания группы клавиш во временном окне;
- Частота набора текста пользователем во временном окне.

Для каждой одиночной клавиши, а также каждого диграфа был зафиксирован порог встречаемости, относительно которого рассматриваемые характеристики либо включались в итоговый вектор признаков, либо нет. Для одиночных нажатий были зафиксированы 80 наиболее часто используемых пользователем клавиш, для диграфов – 150 наиболее часто используемых комбинаций клавиш. Величина паузы, используемая при разбиении на временные окна, составляла 80 секунд. Данные признаки (и соответствующие им параметры) были выбраны на основе промежуточных экспериментов, где они показали наиболее высокие результаты.

Результаты экспериментов представлены в Таблицах 3 и 4.

Таблица 3 — Результаты экспериментов на локальных данных (Исследование параметров построения временных окон)

<b><i>Набор данных 2</i></b>			<b>10000 символов</b>			<b>15000 символов</b>		
Размер окна	Процент перекрытия окон	Количество квантилей при дискретизации	Ограничение на минимальный размер обучающего набора	Количество пользователей после ограничения на обучающий набор	ROC AUC	Ограничение на минимальный размер обучающего набора	Количество пользователей после ограничения на обучающий набор	ROC AUC
200-300	0,01	5	67+	16/20	0,7796	100+	9/20	0,8363
		7			0,7882			0,8533
		10			0,7934			0,8657
	0,25	5		18/20	0,7794		14/20	0,8034
		7			0,7827			0,8134
		10			0,7790			0,8175
	0,5	7		19/20	0,7800		18/20	0,7774
300-500	0,01	3	40+	16/20	0,8079	60+	9/20	0,8726
		5			0,8186			0,8890
		7			<b>0,8206</b>			<b>0,8924</b>
	0,25	3		18/20	0,8147		13/20	0,8238
		5			0,8116			0,8389
		7			0,8036			0,8384
		10			0,7961			0,8375
	0,5	3		19/20	0,8087		17/20	0,8137
		5			0,8110			0,8174
		7			0,8074			0,8139
		10			0,7971			0,8046
		50			0,4994			0,4996

Таблица 4 — Результаты экспериментов на web-данных (Исследование параметров построения временных окон)

<b><i>Набор данных 3</i></b>			<b>10000 символов</b>			<b>15000 символов</b>		
Размер окна	Процент перекрытия окон	Количество квантилей при дискретизации	Ограничение на минимальный размер обучающего набора	Количество пользователей после ограничения на обучающий набор	ROC AUC	Ограничение на минимальный размер обучающего набора	Количество пользователей после ограничения на обучающий набор	ROC AUC
200-300	0,01	3	67+	14/20	0,8128	100+	7/20	0,8738
		5			0,8309			0,9057
		7			0,8341			0,9090
		10			0,8324			0,9079
	0,25	5		16/20	0,8213		11/20	0,8618
		7			0,8218			0,8567
		10			0,8224			0,8593
	0,5	10		18/20	0,8059		16/20	0,8282

<b><i>Набор данных 3</i></b>			<b>10000 символов</b>			<b>15000 символов</b>		
300-500	0,01	3	40+	14/20	0,8604	60+	7/20	0,9288
		5			<b>0,8627</b>			<b>0,9338</b>
		7			0,8576			0,9301
		10			0,8454			0,9218
	0,25	3		16/20	0,8625		11/20	0,8873
		5			0,8626			0,8844
		7			0,8578			0,8789
		10			0,8485			0,8665
	0,5	3		18/20	0,8448		16/20	0,8670
		5			0,8426			0,8628
		7			0,8365			0,8605
		10			0,8290			0,8498

На основании полученных результатов можно сделать следующие выводы:

- Как для данных, собранных локально, так и для данных, собранных при помощи средств web-сбора, качество динамической аутентификации пользователей на основе анализа их клавиатурного почерка является достаточно высоким;
- На точность классификации пользователей в большей части влияют следующие параметры:
  - Размер временного окна (чем больше размер временного окна, тем выше получаемый показатель ROC AUC);
  - Ограничение на минимальный размер обучающего набора (чем больше ограничение, тем выше точность, но меньше пользователей, рассматриваемых после ограничения);
  - Количество квантилей при дискретизации (наилучшие результаты достигаются при количестве квантилей в диапазоне от 3 до 10);
- Увеличение процента перекрытия окон событий в подавляющем большинстве экспериментов отрицательно сказывается на итоговой точности распознавания.

Таким образом, наилучшие результаты, полученные после первого этапа экспериментов, следующие:

- **Локальные данные:**
  - **Размер окна: (300 – 500) событий;**
  - **Процент перекрытия: 0.01;**
  - **Количество квантилей для дискретизации: 7;**

- Итоговое качество аутентификации: более 40 обучающих векторов – 0.8206 ROC AUC, более 60 обучающих векторов – 0.8924 ROC AUC;
- **Web-данные:**
  - **Размер окна: (300 – 500) событий;**
  - **Процент перекрытия: 0.01;**
  - **Количество квантилей для дискретизации: 5;**
  - Итоговое качество аутентификации: более 40 обучающих векторов – 0.8627 ROC AUC, более 60 обучающих векторов – 0.9338 ROC AUC.

### **2.7.2 Исследование признакового пространства модели представления и подбор соответствующих параметров**

Основная задача второго этапа экспериментов заключалась в определении степени влияния на точность динамической аутентификации пользователей по динамике их работы с клавиатурой компьютера следующих параметров построения модели представления: величины паузы при разбиении на временные окна, количества рассматриваемых наиболее часто используемых пользователем одиночных клавиш и диграфов, а также возможности включения в итоговый вектор признаков такого группового признака, как среднее время удержания группы клавиш во временном окне.

В экспериментах рассматривались следующие значения исследуемых параметров:

- Величина паузы при разбиении на временные окна: 40, 60, 80 секунд;
- Количество используемых наиболее часто встречаемых одиночных клавиш: 50, 75, 100;
- Количество используемых наиболее часто встречаемых диграфов: 0, 100, 150, 200;
- Использование или неиспользование группового признака (среднего времени удержания группы клавиш во временном окне).

Отметим, что на данном этапе исследований использовались оптимальные значения параметров построения временных окон, полученные на первом этапе экспериментов.

Ввиду того, что при ограничении в 15000 символов в рассмотрении остаются только 8 из 20 пользователей, в дальнейшем будем рассматривать только ограничение в 10000 введенных пользователем символов.

Результаты экспериментов представлены в Таблицах 5 и 6.

Таблица 5 — Результаты экспериментов на локальных данных (Исследование признакового пространства модели представления и подбор соответствующих параметров)

<b><u>Набор данных 2</u></b>				
Величина паузы при разбиении на окна	Количество используемых наиболее часто встречаемых одиночных клавиш	Количество используемых наиболее часто встречаемых диграфов	Использование групповых признаков	ROC AUC
40	50	100	Да	<b>0,8494</b>
80	50			0,8483
40	75			0,8482
60	50			0,8481
60	75			0,8479

Таблица 6 — Результаты экспериментов на web-данных (Исследование признакового пространства модели представления и подбор соответствующих параметров)

<b><u>Набор данных 3</u></b>				
Величина паузы при разбиении на окна	Количество используемых наиболее часто встречаемых одиночных клавиш	Количество используемых наиболее часто встречаемых диграфов	Использование групповых признаков	ROC AUC
80	75	100	Да	<b>0,8880</b>
	50			0,8879
	100			0,8875
60	75			0,8869
	50			0,8867

На основании полученных результатов можно сделать следующие выводы:

- Использование группового признака (среднего времени удержания группы клавиш во временном окне) положительно влияет на итоговое качество классификации;
- Для достижения высокой точности аутентификации достаточно рассматривать небольшое количество наиболее часто используемых пользователем одиночных клавиш – 50 (в случае локальных данных) или 75 (в случае web-данных), а также 100 наиболее часто используемых пользователем диграфов;
- В среднем прирост значения ROC AUC при наилучших параметрах на данном этапе исследований составляет 2–3% по сравнению с результатами, полученными на первом этапе.

Таким образом, наилучшие результаты после второго этапа исследований следующие:

- **Локальные данные:**
  - **Размер окна: (300 – 500) событий;**
  - **Процент перекрытия: 0.01;**
  - **Количество квантилей для дискретизации: 7;**
  - **Пауза при разбиении на временные окна: 40 секунд;**
  - **Количество рассматриваемых наиболее часто используемых одиночных клавиш: 50;**
  - **Количество рассматриваемых наиболее часто используемых диграфов: 100;**
  - **Использование группового признака (среднего времени удержания группы клавиш во временном окне);**
  - **Итоговое качество аутентификации (от 40 обучающих векторов): 0.8494 ROC AUC.**
- **Web-данные:**
  - **Размер окна: (300 – 500) событий;**
  - **Процент перекрытия: 0.01;**
  - **Количество квантилей для дискретизации: 5;**
  - **Пауза при разбиении на временные окна: 80 секунд;**
  - **Количество рассматриваемых наиболее часто используемых одиночных клавиш: 75;**
  - **Количество рассматриваемых наиболее часто используемых диграфов: 100;**
  - **Использование группового признака (среднего времени удержания группы клавиш во временном окне);**
  - **Итоговое качество аутентификации (от 40 обучающих векторов): 0.8880 ROC AUC.**

### **2.7.3 Исследование методов обработки признаков и подбор соответствующих параметров**

Основная задача третьего этапа экспериментов заключалась в определении комбинации методов обработки признаков, дающей наилучшую точность динамической аутентификации пользователей по динамике их работы с клавиатурой компьютера. На данном этапе исследований проводилось сравнение работы методов стандартизации и

дискретизации признаков по квантилям для обработки полученных признаков, а также сравнение работы метода главных компонент и метода отбора признаков по уровню их стабильности на основе критерия Колмогорова-Смирнова при сокращении размерности признакового пространства, характеризующего клавиатурный почерк пользователя.

В экспериментах рассматривались следующие значения исследуемых параметров:

- Количество отбираемых наиболее стабильных признаков: 100, 150, 200;
- Пороговое значение уровня стабильности признаков: 0.1, 0.3, 0.5;
- Количество отбираемых главных компонент: 20, 50;
- Количество квантилей при дискретизации: 3, 5, 7, 10.

Отметим, что на данном этапе исследований использовались оптимальные значения параметров построения временных окон и модели представления, полученные на первом и втором этапах экспериментов соответственно.

### Эксперименты на локальных данных (Набор данных 2)

При классификации пользователей по динамике их работы с компьютерной клавиатурой при использовании алгоритма **стандартизации признаков** был получен средний по пользователям показатель ROC AUC, равный 0.7872.

Результаты классификации пользователей по динамике работы с компьютерной клавиатурой при использовании **дискретизации признаков по квантилям** представлены в Таблице 7.

Таблица 7 — Результаты классификации пользователей по динамике работы с компьютерной клавиатурой при использовании дискретизации признаков по квантилям

Количество квантилей для дискретизации	ROC AUC
7	<b>0.8494</b>
10	0.8183
5	0.8060
3	0.7496

Наилучшие результаты классификации пользователей по динамике работы с компьютерной клавиатурой с **отбором признаков на основе метода главных компонент и стандартизацией признаков** представлены в Таблице 8.

Таблица 8 — Наилучшие результаты классификации пользователей по динамике работы с компьютерной клавиатурой на основе метода главных компонент и стандартизацией признаков

Количество отбираемых главных компонент	ROC AUC
---	---------

Количество отбираемых главных компонент	ROC AUC
50	<b>0.7631</b>
20	0.7422

Наилучшие результаты классификации пользователей по динамике работы с компьютерной клавиатурой с **отбором признаков на основе метода главных компонент и дискретизацией признаков по квантилям** представлены в Таблице 9.

Таблица 9 — Наилучшие результаты классификации пользователей по динамике работы с компьютерной клавиатурой с отбором признаков на основе метода главных компонент и дискретизацией признаков по квантилям

Количество квантилей для дискретизации	Количество отбираемых главных компонент	ROC AUC
7	50	<b>0.8241</b>
7	20	0.8033

Наилучшие результаты классификации пользователей по динамике работы с компьютерной клавиатурой с **отбором признаков по уровню их стабильности с использованием критерия Колмогорова-Смирнова и стандартизацией признаков** представлены в Таблице 10.

Таблица 10 — Наилучшие результаты классификации пользователей по динамике работы с компьютерной клавиатурой с отбором признаков по уровню их стабильности с использованием критерия Колмогорова-Смирнова и стандартизацией признаков

Количество отбираемых признаков	Уровень значимости	ROC AUC
—	0.1	<b>0.8105</b>
100	—	0.8024
—	0.3	0.8024
—	0.5	0.7975
150	—	0.7963

Наилучшие результаты классификации пользователей по динамике работы с компьютерной клавиатурой с **отбором признаков по уровню их стабильности с использованием критерия Колмогорова-Смирнова и дискретизацией признаков по квантилям** представлены в Таблице 11.

Таблица 11 — Наилучшие результаты классификации пользователей по динамике работы с компьютерной клавиатурой с отбором признаков по уровню их стабильности с использованием критерия Колмогорова-Смирнова и дискретизацией признаков по квантилям

Количество квантилей для дискретизации	Количество отбираемых признаков	Уровень значимости	ROC AUC
7	—	0.5	<b>0.8698</b>
7	150	—	0.8638
7	100	—	0.8625



Количество квантилей для дискретизации	Количество отбираемых признаков	Уровень значимости	ROC AUC
7	200	—	0.8603
5	100	—	0.8574

### Эксперименты на web-данных (Набор данных 3)

При классификации пользователей по динамике их работы с компьютерной клавиатурой при использовании алгоритма **стандартизации признаков** был получен средний по пользователям показатель ROC AUC, равный **0.7213**.

Результаты классификации пользователей по динамике работы с компьютерной клавиатурой при использовании **дискретизации признаков по квантилям** представлены в Таблице 12.

Таблица 12 — Результаты классификации пользователей по динамике работы с компьютерной клавиатурой при использовании дискретизации признаков по квантилям

Количество квантилей для дискретизации	ROC AUC
5	<b>0.8880</b>
7	0.8716
10	0.8703
3	0.8425

Наилучшие результаты классификации пользователей по динамике работы с компьютерной клавиатурой с **отбором признаков на основе метода главных компонент и стандартизацией признаков** представлены в Таблице 13.

Таблица 13 — Наилучшие результаты классификации пользователей по динамике работы с компьютерной клавиатурой на основе метода главных компонент и стандартизацией признаков

Количество отбираемых главных компонент	ROC AUC
50	<b>0.7015</b>
20	0.6845

Наилучшие результаты классификации пользователей по динамике работы с компьютерной клавиатурой с **отбором признаков на основе метода главных компонент и дискретизацией признаков по квантилям** представлены в Таблице 14.

Таблица 14 — Наилучшие результаты классификации пользователей по динамике работы с компьютерной клавиатурой с отбором признаков на основе метода главных компонент и дискретизацией признаков по квантилям

Количество квантилей для дискретизации	Количество отбираемых главных компонент	ROC AUC
5	50	<b>0.8691</b>

Количество квантилей для дискретизации	Количество отбираемых главных компонент	ROC AUC
5	20	0.8214

Наилучшие результаты классификации пользователей по динамике работы с компьютерной клавиатурой с **отбором признаков по уровню их стабильности с использованием критерия Колмогорова-Смирнова и стандартизацией признаков** представлены в Таблице 15.

Таблица 15 — Наилучшие результаты классификации пользователей по динамике работы с компьютерной клавиатурой с отбором признаков по уровню их стабильности с использованием критерия Колмогорова-Смирнова и стандартизацией признаков

Количество отбираемых признаков	Уровень значимости	ROC AUC
—	0.5	<b>0.7425</b>
150	—	0.7398
100	—	0.7245
—	0.3	0.7013
—	0.1	0.6997

Наилучшие результаты классификации пользователей по динамике работы с компьютерной клавиатурой с **отбором признаков по уровню их стабильности с использованием критерия Колмогорова-Смирнова и дискретизацией признаков по квантилям** представлены в Таблице 16.

Таблица 16 — Наилучшие результаты классификации пользователей по динамике работы с компьютерной клавиатурой с отбором признаков по уровню их стабильности с использованием критерия Колмогорова-Смирнова и дискретизацией признаков по квантилям

Количество квантилей для дискретизации	Количество отбираемых признаков	Уровень значимости	ROC AUC
5	150	—	<b>0.9125</b>
5	—	0.5	0.9118
5	100	—	0.9105
5	—	0.3	0.9018
5	—	0.1	0.8977

На основании полученных результатов можно сделать следующие выводы:

- Использование дискретизации признаков по квантилям позволяет существенно улучшить качество аутентификации пользователей по динамике их работы с клавиатурой компьютера по сравнению со стандартизацией признаков (как на локальных, так и на web-данных);
- Качество аутентификации пользователей при использовании отбора признаков по уровню их стабильности на основе критерия Колмогорова-

Смирнова существенно превосходит качество аутентификации при использовании отбора признаков на основе метода главных компонент (как на локальных, так и на web-данных);

- Наилучшие результаты по динамической аутентификации пользователей на основе клавиатурного почерка на локальном наборе данных достигаются при использовании дискретизации признаков по 7 квантилям и отбору признаков, уровень стабильности которых выше 0.5 (ROC AUC = 0.8698);
- Наилучшие результаты по динамической аутентификации пользователей на основе клавиатурного почерка на web-наборе данных достигаются при использовании дискретизации признаков по 5 квантилям и отбору 150 наиболее стабильных признаков (ROC AUC = 0.9125).

#### **2.7.4 Исследование устойчивости работы предложенных алгоритмов к смене используемого оборудования**

Основная задача четвертого этапа экспериментов заключалась в исследовании устойчивости работы комбинации предложенных алгоритмов (отбора признаков по уровню их стабильности на основе критерия Колмогорова-Смирнова с дальнейшей дискретизацией по квантилям наиболее стабильных признаков) к смене используемого оборудования. Под сменой используемого оборудования понимается замена типа клавиатуры компьютера: обучение модели происходит на клавиатуре одного типа, тестирование – на клавиатуре другого типа. В рамках данной работы проводились исследования для двух типов компьютерных клавиатур:

- Мембранная клавиатура с цифровым блоком и плоскими клавишами;
- Мембранная клавиатура с цифровым блоком и цилиндрическими клавишами.

Отметим, что на данном этапе исследований использовались оптимальные значения параметров построения временных окон, модели представления и алгоритмов обработки признаков, полученные на первых трех этапах экспериментов соответственно.

Эксперименты проходили на Тестовом наборе данных №4.

Результаты проведенных экспериментов представлены в Таблице 17.

Таблица 17 — Результаты экспериментов со сменой используемого оборудования

<b><u>Набор данных 4, ROC AUC</u></b>		
	<b>Дискретизация признаков по квантилям</b>	<b>Отбор признаков по уровню стабильности, Дискретизация признаков по квантилям</b>

<b><i>Набор данных 4, ROC AUC</i></b>		
Без смены используемого оборудования	0.9057	0.9108
Со сменой используемого оборудования	0.8361	0.8752

На основании полученных результатов можно сделать следующие выводы:

- Совместное использование отбора признаков по уровню их стабильности с использованием критерия Колмогорова-Смирнова и дискретизации признаков по квантилям позволяет решить проблему падения качества распознавания пользователей при смене используемого оборудования (в случае перехода между мембранной клавиатурой с цифровым блоком и плоскими клавишами и мембранной клавиатурой с цифровым блоком и цилиндрическими клавишами): качество аутентификации падает незначительно (всего на 3.5%) по сравнению с достигнутыми в существующих работах результатами (в них качество распознавания падает порядка на 20%).

## **2.8 Выводы**

В данном разделе проводились исследование и разработка методов обработки данных, характеризующих динамику работы пользователей с клавиатурой компьютера. В результате проведенного исследования были достигнуты следующие результаты:

- Было сформировано стабильное по времени признаковое пространство, использование которого позволяет достичь высокого качества аутентификации пользователей по динамике их работы с компьютерной клавиатурой. Данное признаковое пространство включает в себя следующие характеристики:
  - Среднее время удержания клавиши во временном окне;
  - Среднее время между отпусканием первой и отпусканием второй клавиши во временном окне;
  - Среднее время между нажатием первой и отпусканием второй клавиши во временном окне;
  - Среднее время удержания группы клавиш во временном окне;
  - Частота набора текста пользователем во временном окне.

Отметим, что данное признаковое пространство не зависит от используемого пользователем языка ввода, что позволяет значительно повысить итоговое качество распознавания.

- Был предложен подход к обработке признаков, характеризующих клавиатурный почерк пользователя, путем дискретизации их по квантилям, ранее не используемый для решения данной задачи. Данный подход позволил решить проблему мультимодального распределения характеристических признаков и тем самым повысить качество аутентификации пользователей в среднем на 6% по сравнению с используемой в существующих работах стандартизацией признаков.
- Был предложен подход к сокращению размерности признакового пространства путем отбора наиболее значимых признаков по уровню их стабильности с использованием критерия Колмогорова-Смирнова. Данный метод не только позволил улучшить качество аутентификации пользователей в среднем на 4%, но также и решил проблему падения качества распознавания пользователей во времени, в том числе и при смене используемого оборудования (в случае перехода между мембранной клавиатурой с цифровым блоком и плоскими клавишами и мембранной клавиатурой с цифровым блоком и цилиндрическими клавишами). При использовании данного подхода совместно с дискретизацией признаков по квантилям качество распознавания пользователей при смене используемого оборудования падает в среднем на 3%, что является достаточно хорошим результатом по сравнению с результатами существующих работ (в них качество распознавания при смене используемой аппаратуры падает порядка на 20%). Отметим, что качество распознавания пользователей с использованием сокращения размерности признакового пространства на основе широко распространенного метода главных компонент в среднем на 5% ниже соответствующего качества распознавания при использовании отбора признаков на основе уровня их стабильности.
- Была проведена серия экспериментов, в результате которой было подтверждено высокое качество работы предложенных алгоритмов, а также подобраны оптимальные значения параметров построения временных окон, модели представления и используемых алгоритмов обработки признаков. В результате использования предложенной комбинации алгоритмов – отбора признаков по уровню их стабильности на основе критерия Колмогорова-

Смирнова и дальнейшей дискретизации наиболее стабильных признаков по квантилям удастся достичь качества распознавания порядка 0.87–0.92 ROC AUC, что значительно превосходит качество аутентификации пользователей при использовании методов, рассмотренных в существующих научных работах.

Отметим, что предложенные алгоритмы и подходы тестировались на трех наборах данных, что подтверждает их высокое качество работы и отсутствие привязки к определенному набору тестовых данных.

### 3 ПОСТРОЕНИЕ МОДЕЛИ КЛАВИАТУРНОГО ПОЧЕРКА ПОЛЬЗОВАТЕЛЯ

*При работе (при подготовке) над данным разделом диссертации использованы следующие публикации автора, в которых, согласно Положению о присуждении ученых степеней в МГУ, отражены основные результаты, положения и выводы исследования:*

- *Методы поиска исключений в потоках сложноструктурированных данных / М. А. Казачук, М. И. Петровский, И. В. Машечкин, О. Е. Горохов //Вестник Московского университета. Серия 15: Вычислительная математика и кибернетика. – 2019. – № 3. – С. 17-28.*

В предыдущем разделе данной диссертационной работы было сформировано пространство стабильных по времени признаковых характеристик, а также были предложены новый подход к отбору наиболее значимых признаков на основе уровня их стабильности с использованием критерия Колмогорова-Смирнова и подход к дальнейшей обработке признаков на основе дискретизации их по квантилям, совместное использование которых позволило достичь высокого качества распознавания пользователей на основе анализа их клавиатурного почерка.

Формально, с использованием предложенной комбинации алгоритмов для каждого рассматриваемого пользователя  $U_i$  на основе имеющейся последовательности клавиатурных событий  $H(U_i)$ , (6–7) был сформирован набор векторов признаков  $F(U_i)$ , характеризующих динамику работы данного пользователя с клавиатурой компьютера:

$$H(U_i) = (A_{i1}, A_{i2}, \dots, A_{iN}) \rightarrow F(U_i) = (V_{i1}, V_{i2}, \dots, V_{iM}), \quad (16)$$

где каждый вектор признаков  $V_{ij}$  представляет собой числовой вектор, состоящий из вещественных чисел.

Следующим этапом решения задачи динамической аутентификации пользователей на основе анализа динамики их работы с клавиатурой компьютера является построение модели рассматриваемого пользователя  $U_i$  на основе соответствующего ему набора векторов признаков  $F(U_i)$  :

$$M(V_i^* | F(U_i)), \quad (17)$$

где  $V_i^*$  – вектор признаков, построенный для текущей активности пользователя  $U_i$ ;  $M$  – решающая функция, выполняющая отображение из пространства признаков в пространство действительных чисел.

На основе данной модели, необходимо уметь определять, является ли человек, работающий в данный момент за компьютером, тем, за кого он себя выдает (пользователем, для которого мы построили модель) или нет – данная задача является задачей одноклассовой классификации.

Отметим, что одной из основных проблем задач одноклассовой классификации является отсутствие размеченных примеров данных нелегитимного класса, вследствие чего подбор оптимальных значений метапараметров одноклассовых классификаторов является достаточно затруднительным.

Дополнительно заметим, что зачастую необходимо решать задачу оценки аномальности поведения пользователя за длительный период (например, целую сессию) его работы за компьютером. С помощью классификатора мы сможем получить набор откликов для всех временных окон в рамках рассматриваемого временного интервала. Необходимо на основе полученной последовательности откликов уметь получать единое число – степень аномальности поведения пользователя за продолжительный промежуток времени (являясь агрегированной характеристикой, данная величина позволит более точно оценить аномальность действий пользователя). В существующих работах решение данной проблемы не предлагается.

*Задачей данного раздела является исследование и разработка методов построения модели пользователя, позволяющих достичь высокого качества распознавания (более 0.90 ROC AUC) в данной задаче, разработка метода подбора оптимальных значений метапараметров алгоритмов одноклассовой классификации, а также разработка методов оценки аномальности поведения пользователя на основе анализа целых сессий работы за компьютером.*

*К предлагаемому решению поставленной задачи предъявляются следующие требования:*

- Предлагаемый метод построения модели пользователя должен показывать высокое качество распознавания (превышающее качество работы современных методов машинного обучения в данной задаче – выше 0.90 ROC AUC), а также быть способным работать в режиме, близком к режиму реального времени (поскольку в рассматриваемой задаче необходимо вовремя реагировать на действия злоумышленника);
- Предлагаемый метод подбора оптимальных значений метапараметров алгоритмов одноклассовой классификации должен быть устойчив к смене



тестового набора данных. Иными словами, при использовании данного метода подбора значений метапараметров, рассматриваемые методы одноклассовой классификации должны показывать высокое качество работы в независимости от данных, на которых происходит тестирование построенных моделей. Также предлагаемый метод не должен использовать информацию о специфике данных нелегитимного класса;

- Предлагаемый метод оценки аномальности поведения пользователя на основе анализа целых сессий работы за компьютером должен на основе полученной последовательности откликов уметь получать единое число – степень аномальности поведения пользователя за продолжительный промежуток времени.

### **3.1 Задача поиска аномалий в данных. Определение исключения**

Для решения задачи построения модели клавиатурного почерка пользователя в данной диссертационной работе предлагается рассмотреть методы поиска исключений в данных.

Методы поиска исключений играют важную роль при решении многих прикладных задач, в первую очередь связанных с безопасностью [16]. В таких задачах как правило доступны данные только одного, легитимного класса. А примеров нелегитимного целевого класса мало и зачастую их тяжело выделить путем «ручной» разметки. Данные задачи называются задачами одноклассовой классификации. В них легитимная модель строится без использования образцов других классов, хотя может делаться предположение, что в обучающей выборке может присутствовать определенный небольшой процент наблюдений из целевого нелегитимного класса. Такие задачи возникают в областях компьютерной [14], финансовой [90] и общественной безопасности [91]. Рассматриваемая задача динамической аутентификации пользователей по клавиатурному почерку также относится к данному классу задач. Неформально под аномалией (или исключением) понимается объект или событие в выборке, чьи признаки или их комбинации не соответствуют зависимостям, характерным для остальных объектов или событий в данной выборке [92]. Для поиска аномалий традиционно используются статистические (вероятностные) и метрические подходы, а также методы, основанные на анализе отклонений. В статистическом подходе под исключением понимается резко выделяющееся наблюдение (используется вероятностная интерпретация понятия

исключения как маловероятного события). В метрическом подходе используется геометрическая интерпретация понятия исключения: исключением является наблюдение, удаленное от большей части наблюдений в выборке. Методы, не использующие ни вероятностную, ни геометрическую интерпретацию понятия исключения, относятся к методам, основанным на анализе отклонений. Следует отметить, что данные в задачах, связанных с безопасностью, как правило имеют пространство признаков большой размерности. Вследствие чего, многие признаки оказываются нерелевантными с точки зрения выделения целевого класса, а многие признаки являются взаимозависимыми. Это затрудняет использование традиционных вероятностных и метрических подходов, которые не устойчивы к проклятию размерности. Перспективным подходом в этой ситуации является использование kernel методов (методов, основанных на переходе из исходного пространства признаков (Input space) в пространство признаков большей размерности (RKHS) с использованием потенциальной (kernel) функции и поиске зависимостей в новом результирующем пространстве) [92]. Наиболее популярными среди них являются одноклассовый SVM (или SVC) и kernel-версия метода главных компонент – Kernel PCA. Но данные методы обладают рядом недостатков. В частности, SVC ищет оптимальный центр множества образов наблюдений в пространстве характеристик высокой размерности (RKHS) и ограничивает их гиперсферой минимального радиуса, считая наблюдения, чьи образы выходят за рамки гиперсферы, исключениями. Очевидным недостатком здесь является сферичность области, поскольку зависимости между исходными признаками также могут привести к зависимостям между индуцированными признаками в RKHS, а значит, логичнее использовать не сферические области, а эллиптические. С другой стороны, Kernel PCA строит как раз эллиптические области в RKHS, содержащие образы допустимых наблюдений, что позволяет эффективно работать с сильно зависимыми признаками. Однако, Kernel PCA фиксирует центр распределения и не пересчитывает его с учетом найденных выбросов.

Для преодоления этих недостатков в данной диссертационной работе предлагается новый метод обнаружения аномалий, основанный на анализе отклонений. Основной идеей предложенного метода является переход из исходного пространства признаков в пространство характеристик высокой размерности и дальнейшая нечеткая кластеризация образов наблюдений в результирующем пространстве с использованием метрики Махаланобиса для расчета расстояний между объектами и центром кластера. В пространстве большей размерности строится один общий нечеткий кластер эллипсоидальной формы, где каждый образ наблюдения имеет свою степень принадлежности (типичности). Центр кластера также находится в индуцированном

пространстве и итерационно пересчитывается. Настройки алгоритма кластеризации (параметры регуляризации и степень нечеткости) задаются так, чтобы степень принадлежности «основной части» образов наблюдений кластера в RKHS была достаточно высока (выше заданного порога, например выше 0.5). Исключениями считаются наблюдения со степенью типичности, меньшей заданного порога. Расстояние Махаланобиса вычисляется путем проецирования данных на пространство, заданное собственными векторами матрицы ковариации в RKHS. Благодаря этому удастся учесть разброс дисперсии и корреляции между признаками в RKHS. Тем самым, результирующий кластер будет иметь не сферическую, а эллипсоидальную форму в RKHS, что позволит более точно описать основную часть легитимной обучающей выборки и построить более точную одноклассовую модель соответственно.

### 3.2 Классические *kernel*-методы поиска исключений

Наиболее известными методами обнаружения аномалий в RKHS являются такие методы машинного обучения как SVC [93], Single Class SVM [94], Kernel PCA [95], а также нечеткий метод поиска исключений на основе потенциальных функций (Fuzzy) [96]. Ключевым моментом данных методов является переход из исходного пространства анализируемых объектов  $X$  в пространство характеристик большей размерности  $H$ , в котором образы объектов  $\varphi(x)$  и  $\varphi(y)$  связаны с исходными объектами  $x$  и  $y$  следующим образом [16, 92]:

$$K(x, y) = \langle \varphi(x), \varphi(y) \rangle_H, \quad (18)$$

где  $K$  – ядровая (потенциальная) функция. Расстояние  $d$  между объектами  $x$  и  $y$  в новом пространстве будет вычисляться следующим образом:

$$d(x, y) = \sqrt{K(x, x) - 2K(x, y) + K(y, y)}. \quad (19)$$

Выбор используемой потенциальной функции зависит от специфики задачи. В задачах обнаружения аномалий часто используется радиально-базисная функция Гаусса (RBF):

$$K(x, y) = e^{\frac{-\|x-y\|^2}{2\sigma^2}}, \quad (20)$$

где  $\sigma$  – ширина используемого ядра (параметр алгоритма).

Целью перехода в пространство характеристик большей размерности является эффективное использование более простых геометрических структур для описания зависимостей во входных данных. Демонстрационный пример, отражающий принцип перехода в пространство признаков высокой размерности, представлен на Рисунке 7.

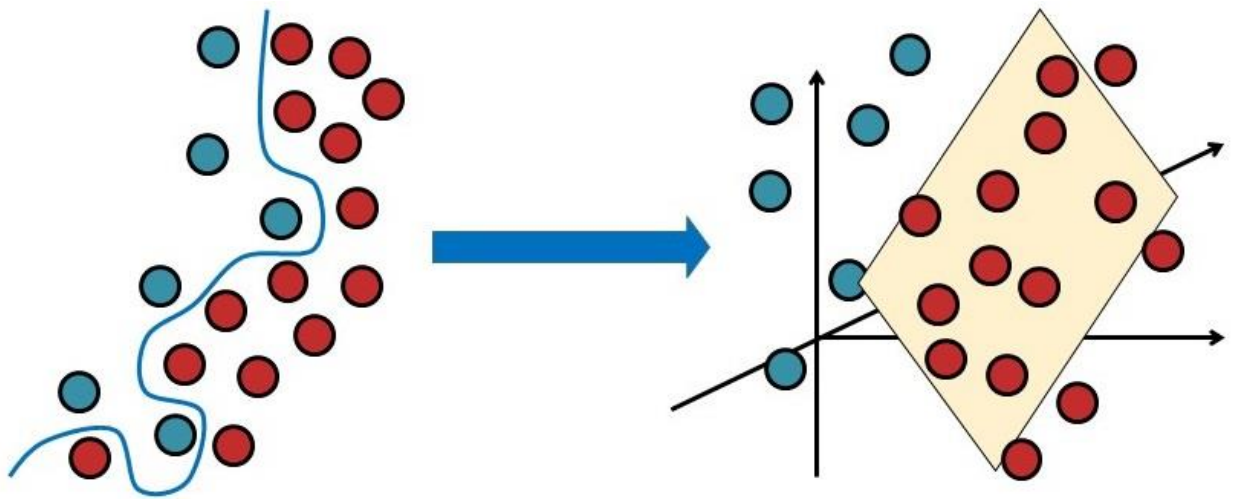


Рисунок 7 — Принцип перехода в пространство признаков  
большой размерности (RKHS).

Рассмотрим методы, использующие переход в пространство характеристик  
большой размерности, подробнее.

Как было отмечено ранее, в методе SVC [16, 93] объекты из исходного множества  
 неявно отображаются с помощью потенциальной функции в пространство характеристик  
 высокой размерности, где далее происходит поиск гиперсферы минимального радиуса,  
 содержащей внутри «основную часть» образов объектов из исходного множества.  
 Искключениями считаются объекты, чей образ лежит за пределами найденной гиперсферы.  
 Таким образом, в данном методе решается следующая задача оптимизации:

$$\min_{\xi \in \mathbb{R}^N, R \in \mathbb{R}, a \in H} \left[ R^2 + \frac{1}{\nu N} \sum_{i=1}^N \xi_i \right], \quad (21)$$

$$\|\varphi(x_i) - a\|_H^2 \leq R^2 + \xi_i, \forall i \in [1, N], \quad (22)$$

где  $R$  – радиус построенной гиперсферы;  $a$  – центр гиперсферы;  $N$  – число объектов в  
 обучающей выборке  $X$ ,  $0 < \nu \leq 1$  – предопределенный процент исключений;  $\xi_i$  –  
 дополнительные переменные. Для решения данной задачи предлагается использовать  
 метод множителей Лагранжа. В результате бинарная решающая функция будет иметь  
 следующий вид:

$$f(z) = \text{sgn}(R^2 - \sum_{i,j=1}^N \beta_i \beta_j K(x_i, x_j) + 2 \sum_{i=1}^N \beta_i K(x_i, z) - K(z, z)), \quad (23)$$

где  $\beta_i$   $i \in [1, N]$  – множители Лагранжа.  $\beta_i = \frac{1}{\nu N}$  для исключений,  $0 < \beta_i < \frac{1}{\nu N}$  для  
 граничных объектов и  $\beta_i = 0$  для остальных объектов из  $X$ ;  $z$  – тестовое наблюдение.

Демонстрационный пример, отражающий принцип работы метода SVC,  
 представлен на Рисунке 8.

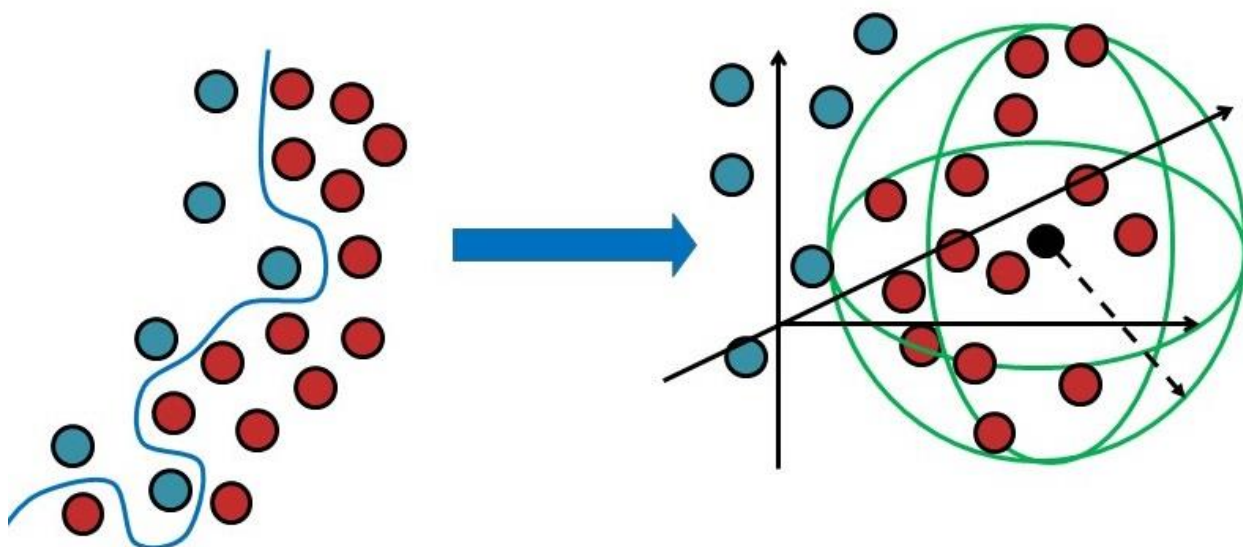


Рисунок 8 — Принцип работы метода SVC.

Метод Single Class SVM [94] аналогичен методу SVC: он находит гиперплоскость, отделяющую «основную часть» образов объектов от начала координат. Исключениями считаются объекты, чей образ лежит ближе, чем найденная гиперплоскость, к началу координат. При использовании RBF-ядра результаты работы методов Single Class SVM и SVC получаются достаточно близкими.

Настраиваемым параметром данных алгоритмов является отношение ожидаемого числа исключений к общему числу объектов рассматриваемой выборки  $\nu$  (21). Основным недостатком данных методов является чрезмерная простота найденных границ для определения исключений в RKHS, которые не учитывают в полной мере зависимости между признаками. Другой недостаток – как у любого метода опорных векторов, результирующая модель зависит только от наблюдений, лежащих на границе, и не дает корректной возможности оценить степень аномальности для наблюдений внутри границы.

Данную проблему решает нечеткий метод поиска исключений на основе потенциальных функций (Fuzzy) [16, 92, 96], который является модификацией метода SVC. В данном методе с помощью потенциальной функции строится отображение исходного множества анализируемых объектов в пространство характеристик. Вместо гиперсферы, содержащей образы анализируемых объектов, в пространстве характеристик строится один общий нечеткий кластер сферической формы, содержащий все образы анализируемых объектов таким образом, чтобы степень принадлежности «основной части» объектов была достаточно высока. Степень принадлежности образа анализируемого объекта этому кластеру интерпретируется как «мера типичности». Исключениями считаются те объекты, чья степень типичности меньше заданного порога.

Таким образом, задача сводится к следующей задаче оптимизации:

$$\min_{U,a,\eta} J(U,a,\eta) = \sum_{i=1}^N u_i^m (\varphi(x_i) - a)^2 + \eta \sum_{i=1}^N (1 - u_i)^m, \quad (24)$$

где  $a$  – центр нечеткого кластера в пространстве характеристик;  $N$  – число анализируемых объектов;  $U$  – вектор значений, где  $u_i \in [0,1]$  – степень типичности  $i$ -го объекта;  $m > 1$  – степень нечеткости (параметр, определяющий скорость убывания значения степени принадлежности в зависимости от расстояния до центра кластера) и  $\eta > 0$  – параметр, контролирующий размер или радиус нечеткого кластера в RKHS.

На основе итерационного метода, минимизирующего функционал  $J(U,a,\eta)$ , для каждого анализируемого объекта  $z$  находится значение его меры типичности  $u(z)$ . Вычисления производятся следующим образом:

$$u(z) = \left[ 1 + \left( \frac{\sum_{j=1}^N u_j^m \sum_{i=1}^N u_i^m K(x_i, x_j)}{\eta (\sum_{i=1}^N u_i^m)^2} - 2 \frac{\sum_{i=1}^N u_i^m K(z, x_i)}{\eta \sum_{i=1}^N u_i^m} + \frac{K(z, z)}{\eta} \right)^{\frac{1}{m-1}} \right]^{-1}, \quad (25)$$

где  $K$  – используемая ядровая (потенциальная) функция. Данный метод является методом поиска аномалий, основанным на анализе отклонений. Заметим, что в отличие от метода SVC, в Fuzzy каждому объекту выборки присваивается определенный вес (степень типичности), благодаря чему уменьшается степень воздействия менее значимых (менее типичных) объектов выборки  $X$  и увеличивается влияние более значимых объектов данной выборки на результат классификации нового рассматриваемого объекта.

Стоит заметить, что во входных данных могут присутствовать корреляции и зависимости, которые будут сохраняться в модифицированном виде и в RKHS. В следствие чего использование гиперсферы, сферического нечеткого кластера или разделяющей гиперплоскости для построения модели не является оптимальным: в данном случае обучающую выборку необходимо описать с помощью эллипсоида в RKHS. Использование эллипсоида вместо гиперсферы позволяет учесть масштаб разброса данных по разным направлениям, взаимозависимости признаков и более точно описать строящуюся модель.

Данную проблему частично решает метод поиска аномалий Kernel PCA [16, 95]. В данном методе после перехода в пространство высокой размерности выполняется отбор главных компонент методом PCA, а далее происходит построение уже не гиперсферы, а эллипсоида, содержащего большую часть образов объектов исходного множества. Главным компонентам соответствуют собственные векторы с наибольшими собственными значениями. Количество отбираемых главных компонент является параметром алгоритма.

В методе Kernel PCA в качестве меры аномальности вводится понятие реконструкционной ошибки, которую необходимо минимизировать. Для каждого объекта  $x_i$  обучающей выборки данное значение вычисляется следующим образом:

$$\|\tilde{\varphi}(x_i) - P\tilde{\varphi}(x_i)\|_H^2 = (\tilde{\varphi}(x_i), \tilde{\varphi}(x_i)) - 2(\tilde{\varphi}(x_i), P\tilde{\varphi}(x_i)) + (P\tilde{\varphi}(x_i), P\tilde{\varphi}(x_i)), \quad (26)$$

где  $\tilde{\varphi}(x_i)$  – центрированный образ данного объекта в пространстве характеристик высокой размерности:  $\tilde{\varphi}(x_i) = \varphi(x_i) - \frac{1}{N} \sum_{i=1}^N \varphi(x_i)$ ;  $P\tilde{\varphi}(x_i)$  – проекция  $\tilde{\varphi}(x_i)$  на подпространство с максимальной дисперсией данных.

Таким образом, уровень аномальности  $p(z)$  тестируемого вектора  $z$  вычисляется следующим образом:

$$p(z) = K(z, z) - \frac{2}{N} \sum_{i=1}^N K(z, x_i) + \frac{1}{N^2} \sum_{i,j=1}^N K(x_i, x_j) - \sum_{l=1}^q \left( \sum_{i=1}^N a_i^l (K(z, x_i) - \frac{1}{N} \sum_{r=1}^N K(x_i, x_r) - \frac{1}{N} \sum_{r=1}^N K(z, x_r) + \frac{1}{N^2} \sum_{r,s=1}^N K(x_r, x_s)) \right)^2, \quad (27)$$

где  $N$  – число анализируемых объектов;  $a_i^l$  – весовые коэффициенты;  $q$  – количество используемых главных компонент. Исключениями считаются объекты с наибольшим значением уровня аномальности.

Но как видно из формул (26, 27) результирующая модель хотя и строит эллипсоидные области в RKHS, центр этих областей фиксируется в центре масс распределения и не пересчитывается оптимальным образом. Кроме того, модель существенно зависит от параметра  $q$ , задающего число оставленных главных компонент.

Демонстрационный пример, отражающий принцип работы метода Kernel PCA, представлен на Рисунке 9.

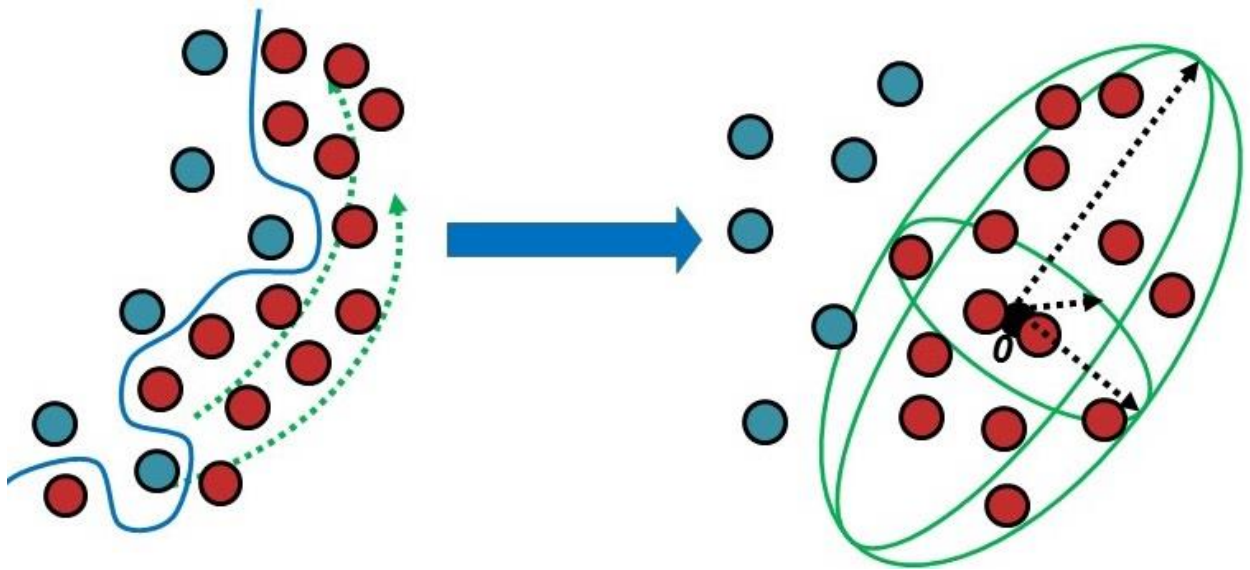


Рисунок 9 — Принцип работы метода Kernel PCA.

### **3.3 Нечеткий метод выявления аномалий в данных на основе эллиптической кластеризации в RKHS**

Для решения вышеописанных проблем, в данной диссертационной работе предлагается нечеткий метод выявления аномалий в данных на основе эллиптической кластеризации (Ellipsoidal Single Fuzzy Clustering – ESFC) в RKHS. Данный метод является модификацией описанного выше нечеткого метода поиска исключений на основе потенциальных функций (Fuzzy) [96]. Он сочетает в себе достоинства данного метода и дополняет их достоинствами методов SVM [93] и Kernel PCA [95]. Ключевой особенностью предложенного метода является использование метрики Махаланобиса для вычисления расстояний между объектами и центром кластера в пространстве признаков высокой размерности [16]. Формирование данного пространства осуществляется неявно с использованием радиально-базисной функции Гаусса. Расстояние Махаланобиса позволяет учесть разброс данных и значительные корреляции в RKHS. Данная метрика вычисляется путем проецирования данных на пространство, заданное собственными векторами матрицы ковариации обучающей выборки в RKHS. В результате построенный кластер, в отличие от кластера в методе Fuzzy, имеет не сферическую, а эллипсоидальную форму (тем самым, используется преимущество метода Kernel PCA), что позволяет построить более точную модель. Аналогично методу Fuzzy, каждому объекту выборки присваивается определенный вес (степень типичности), благодаря чему уменьшается степень воздействия на центр кластера и его форму менее значимых (менее типичных) объектов обучающей выборки  $X$  и увеличивается влияние более значимых объектов данной выборки на результат классификации нового рассматриваемого объекта. Сама степень типичности тем выше, чем ближе в метрике Махаланобиса образ наблюдения к центру кластера в RKHS. В отличие от метода Kernel PCA, в предлагаемом методе центр кластера пересчитывается на каждой итерации оптимальным образом, а не фиксируется в центре масс распределения, что позволяет более точно описать обучающую выборку. Также, в данном методе, в отличие от метода SVM, нет необходимости хранить в оперативной памяти всю обучающую выборку – достаточно лишь вектора степеней типичности входящих в нее наблюдений.

Рассмотрим данный метод более подробно.



### 3.3.1 Формулировка и обоснование метода

Рассмотрим задачу нечеткой кластеризации в пространстве характеристик высокой размерности (RKHS) в случае построения одного кластера. Имеется конечное множество анализируемых объектов  $\{x_i\}_{1 \leq i \leq N} \subset X$ . Также, на  $X \times X$  определена потенциальная функция  $K: X \times X \rightarrow R_0^+$   $\varphi: X \rightarrow H$ , которая задает отображение исходного признакового пространства в пространство признаков большей размерности. Необходимо в RKHS (в данном случае – в  $H$ ) построить единый кластер эллипсоидальной формы, включающий в себя все образы анализируемых объектов таким образом, что степень принадлежности (типичности)  $u_i$  каждого образа  $\varphi(x_i)$  данному кластеру вычисляется путем решения следующей задачи оптимизации:

$$\min_{U, a, \eta} E(U, a, \eta) = \sum_{i=1}^N u_i^m \|a - \varphi(x_i)\|_C^2 + \eta \sum_{i=1}^N (1 - u_i)^m, \quad (28)$$

где  $a$  – центр нечеткого кластера в пространстве характеристик;  $N$  – число анализируемых объектов;  $U$  – вектор значений, где  $u_i \in [0, 1]$  – степень типичности  $i$ -го объекта;  $m > 1$  – степень нечеткости (параметр, определяющий скорость убывания значения степени принадлежности в зависимости от расстояния до центра кластера) и  $\eta > 0$  – параметр, контролирующий размер кластера;  $\|a - \varphi(x_i)\|_C^2$  – квадрат расстояния Махаланобиса в пространстве характеристик от образа  $\varphi(x_i)$  до  $a$ :

$$\|a - \varphi(x_i)\|_C^2 = (a - \varphi(x_i))^T C^{-1} (a - \varphi(x_i)), \quad (29)$$

где  $C$  – матрица ковариации. Введем обозначение:  $M = C^{-1}$ . Таким образом, матрица  $M$  является обратной к матрице  $C$ . Исключениями будут считаться объекты со степенью типичности, меньшей порога, заданного априори [16].

Для каждого входного анализируемого объекта  $x_k$  расстояние Махаланобиса  $D_k(a)$  между его образом  $\varphi(x_k)$  и центром нечеткого кластера  $a$  в пространстве характеристик  $H$  вычисляется следующим образом:

$$D_k(a) = \|a - \varphi(x_k)\|_C^2 = (a - \varphi(x_k))^T M (a - \varphi(x_k)) = \sum_{j=1}^N \left( \sum_{i=1}^N (a - \varphi(x_k)) M_{ij} \right) (a - \varphi(x_k)) = \sum_{j=1}^N \sum_{i=1}^N M_{ij} a a - M_{ij} a \varphi(x_k) - M_{ij} \varphi(x_k) a + M_{ij} \varphi(x_k) \varphi(x_k). \quad (30)$$

Докажем, что центр нечеткого кластера  $a$  в RKHS можно выразить через линейную комбинацию образов входных объектов  $\varphi(x_j)$  следующим образом:

$$a = \sum_{j=1}^N a_j \varphi(x_j). \quad (31)$$

Предположим, что это не так, и  $a = a^* + a^{**}$ , где  $a^*$  – слагаемое, представимое в виде линейной комбинации  $\varphi(x_j)$ ,  $a^{**}$  – слагаемое, не представимое в виде линейной

комбинации  $\varphi(x_j)$  соответственно. Таким образом,  $a^* = \sum_{j=1}^N a_j \varphi(x_j)$  и  $\forall j \in [1, N]$  выполнено  $\langle a^*, \varphi(x_j) \rangle_c = 0$ , а также,  $\langle a^*, a^{**} \rangle_c = 0$ .

Следовательно,  $\forall k \in [1, N]$  выполняется:

$$\begin{aligned}
 D_k(a) &= \|a - \varphi(x_k)\|_c^2 = (a - \varphi(x_k))^T M (a - \varphi(x_k)) = \\
 &= \sum_{j=1}^N \left( \sum_{i=1}^N (a - \varphi(x_k)) M_{ij} \right) (a - \varphi(x_k)) = \\
 &= \sum_{j=1}^N \sum_{i=1}^N M_{ij} a a - M_{ij} a \varphi(x_k) - M_{ij} \varphi(x_k) a + M_{ij} \varphi(x_k) \varphi(x_k) = \\
 &= \sum_{j=1}^N \sum_{i=1}^N M_{ij} (\langle a, a \rangle_c - 2 \langle \varphi(x_k), a \rangle_c + K(x_k, x_k)) = \\
 &= \sum_{j=1}^N \sum_{i=1}^N M_{ij} (\langle a^* + a^{**}, a^* + a^{**} \rangle_c - 2 \langle \varphi(x_k), a^* + a^{**} \rangle_c + K(x_k, x_k)) = \\
 &= \sum_{j=1}^N \sum_{i=1}^N M_{ij} (\langle a^*, a^* \rangle_c + \langle a^{**}, a^{**} \rangle_c - 2 \langle \varphi(x_k), a^* \rangle_c + K(x_k, x_k)) = \\
 &= \|a^* - \varphi(x_k)\|_c^2 + \langle a^{**}, a^{**} \rangle_c.
 \end{aligned} \tag{32}$$

Получаем, что  $\|a - \varphi(x_k)\|_c^2 \geq \|a^* - \varphi(x_k)\|_c^2$ , причем равенство достигается только при  $\langle a^{**}, a^{**} \rangle_c = 0$ . Таким образом,  $a = \sum_{j=1}^N a_j \varphi(x_j)$ , что и требовалось доказать.

Для решения поставленной задачи оптимизации, рассмотрим необходимые условия минимума функционала (28). Для достижения экстремума  $E$ , необходимо равенство нулю его частных производных  $\frac{\partial E}{\partial u_i}$  и  $\frac{\partial E}{\partial a}$ :

$$1) \quad \frac{\partial E}{\partial u_i} = 0$$

Получаем:

$$D_i(a) = \|a - \varphi(x_i)\|^2 = (a - \varphi(x_i))^T M (a - \varphi(x_i))$$

$$E = \sum_{i=1}^N u_i^m D_i(a) + \eta \sum_{i=1}^N (1 - u_i)^m$$

$$\frac{\partial E}{\partial u_i} = m u_i^{m-1} D_i(a) - \eta m (1 - u_i)^{m-1} = 0$$

Из данного уравнения выражаем  $u_i$  – степень типичности  $i$ -го анализируемого объекта:

$$u_i = \left[ 1 + \left( \frac{D_i(a)}{\eta} \right)^{\frac{1}{m-1}} \right]^{-1}. \quad (33)$$

$$2) \quad \frac{\partial E}{\partial a} = 0$$

Получаем:

$$E = \sum_{i=1}^N u_i^m (a - \varphi(x_i))^T M (a - \varphi(x_i)) + \eta \sum_{i=1}^N (1 - u_i)^m$$

$$E = \sum_{i=1}^N u_i^m \left( (M^T (a - \varphi(x_i))), (a - \varphi(x_i)) \right) + \eta \sum_{i=1}^N (1 - u_i)^m$$

Поскольку  $M$  – симметричная матрица, справедливо равенство  $M = M^T$ . Отсюда:

$$E = \sum_{i=1}^N u_i^m \left( (M(a - \varphi(x_i))), (a - \varphi(x_i)) \right) + \eta \sum_{i=1}^N (1 - u_i)^m$$

$$\frac{\partial E}{\partial a} = \sum_{i=1}^N u_i^m M(a - \varphi(x_i)) = 0$$

$$M \left( \sum_{i=1}^N u_i^m a - \sum_{i=1}^N u_i^m \varphi(x_i) \right) = 0$$

Из данного уравнения выражаем  $a$  – центр нечеткого кластера в пространстве характеристик высокой размерности:

$$a = \frac{\sum_{i=1}^N u_i^m \varphi(x_i)}{\sum_{i=1}^N u_i^m}. \quad (34)$$

На основании (31), получаем:

$$a_i = \frac{(u_i)^m}{\sum_{i=1}^N (u_i)^m}. \quad (35)$$

Таким образом, необходимым условием минимума (28) является выполнение равенств (33) и (35).

Также, на основании (33) можем заключить, что  $u_i \in [0,1]$ . Дополнительно, из (33) мы можем определить семантику параметра  $\eta$ : он определяет радиус кластера, на котором образы анализируемых объектов имеют степень типичности, равную 0.5. Тем самым, он отделяет более типичные объекты кластера от менее типичных. Более типичные объекты будут иметь степень типичности, большую 0.5. Менее типичные – степень типичности, меньшую 0.5, соответственно. Значение параметра  $\eta$  можно либо задавать явно, либо вычислять на основе задаваемого априори ожидаемого процента исключений (аналогично методу SVC).

Дополнительно, распишем ковариационную матрицу  $C$ :

$$\begin{aligned}
C_{ij} &= (\varphi(x_i) - a, \varphi(x_j) - a) = \\
&\left( \varphi(x_i) - \frac{\sum_{r=1}^N u_r^m \varphi(x_r)}{\sum_{r=1}^N u_r^m}, \varphi(x_j) - \frac{\sum_{s=1}^N u_s^m \varphi(x_s)}{\sum_{s=1}^N u_s^m} \right) = \\
&K_{ij} - \frac{\sum_{r=1}^N u_r^m K_{ir}}{\sum_{r=1}^N u_r^m} - \frac{\sum_{r=1}^N u_r^m K_{rj}}{\sum_{r=1}^N u_r^m} + \frac{\sum_{r,s=1}^N u_r^m u_s^m K_{rs}}{(\sum_{r=1}^N u_r^m)^2}.
\end{aligned} \tag{36}$$

Исходя из (30, 34–36), рассчитаем значение выражения  $\|a\|_C^2$ , то есть квадрат нормы  $a$  в пространстве характеристик высокой размерности:

$$\begin{aligned}
\|a\|_C^2 &= \sum_{j=1}^N \sum_{i=1}^N M_{ij} a_i a_j = \sum_{j=1}^N \sum_{i=1}^N M_{ij} a_i a_j K_{ij} = \\
&\sum_{j=1}^N \sum_{i=1}^N M_{ij} K_{ij} \frac{(u_i)^m (u_j)^m}{\sum_{s=1}^N \sum_{r=1}^N (u_s)^m (u_r)^m}.
\end{aligned} \tag{37}$$

Поскольку  $a_i \geq 0$  и  $\sum_{i=1}^N a_i = 1$ , то при условии, что  $M_{ij}$  и  $K_{ij}$  ограничены константой, можем заключить, что значение  $\|a\|_C^2$  также является ограниченным константой. Следовательно, максимальное расстояние в RKHS ограничено.

### 3.3.2 Алгоритм поиска исключений с использованием процедуры блочного покоординатного спуска

Для решения рассматриваемой задачи оптимизации (28) и поиска исключений соответственно, предлагается следующий итерационный алгоритм на основе процедуры блочного покоординатного спуска [16, 92, 97]:

#### Шаг 0. Инициализация.

$$u_i^{(0)} = \text{random}([0, 1]), \eta^{(0)} = 0.5 \tag{38}$$

#### Шаг 1. $l$ -ая итерация.

Вычисление центра нечеткого кластера, соответствующего данной итерации,  $a^{(l)} = \arg \min E(U^{(l-1)}, a^{(l-1)}, \eta^{(l-1)})$  следующим образом:

$$a^{(l)} = \sum_{i=1}^N a_i^{(l)} \varphi(x_i), \text{ где } a_i^{(l)} = \frac{(u_i^{(l-1)})^m}{\sum_{i=1}^N (u_i^{(l-1)})^m}. \tag{39}$$

Поскольку ковариационная матрица  $C$  зависит от степеней типичности образов анализируемых объектов  $\varphi(x_i)$  и центра нечеткого кластера  $a$  в пространстве характеристик, на каждой итерации данного алгоритма необходимо осуществлять ее пересчет по следующей формуле:

$$C_{ij}^{(l)} = (\varphi(x_i) - a^{(l)}) (\varphi(x_j) - a^{(l)}) =$$

$$\left( \varphi(x_i) - \frac{\sum_{r=1}^N (u_r^{(l-1)})^m \varphi(x_r)}{\sum_{r=1}^N (u_r^{(l-1)})^m} \right) \left( \varphi(x_j) - \frac{\sum_{s=1}^N (u_s^{(l-1)})^m \varphi(x_s)}{\sum_{s=1}^N (u_s^{(l-1)})^m} \right) =$$

$$K_{ij} - \frac{\sum_{r=1}^N (u_r^{(l-1)})^m (K_{ir} + K_{rj})}{\sum_{r=1}^N (u_r^{(l-1)})^m} + \frac{\sum_{r,s=1}^N (u_r^{(l-1)})^m (u_s^{(l-1)})^m K_{rs}}{(\sum_{r=1}^N (u_r^{(l-1)})^m)^2}, \quad (40)$$

где  $K_{ij} = K(x_i, x_j)$ . Далее происходит вычисление обратной к ней матрицы  $M = C^{-1}$ .

**Шаг 2.** Расчет расстояния  $D_k(a^{(l)})$  от центра кластера  $a^{(l)}$  до образа  $k$ -го наблюдения  $x_k$  на данной ( $l$ -ой) итерации для каждого наблюдения обучающей выборки по следующей формуле:

$$D_k(a^{(l)}) = \|a^{(l)} - \varphi(x_k)\|_C^2 = (a^{(l)} - \varphi(x_k))^T M (a^{(l)} - \varphi(x_k)) =$$

$$\sum_{j=1}^N \sum_{i=1}^N \left( M_{ij}^{(l)} \frac{\sum_{r,s=1}^N (u_r^{(l-1)})^m (u_s^{(l-1)})^m K_{rs}}{(\sum_{r=1}^N (u_r^{(l-1)})^m)^2} - 2M_{ij}^{(l)} \frac{\sum_{r=1}^N (u_r^{(l-1)})^m K_{rk}}{\sum_{r=1}^N (u_r^{(l-1)})^m} + M_{ij}^{(l)} M_{kk}^{(l)} \right) =$$

$$\sum_{j=1}^N \sum_{i=1}^N M_{ij}^{(l)} M_{kk}^{(l)}. \quad (41)$$

Если параметр  $\eta$  не зафиксирован, он вычисляется как расстояние до образа  $(N - q)$ -го наиболее удаленного от  $a$  наблюдения, в результате чего  $q$  наблюдений являются нетипичными (имеют степень типичности, меньшую 0.5). Параметр  $q$  (ожидаемый процент исключений) должен задаваться априори и быть фиксированным на протяжении всей работы алгоритма.

$$\eta^{(l)} = D_{i:q}^2(a^{(l)}), D_{i:1}^2(a^{(l)}) \geq D_{i:2}^2(a^{(l)}) \geq \dots \geq D_{i:N}^2(a^{(l)}). \quad (42)$$

**Шаг 3.** Вычисление новых значений степеней типичности элементов обучающей выборки  $U^{(l)} = \operatorname{argmin} E(U^{(l)}, a^{(l)}, \eta^{(l)})$ :

$$u_i^{(l)} = \left[ 1 + \left( \frac{D_i(a^{(l)})}{\eta^{(l)}} \right)^{\frac{1}{m-1}} \right]^{-1}. \quad (43)$$

Если  $\|U^{(l)} - U^{(l+1)}\|_F > \varepsilon$ , то переход на **Шаг 1**, иначе выход из алгоритма.

Значение решающей функции для расчета степени типичности  $u(z)$  рассматриваемого тестового наблюдения  $z$  вычисляется следующим образом:

$$u(z) = \left[ 1 + \left( \frac{D_z(a)}{\eta} \right)^{\frac{1}{m-1}} \right]^{-1}, \quad (44)$$

где расстояние Махаланобиса  $D_z(a)$  от образа  $\varphi(z)$  анализируемого объекта  $z$  до центра нечеткого кластера  $a$ , задаваемого формулой (39) в RKHS, вычисляется по следующей формуле:

$$D_z(a) = \|a - \varphi(z)\|_c^2 = (a - \varphi(z))^T M(a - \varphi(z)) = \sum_{j=1}^N \sum_{i=1}^N \left( M_{ij} \frac{\sum_{r,s=1}^N u_r^m u_s^m K_{rs}}{(\sum_{r=1}^N u_r^m)^2} - 2M_{ij} \frac{\sum_{r=1}^N u_r^m K(x_r, z)}{\sum_{r=1}^N u_r^m} + M_{ij} K(z, z) \right). \quad (45)$$

Здесь степень типичности  $u_r$ , соответствующая  $r$ -ому наблюдению обучающей выборки (43), обратная ковариационная матрица  $M$  (40) и параметр  $\eta$  (42) вычисляются по предложенному алгоритму ESFC.  $K(x_r, z)$  – значение потенциальной функции, вычисляемой для тестового наблюдения  $z$  и  $r$ -ого наблюдения  $x_r$  обучающей выборки  $X$ .

### 3.3.3 Исследование сходимости метода

Как было замечено ранее, предложенный итерационный алгоритм является алгоритмом блочного покоординатного спуска. В данной задаче мы можем выделить три независимых блока по  $N$  переменных:  $\psi = \psi_U \times \psi_a \times \psi_\eta$ . Для доказательства сходимости предложенного алгоритма необходимо доказать, что он сходится линейно к стационарной точке  $(U^*, a^*, \eta^*)$  при  $\forall$  начальном приближении [16, 92]. Для этого необходимо показать, что:

- 1)  $E(\bar{U}, \bar{a}, \eta)$  имеет глобальный минимум при фиксированных  $\bar{U} = (\bar{u}_1, \dots, \bar{u}_N) \in \psi_U = [0,1]^N$  и  $\bar{a} = (\bar{a}_1, \dots, \bar{a}_N) \in \psi_a = \{\bar{a} \in R^N \mid \sum_{i=1}^N \bar{a}_i = 1\}$ . В данной задаче это верно, поскольку  $E(\bar{U}, \bar{a}, \eta)$  является линейной возрастающей функцией одной переменной, заданной на отрезке;
- 2)  $E(\bar{U}, a, \bar{\eta})$  имеет глобальный минимум при фиксированных  $\bar{U} = (\bar{u}_1, \dots, \bar{u}_N) \in \psi_U = [0,1]^N$  и  $\bar{\eta}$ . Для доказательства данного факта вычислим значение  $\frac{\partial^2 E(\bar{U}, a, \bar{\eta})}{\partial a_i \partial a_j}$ :

$$\frac{\partial^2 E(\bar{U}, a, \bar{\eta})}{\partial a_i \partial a_j} = 2 \sum_{i=1}^N u_i^m \sum_{j,k=1}^N M_{kj}. \quad (46)$$

Заметим, что матрица  $M$  является матрицей, обратной к положительно определенной симметричной матрице  $C$ . Следовательно,  $M$  – также положительно определенная симметричная матрица.

Получили, что матрица вторых производных  $\frac{\partial^2 E(\bar{U}, a, \bar{\eta})}{\partial a_i \partial a_j}$  является положительно определенной. Следовательно, функция  $E(\bar{U}, a, \bar{\eta})$  является выпуклой и имеет глобальный минимум в точке

$$a_i = \frac{(u_i)^m}{\sum_{i=1}^N (u_i)^m}; \quad (47)$$

- 3)  $E(U, \bar{a}, \bar{\eta})$  имеет глобальный минимум при фиксированных  $\bar{a} = (\bar{a}_1, \dots, \bar{a}_N) \in \psi_a = \{\bar{a} \in R^N \mid \sum_{i=1}^N \bar{a}_i = 1\}$  и  $\bar{\eta}$ . Вычислим значение  $\frac{\partial^2 E(U, \bar{a}, \bar{\eta})}{\partial u_i \partial u_j}$ :

$$\frac{\partial^2 E(U, \bar{a}, \bar{\eta})}{\partial u_i \partial u_j} = 0, i \neq j$$

$$\frac{\partial^2 E(U, \bar{a}, \bar{\eta})}{(\partial u_i)^2} = m(m-1)[u_i^{m-2} D_i(a) + \eta(1-u_i)^{m-2}]. \quad (48)$$

Получили, что данная матрица вторых производных  $\frac{\partial^2 E(U, \bar{a}, \bar{\eta})}{(\partial u_i)^2}$  также является положительно определенной. Следовательно, функция  $E(U, \bar{a}, \bar{\eta})$  является выпуклой и имеет глобальный минимум в точке

$$u_i = \left[ 1 + \left( \frac{D_i(a)}{\eta} \right)^{\frac{1}{m-1}} \right]^{-1}. \quad (49)$$

Таким образом, мы показали, что предложенный метод линейно сходится к некоторой стационарной точке  $(U^*, a^*, \eta^*)$  при  $\forall$  начальном приближении. Следовательно, данный метод является сходящимся, что и требовалось доказать.

### 3.3.4 Исследование работы метода на простейших демонстрационных примерах

Рассмотрим работу предлагаемого метода ESFC на простейших демонстрационных примерах и сравним ее с работой методов SVC и Kernel PCA. На Рисунках 10–11 темными кругами изображены объекты простейшей искусственно сгенерированной обучающей выборки и показаны проекции контуров пороговых значений соответствующих решающих функций, построенных с использованием данных методов. Рассматривалось построение моделей с использованием как линейной (см. Рисунок 10), так и RBF (см. Рисунок 11) потенциальных функций. В случае RBF-функции, использовалось значение соответствующего параметра – ширины ядра, равное 0.01.

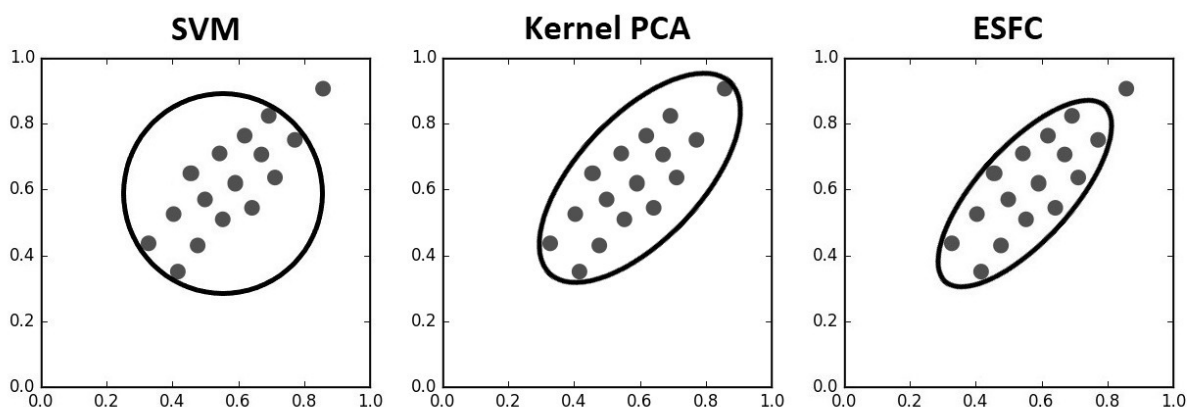


Рисунок 10 — Сравнение работы методов SVC, Kernel PCA и ESFC при использовании линейного ядра

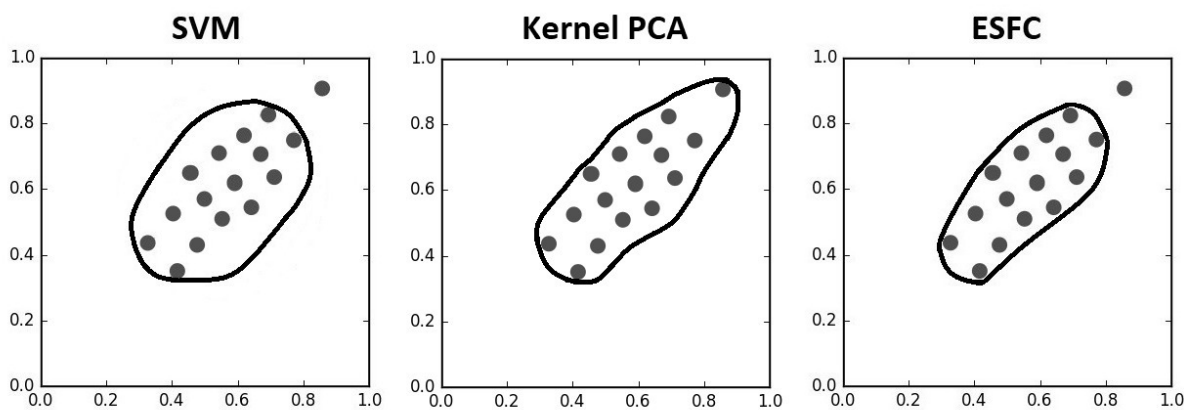


Рисунок 11 — Сравнение работы методов SVC, Kernel PCA и ESFC при использовании RBF ядра

Данные рисунки показывают, что предложенный алгоритм ESFC позволяет построить более точные контуры, описывающие рассматриваемое распределение данных, заключив его в области с наименьшей площадью по сравнению с другими рассматриваемыми методами машинного обучения. Это достигается за счет использования эллипсоидальной формы строящегося кластера (в отличие от метода SVC), центр которого пересчитывается на каждой итерации и тем самым является оптимальным (в отличие от метода Kernel PCA).

Для всех рассмотренных алгоритмов при использовании линейного ядра соответствующие контуры пороговых значений решающих функций являются гладкими и аналитически задаются уравнениями окружностей или эллипсов. При использовании RBF-ядра, решающие контуры имеют более сложную форму, поскольку они являются проекциями из пространства характеристик высокой размерности в исходное признаковое пространство. Предложенный алгоритм строит более точные границы по сравнению с методами SVC и Kernel PCA как при использовании линейной, так и при использовании RBF потенциальных функций.

### 3.3.5 Оценка сложности работы алгоритма

Вычислим сложность [92] предложенного алгоритма.

#### Шаг 0.

На данном шаге происходит вычисление  $N$  элементов  $u_i^{(0)}$ . Следовательно, *данный шаг требует  $O(N)$  операций*. Далее алгоритм сходится линейно, и его сложность есть произведение числа итераций  $L$  на сложность каждой итерации.

#### Шаг 1.



На первом этапе данного шага вычисляется общий член  $\sum_{i=1}^N \left(u_i^{(l-1)}\right)^m$ , что требует  $O(N)$  операций. Далее вычисляются  $N$  значений  $a_i^{(l)}$ , что также требует  $O(N)$  операций.

После этого происходит расчет обратной ковариационной матрицы  $M$ :

- Вначале необходимо рассчитать матрицу значений потенциальной функции  $K$ , что требует  $O(dN^2)$  операций, где  $d$  зависит от используемой потенциальной функции и размерности входных данных;
- После этого вычисляются элементы ковариационной матрицы  $C$ , что требует  $O(N^2)$  операций;
- Далее осуществляется расчет обратной к  $C$  матрицы – непосредственно матрицы  $M$ . Вычисления сводятся к LU-разложению матрицы  $C$  и дальнейшему решению системы линейных уравнений. Сложность данного этапа составляет  $O(\frac{2}{3}N^3)$  операций.

Таким образом, данный шаг требует  $O(\frac{2}{3}N^3)$  операций и наиболее трудоемкой является процедура расчета обратной матрицы.

## Шаг 2.

На первом этапе данного шага необходимо осуществить расчет расстояний  $D_k(a^{(l)})$  от центра кластера  $a^{(l)}$  до образа  $k$ -го наблюдения  $x_k$  для всех  $k = \overline{1, N}$ . Расчет общего члена  $\sum_{j=1}^N \sum_{i=1}^N M_{ij}^{(l)}$  занимает  $O(N^2)$  операций. Далее, рассчитываются  $N$  произведений общего члена на  $M_{kk}^{(l)}$ , что требует  $O(N)$  операций. Дальнейшая сортировка  $N$  рассчитанных значений  $D_k(a^{(l)})$  требует  $N \log N$  операций.

Таким образом, данный шаг требует  $O(N^2)$  операций.

## Шаг 3.

На данном шаге происходит вычисление  $N$  элементов  $u_i^{(l)}$ . Следовательно, данный шаг требует  $O(N)$  операций.

Таким образом, получили, что сложность процедуры обучения модели предложенным алгоритмом составляет  $O(\frac{2}{3}LN^3)$  операций, где  $L$  – число итераций. Отметим, что поскольку предложенный алгоритм строит модель, для ее построения можно использовать случайную выборку меньшего размера  $N_{New} < N$ , тем самым понижая сложность алгоритма.

Оценим сложность предложенного алгоритма при классификации нового тестового наблюдения  $z$ .

Расчет общего члена  $\sum_{j=1}^N \sum_{i=1}^N M_{ij}$  занимает  $O(N^2)$  операций. Далее необходимо рассчитать  $N$  значений потенциальной функции  $K(x_r, z)$ ,  $r = \overline{1, N}$ , что требует  $O(dN)$  операций, где  $d$  зависит от используемой потенциальной функции и размерности входных данных. Вычисление члена  $\frac{\sum_{r,s=1}^N u_r^m u_s^m K_{rs}}{(\sum_{r=1}^N u_r^m)^2}$  требует  $O(N^2)$  операций, вычисление члена  $\frac{\sum_{r=1}^N u_r^m K(x_r, z)}{\sum_{r=1}^N u_r^m} - O(N)$  операций.

Таким образом, получили, что сложность процедуры классификации нового тестового наблюдения предложенным алгоритмом составляет  $O(N^2)$  операций.

### **3.4 Метод подбора оптимальных значений метопараметров алгоритмов одноклассовой классификации**

Одной из основных проблем одноклассовых методов машинного обучения является подбор значений метопараметров данных алгоритмов, задаваемых априори. Поскольку в реальных ситуациях нам доступны данные только легитимного класса, а примеры целевого нелегитимного класса либо отсутствуют, либо не отмечены в обучающей выборке, обычные методы подбора и корректировки значений параметров алгоритмов классификации с использованием валидационного набора данных (holdout-набора), содержащего размеченные примеры обоих классов, как для методов обучения с учителем, использовать нельзя [16].

Для решения вышеописанной проблемы в данной диссертационной работе предлагается метод подбора оптимальных значений метопараметров алгоритмов одноклассовой классификации на основе валидационного набора данных – адаптация аналогичного подхода для задачи классификации неразмеченного набора данных (задачи обучения без учителя). Основной идеей предлагаемого метода является фиксация значения ключевого метопараметра – ожидаемого процента исключений и перебор по сетке значений всех остальных метопараметров рассматриваемого алгоритма машинного обучения таким образом, чтобы процент исключений на валидационном наборе данных совпал с ожидаемым процентом исключений для обучающей выборки. Рассмотрим данный метод более подробно.

Для каждого рассматриваемого легитимного класса все имеющиеся по нему данные предлагается разбивать случайно в отношении 50/50 на обучающую и holdout-выборку. Заметим, что одним из параметров всех рассматриваемых kernel методов поиска аномалий является метапараметр, который можно охарактеризовать как «ожидаемое отношение числа исключений к общему числу объектов в рассматриваемой выборке» (или ожидаемый процент исключений). Например, для алгоритма SVC это параметр  $\nu$  (21) – отношение ожидаемого числа исключений к общему числу объектов рассматриваемой выборки. Для методов Fuzzy и ESFC данный параметр выражается отношением  $\frac{q}{N}$ , где  $q$  – число нетипичных объектов (то есть объектов, имеющих степень принадлежности меньше 0.5),  $N$  – размер обучающей выборки (42). Для Kernel PCA этот параметр обозначает долю объектов с наибольшим уровнем аномальности, пороговое значение данной величины вычисляется на обучающей выборке (27) и используется при дальнейшей классификации. Остальными метапараметрами рассматриваемых алгоритмов являются ширина ядра, а также степень нечеткости для алгоритмов Fuzzy и ESFC, и количество главных компонент для метода Kernel PCA. Предлагается для каждого рассматриваемого метода одноклассовой классификации фиксировать значение параметра «ожидаемый процент исключений» и варьировать значениями оставшихся параметров для достижения их оптимальной комбинации, то есть комбинации, при которой значения параметра – «ожидаемого процента исключений» для обучающего и валидационного (holdout) наборов данных окажутся наиболее близкими. В частности, для метода SVC необходимо найти перебором такое значение ширины ядра, при котором на тренировочном и на валидационном наборах данных пропорции наблюдений с отрицательным значением решающей функции будут совпадать. Для методов Fuzzy и ESFC с целью достижения одинаковых на тестовом и валидационном наборах данных пропорций наблюдений со степенью типичности, меньшей 0.5, перебираются по решетке значения параметров – ширины ядра и степени нечеткости. Аналогично для алгоритма Kernel PCA. При этом, обучающий и валидационный набор данных будут оставаться неизменными. Отметим, что равенство значений ожидаемого процента исключений для обучающего и holdout наборов данных означает стабильность работы рассматриваемого метода машинного обучения при найденных данным алгоритмом значениях метапараметров. Экспериментально было получено, что оптимально рассматривать следующие значения ожидаемого процента исключений – 5%, 10%, 15%.

Заметим, что в некоторых существующих работах подбор значений метапараметров для одноклассовых методов машинного обучения осуществляется путем

искусственной генерации объектов нелегитимного класса на основе предположений о типе его распределения, однако, не всегда можно сделать верное предположение о распределении данных нелегитимного класса [36]. Предложенный метод решает данную проблему.

### **3.5 Использование *t*-статистики Уэлша для оценки аномальности поведения пользователя на основе анализа целых сессий работы за компьютером**

Рассмотренные в данной работе методы машинного обучения (SVM, Kernel PCA, Fuzzy, ESFC и т.д.) позволяют построить одноклассовую модель поведения пользователей на основе построенных векторов признаков и осуществлять классификацию для каждого поступающего в систему нового тестового вектора признаков. В результате проведенного исследования было получено, что оптимальным является построение вектора признаков для каждых 300–500 клавиатурных событий. Однако, зачастую необходимо решать задачу оценки аномальности поведения пользователей за длительный промежуток времени – например, за целую сессию работы пользователя за компьютером. С помощью классификатора мы сможем получить набор откликов для всех векторов признаков, построенных за рассматриваемый временной интервал. Необходимо на основе полученной последовательности откликов уметь получать единое число – степень аномальности поведения пользователя за продолжительный промежуток времени.

Для решения поставленной задачи в данной диссертационной работе предлагается метод оценки аномальности поведения пользователя на основе анализа целых сессий работы за компьютером с использованием *t*-статистики Уэлша.

Статистический тест Уэлша [12, 16, 98] используется для проверки гипотезы о том, что два рассматриваемых распределения имеют равные средние значения. Его отличительной особенностью является тот факт, что он применим для сравнения выборок, имеющих разные дисперсии, и даже для выборок, имеющих разный размер. Условием применимости данного метода является нормальность распределения анализируемых выборок. В рамках поставленной задачи, в качестве двух рассматриваемых распределений будут выступать результаты классификации легитимного и тестового пользователей (последовательности откликов классификатора), полученные за продолжительные промежутки времени их работы за компьютером.

Для анализа динамики работы пользователя с клавиатурой компьютера необходимо разделить выборку данных легитимного пользователя на две части:  $X_{train}$  и

$X_{holdout}$ , взять анализируемую тестовую выборку  $X_{test}$  и рассмотреть величины откликов классификатора на соответствующих выборках:  $M_{holdout}$  и  $M_{test}$ . Обучение модели должно происходить на выборке  $X_{train}$ . В качестве «типичных» данных легитимного пользователя используется выборка  $X_{holdout}$ . Проверяется гипотеза о том, что два рассматриваемых распределения аномальностей (результатов классификации легитимного ( $M_{holdout}$ ) и тестового ( $M_{test}$ ) пользователей – последовательностей откликов классификатора, полученных за продолжительные промежутки времени работы пользователей за компьютером) имеют равные средние значения. Осуществляется расчет статистики (50):

$$t = \frac{\overline{M_{holdout}} - \overline{M_{test}}}{\sqrt{\frac{s_{holdout}^2}{N_{holdout}} + \frac{s_{test}^2}{N_{test}}}}, \quad (50)$$

где  $\overline{M_{holdout}}$  – среднее значение элементов первой выборки (откликов классификатора на выборке holdout легитимного пользователя);

$\overline{M_{test}}$  – среднее значение элементов второй выборки (откликов классификатора на выборке test тестового пользователя);

$s_{holdout}^2$  – дисперсия элементов первой выборки;

$s_{test}^2$  – дисперсия элементов второй выборки;

$N_{holdout}$  – число элементов первой выборки;

$N_{test}$  – число элементов второй выборки.

Далее вводится понятие числа степеней свободы:

$\nu_{holdout} = N_{holdout} - 1$  – число степеней свободы для первой выборки;

$\nu_{test} = N_{test} - 1$  – число степеней свободы для второй выборки.

Имея значения статистики  $t$  и числа степеней свободы, по их таблице соответствия находим значение  $p - value$ . Данное значение является агрегационной характеристикой, обобщающей все отклики классификатора для данных тестового пользователя за рассматриваемый период (целую сессию) его работы за компьютером ( $M_{test}$ ). Чем выше  $p - value$ , тем выше вероятность, что перед нами находится легитимный пользователь. Чем ниже значение  $p - value$ , тем выше вероятность, что перед нами находится нелегитимный пользователь (злоумышленник).

Продemonстрируем работу алгоритма на следующем примере. Собрав holdout-выборку и посчитав среднее значение всех ее элементов, мы начинаем подавать на вход алгоритму тестовые векторы. С приходом каждого нового тестового вектора, мы обновляем величину среднего значения накопившихся к данному моменту элементов

тестовой выборки. Далее мы вычисляем  $t$ -статистику Уэлша и соответствующий ей  $p$  – *value*.

Таким образом, мы можем построить график  $p$  – *value* для легитимного и нелегитимного пользователей. Также возможно построить график  $\log(1 + p - value)$ . Здесь чем ближе график пользователя к нулю, тем выше вероятность, что он является злоумышленником.

Рассмотрим Рисунок 12:

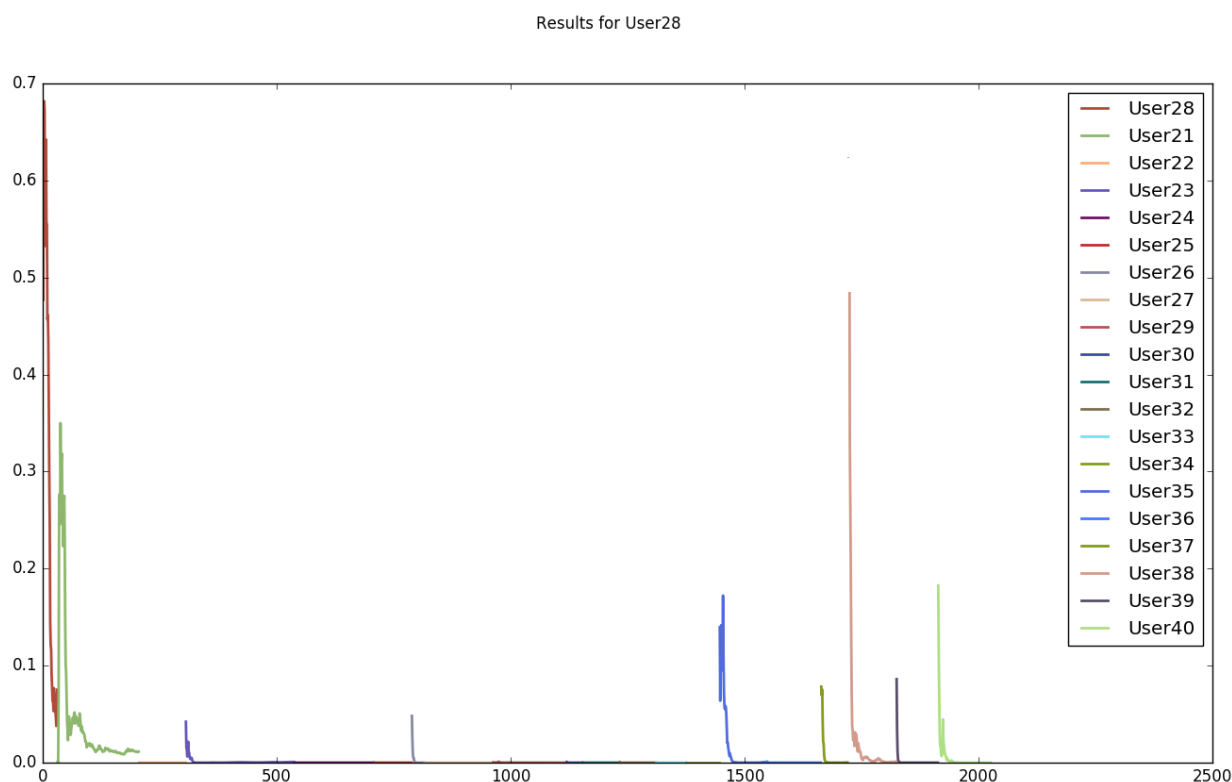


Рисунок 12 — График  $\log(1 + p - value)$ , демонстрирующий качество распознавания пользователя по динамике его работы с клавиатурой

Данный график демонстрирует качество распознавания выборочного пользователя по динамике его работы с клавиатурой. Красным цветом обозначена динамика работы искомого легитимного пользователя с клавиатурой. Другими цветами обозначена динамика работы других пользователей с клавиатурой (злоумышленников). Как мы видим, кривые графиков злоумышленников намного ниже кривых графика легитимного пользователя, что свидетельствует о высоком качестве его распознавания.

### 3.6 Экспериментальное исследование

С целью проверки качества работы предложенного нечеткого метода выявления аномалий в данных на основе эллиптической кластеризации (ESFC) в RKHS, а также предложенных методов подбора оптимальных значений метапараметров алгоритмов одноклассовой классификации и оценки аномальности целых сессий работы пользователей за компьютером, была проведена серия экспериментальных исследований как на локальном, так и на web-наборе данных, характеризующих динамику работы пользователей с клавиатурой персонального компьютера (ноутбука). Исследовалось качество работы алгоритмов при классификации как отдельных событий (отдельных векторов признаков), так и целых сессий работы пользователей за компьютером. Для оценки качества работы алгоритмов помимо рассмотренного в предыдущей главе усредненного по всем пользователям значения площади под ROC-кривой (ROC AUC) также рассматривалось медианное значение данной величины и ее разброс (межквартильное расстояние). Комбинация данных показателей позволяет более объективно оценить качество работы предложенных алгоритмов. Также было проведено сравнение работы предложенного метода машинного обучения ESFC с работой рассмотренных выше алгоритмов SVC, Fuzzy и Kernel PCA. В экспериментах использовалось признаковое пространство, сформированное в предыдущей главе данной диссертационной работы. В качестве методов постобработки признаков использовался отбор признаков на основе уровня их стабильности с использованием критерия Колмогорова-Смирнова с последующей дискретизацией по квантилям наиболее стабильных признаков. Для построения и дальнейшей обработки признаков использовались следующие оптимальные значения параметров алгоритмов, выявленные на предыдущем этапе исследований (см. Раздел 2.7):

- **Локальные данные:**
  - **Размер окна: (300 – 500) событий;**
  - **Процент перекрытия: 0.01;**
  - **Количество квантилей для дискретизации: 7;**
  - **Пауза при разбиении на временные окна: 40 секунд;**
  - **Количество рассматриваемых наиболее часто используемых одиночных клавиш: 50;**
  - **Количество рассматриваемых наиболее часто используемых диграфов: 100;**

- Использование группового признака (среднего времени удержания группы клавиш во временном окне);
- Отбор признаков, уровень стабильности которых выше 0.5
- Web-данные:
  - Размер окна: (300 – 500) событий;
  - Процент перекрытия: 0.01;
  - Количество квантилей для дискретизации: 5;
  - Пауза при разбиении на временные окна: 80 секунд;
  - Количество рассматриваемых наиболее часто используемых одиночных клавиш: 75;
  - Количество рассматриваемых наиболее часто используемых диграфов: 100;
  - Использование группового признака (среднего времени удержания группы клавиш во временном окне);
  - Отбор 150 наиболее стабильных признаков.

Оптимальные значения метапараметров рассмотренных алгоритмов одноклассовой классификации, найденные предложенным в данной работе методом, представлены в Таблицах 18–19. Рассматривались следующие значения ключевого метапараметра – ожидаемого процента исключений: 5%, 10%, 15%. Значения оставшихся метапараметров рассматриваемых алгоритмов перебирались по решетке с использованием жадного алгоритма.

Таблица 18 — Найденные с помощью предложенного алгоритма оптимальные значения метапараметров алгоритмов одноклассовой классификации (Локальные данные)

<b><u>Набор данных 2</u></b>	<b>Название параметра</b>	<b>Значение параметра</b>
SVC	Ширина ядра	0.01
	Процент исключений	10%
Kernel PCA	Ширина ядра	0.01
	Процент исключений	10%
	Количество главных компонент	45
Fuzzy	Ширина ядра	0.007
	Процент исключений	10%
	Степень нечеткости	1.5
ESFC	Ширина ядра	0.005
	Процент исключений	5%
	Степень нечеткости	1.5



Таблица 19 — Найденные с помощью предложенного алгоритма оптимальные значения метапараметров алгоритмов одноклассовой классификации (Web-данные)

<u>Набор данных 3</u>	Название параметра	Значение параметра
SVC	Ширина ядра	0.01
	Процент исключений	10%
Kernel PCA	Ширина ядра	0.01
	Процент исключений	10%
	Количество главных компонент	35
Fuzzy	Ширина ядра	0.008
	Процент исключений	5%
	Степень нечеткости	1.3
ESFC	Ширина ядра	0.007
	Процент исключений	5%
	Степень нечеткости	1.5

При использовании найденных оптимальных значений метапараметров методов машинного обучения, были получены следующие результаты классификации (см. Таблицы 20–21):

Таблица 20 — Результаты сравнения качества работы рассматриваемых методов машинного обучения (Локальные данные)

<u>Набор данных 2</u>	Классификация векторов		Классификация сессий	
	Mean ROC AUC	Median $\pm$ Interquartile Range	Mean ROC AUC	Median $\pm$ Interquartile Range
SVC	0.8698	0.8780 $\pm$ 0.1228	0.8902	0.9003 $\pm$ 0.1004
Kernel PCA	0.8721	0.8805 $\pm$ 0.1203	0.8943	0.9056 $\pm$ 0.0905
Fuzzy	0.8839	0.8949 $\pm$ 0.1150	0.9007	0.9118 $\pm$ 0.0875
ESFC	<b>0.8979</b>	<b>0.9163<math>\pm</math>0.1039</b>	<b>0.9215</b>	<b>0.9316<math>\pm</math>0.0684</b>

Таблица 21 — Результаты сравнения качества работы рассматриваемых методов машинного обучения (Web-данные)

<u>Набор данных 3</u>	Классификация векторов		Классификация сессий	
	Mean ROC AUC	Median $\pm$ Interquartile Range	Mean ROC AUC	Median $\pm$ Interquartile Range
SVC	0.9125	0.9213 $\pm$ 0.0932	0.9317	0.9415 $\pm$ 0.0584
Kernel PCA	0.9153	0.9245 $\pm$ 0.0821	0.9357	0.9445 $\pm$ 0.0556
Fuzzy	0.9234	0.9365 $\pm$ 0.0745	0.9506	0.9527 $\pm$ 0.0474
ESFC	<b>0.9427</b>	<b>0.9516<math>\pm</math>0.0539</b>	<b>0.9678</b>	<b>0.9719<math>\pm</math>0.0281</b>

На основании полученных результатов можно сделать следующие выводы:

- Предложенный нечеткий метод выявления аномалий в данных на основе эллиптической кластеризации в RKHS показывает наилучшее качество работы по сравнению со всеми остальными рассматриваемыми алгоритмами (SVC, Kernel PCA, Fuzzy) как на локальном, так и на web-наборе данных, характеризующих клавиатурный почерк пользователей;
- С помощью предложенного метода подбора оптимальных значений метапараметров алгоритмов одноклассовой классификации удается достичь высокого качества распознавания пользователей по динамике их работы с клавиатурой персонального компьютера (ноутбука);
- С помощью предложенного метода оценки аномальности поведения пользователя на основе анализа целых сессий работы за компьютером удается достичь достаточно высокого качества распознавания: медианное значение показателя ROC AUC достигает порядка 0.93–0.97;
- В среднем прирост значения ROC AUC на данном этапе исследований составляет порядка 3% по сравнению с результатами, полученными на предыдущем этапе (см. Раздел 2).

### **3.7 Выводы**

В данном разделе проводились исследование и разработка методов построения одноклассовой модели пользователя по динамике его работы с клавиатурой персонального компьютера (ноутбука), а также разработка методов подбора оптимальных значений метапараметров алгоритмов одноклассовой классификации и оценки аномальности целых сессий работы пользователей за компьютером. В результате проведенного исследования были достигнуты следующие результаты:

- Был разработан нечеткий метод выявления аномалий в данных на основе эллиптической кластеризации (ESFC) в RKHS, показавший наилучшие результаты по сравнению с методами SVC, Kernel PCA и Fuzzy в задаче динамической аутентификации пользователей на основе анализа их клавиатурного почерка. Данный метод сочетает в себе достоинства методов SVC, Fuzzy и Kernel PCA и решает их основные проблемы. Ключевой особенностью предложенного метода является построение эллипсоидального (в отличие от методов SVC и Fuzzy, работающих с гиперсферой и сферическим кластером соответственно) кластера в RKHS, включающего в себя все образы объектов рассматриваемой выборки и

задающего степень принадлежности (типичности) для каждого рассматриваемого образа. За счет использования эллиптической формы кластера удастся более точно описать основную часть легитимной обучающей выборки и построить более точную одноклассовую модель соответственно. Центр данного кластера пересчитывается на каждой итерации (в отличие от метода Kernel PCA) и тем самым является оптимальным. Для вычисления расстояния между объектами и центром нечеткого кластера используется метрика Махаланобиса, учитывающая разброс дисперсии и корреляции между признаками в пространстве высокой размерности. Качество работы данного метода в среднем на 3% превышает качество распознавания методов SVC и Kernel PCA и на 1.5–2% превышает качество работы метода Fuzzy. Дополнительно была доказана сходимость данного метода, а также проведена оценка сложности его работы.

- Был предложен метод подбора оптимальных значений метапараметров алгоритмов одноклассовой классификации, позволяющий строить стабильные к смене тестового набора данных модели без использования информации о данных нелегитимного класса. С использованием данного метода удалось достичь высокого качества распознавания пользователей по динамике их работы с клавиатурой компьютера.
- Разработан метод оценки аномальности поведения пользователей на основе анализа целых сессий работы за компьютером с использованием t-статистики Уэлша, обобщающий все отклики классификатора для данных тестового пользователя за рассматриваемый период (целую сессию) его работы за компьютером и позволяющий достичь высокого качества аутентификации.
- Была проведена серия экспериментов, в результате которой было подтверждено высокое качество работы предложенных алгоритмов: удалось достичь качества распознавания порядка 0.9–0.95 ROC AUC (в случае распознавания отдельных векторов признаков) и 0.93–0.97 ROC AUC (в случае анализа целых сессий работы за компьютером) как на локальных, так и на web-данных клавиатурного почерка пользователей, что значительно превосходит качество аутентификации пользователей при использовании методов, рассмотренных в существующих научных работах.

## 4 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ЭКСПЕРИМЕНТАЛЬНОГО ОБРАЗЦА ПРОГРАММНОГО КОМПЛЕКСА

В предыдущих разделах данной диссертационной работы было сформировано стабильное по времени признаковое пространство, а также разработана комбинация алгоритмов обработки признаков и построения модели пользователя, состоящая из отбора признаков по уровню их стабильности на основе критерия Колмогорова-Смирнова, дальнейшей дискретизации по квантилям наиболее стабильных признаков, построения модели поведения пользователя нечетким методом выявления аномалий в данных на основе эллиптической кластеризации в RKHS с подбором оптимальных значений метопараметров данного метода собственно разработанным алгоритмом, совместное использование которых позволило достичь высокого качества распознавания пользователей на основе анализа их клавиатурного почерка. Оценка аномальности поведения пользователя может производиться как за короткий, так и за продолжительный период его работы – с использованием разработанного метода на основе анализа целых сессий работы за компьютером с использованием t-статистики Уэлша.

Данный раздел посвящен задачам разработки и реализации экспериментального образца программного комплекса динамической аутентификации пользователей по динамике их работы с клавиатурой персонального компьютера (ноутбука), в основе работы которого лежат предложенные в данной работе алгоритмы.

Решение поставленных на данном этапе задач состоит из следующих шагов:

- разработка сценариев функционирования экспериментального образца программного комплекса (ЭО ПК);
- разработка архитектуры ЭО ПК;
- разработка структуры представления пользовательской поведенческой информации, хранимой и обрабатываемой в ЭО ПК;
- разработка программных модулей сбора, хранения пользовательской поведенческой информации, построения моделей поведения пользователей и их применения для проведения динамической аутентификации пользователей на основе анализа их клавиатурного почерка.

## **4.1 Сценарии функционирования**

Разрабатываемый ЭО ПК должен поддерживать три базовых сценария функционирования [99]:

- Сбор поведенческой информации о взаимодействии пользователей с клавиатурой компьютера;
- Построение индивидуальных моделей поведения пользователей;
- Применение индивидуальных моделей поведения пользователей.

### **4.1.1 Сбор поведенческой информации о взаимодействии пользователей с клавиатурой компьютера**

Задачей данного этапа являются сбор и сохранение данных, описывающих взаимодействие пользователей с клавиатурой компьютера. Подразумевается, что существует некоторое количество рабочих мест пользователей (персональных компьютеров или ноутбуков), на каждом из которых с помощью специализированных агентов мониторинга собирается пользовательская поведенческая информация, сохраняется локально на рабочем месте пользователя и с помощью агента консолидации передается в единое сетевое хранилище данных (на сервер консолидации) для дальнейшего анализа.

В качестве пользовательской поведенческой информации в данной задаче выступает набор записей о работе пользователей с клавиатурой компьютера следующего вида:

- Имя пользователя;
- Код используемой клавиши;
- Тип события (нажатие / отжатие);
- Временная метка, соответствующая произошедшему событию.

Также, для дополнительного анализа может использоваться имя процесса, в рамках которого производится взаимодействие пользователя с клавиатурой компьютера. Однако, как показала предварительная серия экспериментов, использование данной характеристики не позволяет улучшить качество распознавания пользователей.

Собирать информацию предлагается локально в фоновом режиме с использованием специального hook-a (перехватчика), осуществляющего перехват пришедших в систему событий от клавиатуры (посредством WinAPI) и сохраняющего полученную информацию локально на рабочем месте пользователя.

Далее осуществляется передача собранной информации в единое сетевое хранилище для последующей обработки. Вариант непосредственной записи данных в общее сетевое хранилище является наиболее перспективным в данной задаче по сравнению со стратегиями ручного копирования данных либо же использования клиент-серверной архитектуры, поскольку требует минимального количества трудозатрат и обладает высокой эффективностью. Поддержка промежуточного локального хранения собранной информации позволяет оптимизировать нагрузку на сеть передачи данных, а также позволяет не потерять собранные данные в случае отсутствия соединения с сервером консолидации.

Поскольку ранее было установлено, что в собранных данных могут присутствовать некоторые шумовые события (события, полученные при работе пользователей с клавиатурными тренажерами, а также возможные непарные события нажатия / отжатия клавиш либо же ситуации длительного удерживания клавиш), собранные в едином сетевом хранилище данные необходимо отфильтровывать агентом консолидации перед процедурой построения индивидуальных поведенческих моделей.

Таким образом, решение задачи состоит из следующих этапов [99]:

- Сбор и сохранение поведенческой информации на рабочих местах пользователей;
- Передача собранной информации в единое сетевое хранилище (сервер консолидации), сохранение полученной информации на сервере консолидации;
- Фильтрация полученной информации (удаление шумовых событий) агентом консолидации.

#### **4.1.2 Построение индивидуальных моделей поведения пользователей**

В рамках настоящей работы было сформировано признаковое пространство, с высокой точностью описывающее клавиатурный почерк пользователей, а также разработаны:

- Подход к сокращению размерности признакового пространства путем отбора наиболее значимых признаков на основе уровня их стабильности с использованием критерия Колмогорова-Смирнова;
- Подход к обработке наиболее стабильных признаков на основе дискретизации их по квантилям;

- Нечеткий метод выявления аномалий в данных на основе эллиптической кластеризации в RKHS;
- Метод подбора оптимальных значений метапараметров алгоритмов одноклассовой классификации без использования знаний о специфике данных нелегитимного класса.

Необходимо на основе собранной информации о взаимодействии пользователей с клавиатурой компьютера реализовать процесс построения соответствующих индивидуальных моделей поведения пользователей. Решение данной задачи состоит из следующих этапов [99]:

- Формирование обучающей выборки путем возможной фильтрации собранных данных, лежащих в едином хранилище (по времени и т.д.);
- Построение признакового пространства на основе данных из сформированной обучающей выборки, применение разработанных методов обработки признаков и построения модели пользователя – по завершении данного этапа мы получим сформированную индивидуальную поведенческую модель;
- Сохранение построенной индивидуальной поведенческой модели в хранилище моделей с целью дальнейшего применения к тестовым данным клавиатурного почерка пользователей.

Необходимо отметить, что поскольку было установлено, что в данной задаче индивидуальные поведенческие модели имеют свойство постепенно устаревать, необходимо предусмотреть поддержку возможности их периодического обновления.

#### **4.1.3 Применение индивидуальных моделей поведения пользователей**

В результате применения построенных индивидуальных поведенческих моделей к тестовым данным клавиатурного почерка неизвестных пользователей мы получим некоторое число – степень аномальности тестовых данных. На основе данного числа делается вывод – является ли рассматриваемый пользователь, данные которого мы проверяем, легитимным. Оценивать степень аномальности предлагается как для отдельных тестируемых векторов признаков, так и для целой сессии работы тестируемого пользователя за компьютером.

Применять построенные поведенческие модели можно как к тестовым данным, собранным ранее (в отложенном режиме), так и к тестовым данным, собираемым в

данный момент (в режиме, близком к режиму реального времени). При использовании отложенного режима применения модели, данные для классификации берутся из единого сетевого хранилища. Данный режим может использоваться аналитиком для глубокого анализа поведения пользователей, сравнения работы различных алгоритмов и т.д. и не требует скорейшего реагирования системы на действия злоумышленника. Также применение поведенческих моделей в отложенном режиме может задаваться расписанием. Однако, если мы работаем в режиме, близком к режиму реального времени, необходимо осуществлять скорейшее применение модели. В виду того, что на пересылку данных в единое сетевое хранилище требуется время, и не всегда может быть установлено соединение с единым хранилищем, в данном режиме осуществлять классификацию тестовых данных необходимо локально на рабочем месте пользователя.

Соответственно, если мы используем тестовые данные, собранные ранее, то решение задачи состоит из следующих этапов [99]:

- Формирование тестовой выборки путем возможной фильтрации собранных данных, лежащих в едином хранилище (по времени и т.д.);
- Выбор необходимой индивидуальной поведенческой модели;
- Применение выбранной индивидуальной поведенческой модели к сформированной тестовой выборке: построение векторов признаков, классификация данных, анализ полученных результатов (включая блокировку системы в случае обнаружения злоумышленника).

В случае использования тестовых данных, собираемых в режиме, близком к режиму реального времени, решение поставленной задачи будет состоять из следующих этапов [99]:

- Выбор необходимой индивидуальной поведенческой модели;
- Применение выбранной индивидуальной поведенческой модели к сформированной тестовой выборке: построение векторов признаков, классификация данных, анализ полученных результатов (включая блокировку системы в случае обнаружения злоумышленника);
- Локальное сохранение результатов аутентификации пользователей и их передача агенту консолидации.

Отметим, что разрабатываемая система должна с высоким качеством аутентифицировать пользователя в независимости от того, пересаживался ли рассматриваемый пользователь с одного компьютера на другой (происходила ли смена используемого оборудования или нет).



## **4.2 Программная реализация**

Рассмотрим предлагаемую программную реализацию экспериментального образца программного комплекса. Требованиями к разрабатываемой системе являются функционирование на ОС Windows версии 7 и выше, а также оперативность работы, эффективное использование ресурсов и устойчивость к возможным ошибкам.

### **4.2.1 Архитектура системы**

Предлагаемое архитектурное решение представляет собой мультиагентную систему [7, 9, 11, 99], состоящую из следующих модулей:

1. Агент мониторинга. Данный агент устанавливается непосредственно на рабочее место пользователя и состоит из двух параллельно работающих модулей: модуля сбора информации и модуля классификации. Модуль сбора информации в фоновом режиме осуществляет сбор информации о взаимодействии пользователя с клавиатурой компьютера, сохраняет ее локально и периодически (в соответствии с заданным режимом передачи) осуществляет передачу сохраненной информации агенту консолидации. Модуль классификации работает на рабочем месте пользователя и осуществляет применение индивидуальной поведенческой модели пользователя, полученной из хранилища моделей, в режиме, близком к режиму реального времени, к данным клавиатурного почерка, собранным локально на данном рабочем месте. В случае обнаружения аномального поведения пользователя, доступ к системе блокируется. Полученные результаты классификации также передаются агенту консолидации.
2. Агент консолидации. Данный агент устанавливается на сервере консолидации и включает в себя модуль консолидации. Модуль консолидации осуществляет прием данных клавиатурного почерка и локальной классификации пользователей от агентов мониторинга, обеспечивает их хранение в едином сетевом хранилище (на сервере консолидации), фильтрацию собранной информации и предоставляет доступ к хранящимся данным для их дальнейшего анализа и построения индивидуальных моделей поведения пользователей. Также данный модуль осуществляет передачу копий построенных поведенческих моделей из хранилища моделей на рабочие места соответствующих пользователей с целью ускорения процедуры аутентификации, а также во избежание проблем, связанных с возможным

обрывом соединения между рабочими местами пользователей и сервером консолидации.

3. Модуль построения индивидуальных моделей поведения пользователей. Данный модуль осуществляет построение моделей поведения пользователей на основе данных, хранящихся в едином сетевом хранилище, и сохраняет построенные модели в хранилище моделей (находящемся на сервере консолидации).
4. Модуль анализа клавиатурного почерка пользователей в отложенном режиме. Данный модуль осуществляет применение построенных моделей поведения пользователей, полученных из хранилища моделей, к данным клавиатурного почерка пользователей, хранящихся в едином сетевом хранилище.
5. Автоматизированное рабочее место (АРМ) аналитика. АРМ аналитика представляет собой графический интерфейс, позволяющий управлять модулем консолидации, строить и перестраивать (обновлять) индивидуальные модели поведения пользователей (с использованием модуля построения индивидуальных моделей поведения пользователей), применять построенные модели в отложенном режиме (с использованием модуля анализа клавиатурного почерка пользователей в отложенном режиме), наблюдать за результатами применения моделей на рабочих местах пользователей в режиме, близком к режиму реального времени (получая данную информацию из единого сетевого хранилища), оценивать степень аномальности поведения пользователя на основе анализа целых сессий работы за компьютером, а также строить подробные отчеты о работе системы.

Общая схема архитектуры предлагаемого ЭО ПК представлена на Рисунке 13.



Рисунок 13 — Архитектура разработанного ЭО ПК

Каждый из рассмотренных модулей состоит из компонентов, выполняющих определённые действия над исходными или промежуточными данными. Компонент – это программный подмодуль определённого типа, выполнение которого определяется конфигурационным файлом. Компоненты делятся на два основных вида: задания и процессоры, и могут состоять из нескольких подкомпонентов. Взаимодействие между компонентами осуществляется посредством текстовых и конфигурационных файлов. Текстовые файлы содержат входные для компонентов данные или результаты работы компонентов. Конфигурационные файлы содержат настройки алгоритмов, реализованных в компонентах. Соответствие между программными модулями и входящими в них программными компонентами приведено в Таблице 22.

Таблица 22 — Соответствие между программными модулями и входящими в них программными компонентами

Программный модуль	Программные компоненты, входящие в его состав
Модуль сбора информации агента мониторинга	Программный компонент сбора данных клавиатурного почерка пользователей
Модуль классификации агента мониторинга	Программный компонент обработки данных клавиатурного почерка пользователей
	Программный компонент построения

Программный модуль	Программные компоненты, входящие в его состав
	векторов признаков пользователя
	Программный компонент обработки векторов признаков пользователя
	Программный компонент классификации новых данных
Модуль консолидации агента консолидации	Программный компонент обработки данных клавиатурного почерка пользователей
Модуль анализа клавиатурного почерка пользователей в отложенном режиме	Программный компонент построения векторов признаков пользователя
	Программный компонент обработки векторов признаков пользователя
	Программный компонент классификации новых данных
Модуль построения индивидуальных моделей поведения пользователей	Программный компонент построения структуры векторов признаков
	Программный компонент построения векторов признаков пользователя
	Программный компонент обработки векторов признаков пользователя
	Программный компонент построения модели пользователя
АРМ аналитика	Программный компонент визуализации
	Программный компонент оценки аномальности поведения пользователя на основе анализа целых сессий работы за компьютером

Задание – это вид компонента, задачами которого являются: чтение необходимых исходных данных; обработка и возможная фильтрация прочитанных данных; вызов алгоритма обработки данных (процессора) и запись выходных данных в результирующий файл [12, 13].

Процессор – это вид компонента, основной задачей которого является выполнение алгоритма обработки данных, включая обработку векторов признаков, построение модели, классификацию и т.д. Параметры используемого процессором алгоритма задаются при создании процессора и сохраняются в конфигурационном файле.

Последовательность заданий может выполняться как в автоматическом режиме, так и поэтапно. Поэтапный режим выполнения последовательностей заданий используется на АРМ аналитика, производящего глубокое исследование процессов построения и применения индивидуальных моделей пользователей.

Разбиение всего процесса аутентификации на последовательность заданий позволяет откатиться на шаг назад при появлении возможных ошибок (при этом, анализируемые данные и промежуточные результаты работы не потеряются). Тем самым, повышается устойчивость разработанной системы к возможным ошибкам. Дополнительно отметим, что разрабатываемый ЭО ПК поддерживает многозадачность: одновременно могут выполняться задания для осуществления разных этапов аутентификации нескольких пользователей (например, в модуле построения индивидуальных моделей поведения пользователей может осуществляться одновременное построение векторов признаков и моделей для разных профилей, а не только для одного), что также повышает оперативность работы предлагаемой системы, а также позволяет эффективнее использовать имеющиеся ресурсы.

Заметим, что разработанная архитектура поддерживает аутентификацию пользователя при смене используемого оборудования.

#### **4.2.2 Описание программных компонентов**

##### **4.2.2.1 Программный компонент сбора данных клавиатурного почерка пользователей**

Данный программный компонент входит в состав модуля сбора информации агента мониторинга и в фоновом режиме осуществляет сбор поведенческой информации о динамике работы пользователей с клавиатурой компьютера.

Программный компонент состоит из двух файлов: динамической библиотеки-перехватчика (.dll), предназначенной для встраивания в процессы Windows для перехвата целевых событий, и исполняемого файла-инжектора (.exe), предназначенного для управления данной библиотекой. Подключение перехватчика локальных событий от клавиатуры осуществляется средствами WinApi (SetWindowsHookEx / UnhookWindowsHookEx). Перехватчик и инжектор используют параллельную обработку и

взаимодействуют с помощью нитей (threads) и стандартных средств синхронизации Windows (критические секции, разделяемая память, мьютексы) [9, 11]. При запуске приложения в целевом каталоге создаются (в случае отсутствия) логи программы, которые могут использоваться в дальнейшем в информационно-отладочных целях.

Для каждой сессии работы пользователя за компьютером создается директория, содержащая в своем названии имя пользователя, дату и время начала сессии [9, 11]. Данная директория включает в себя .csv-файл, содержащий записанные данные о динамике работы пользователей с клавиатурой компьютера в порядке их прихода в систему, а также файл в формате JSON, содержащий информацию об аппаратно-программной конфигурации клиентской машины, соответствующей времени начала сессии (базовые сведения об используемых аппаратуре, IP-адресе рабочей машины, ОС, версии программы). Примеры содержимого данных файлов приведены на Рисунках 14–15.

time	username	session_id	hwnd_ori	processna	PID	hookmodi	provocati	virtualcod	scancode	keyup	prev_keys	repeat-co	alt_down	extended	langID	keybdID	raw_data_debug
2016-04-21afilippov	0x1	0x100cc	FlashMon		2916	0	0	13	28	1	1	1	0	0	0x409	0x409	0xc01c0001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	67	46	0	0	1	0	0	0x409	0x409	0x2e0001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	67	46	1	1	1	0	0	0x409	0x409	0xc02e0001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	72	35	0	0	1	0	0	0x409	0x409	0x230001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	72	35	1	1	1	0	0	0x409	0x409	0xc0230001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	8	14	0	0	1	0	0	0x409	0x409	0xe0001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	8	14	1	1	1	0	0	0x409	0x409	0xc00e0001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	8	14	0	0	1	0	0	0x409	0x409	0xe0001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	8	14	1	1	1	0	0	0x409	0x409	0xc00e0001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	89	21	0	0	1	0	0	0x409	0x409	0x150001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	89	21	1	1	1	0	0	0x409	0x409	0xc0150001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	79	24	0	0	1	0	0	0x409	0x409	0x180001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	85	22	0	0	1	0	0	0x409	0x409	0x160001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	79	24	1	1	1	0	0	0x409	0x409	0xc0180001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	85	22	1	1	1	0	0	0x409	0x409	0xc0160001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	84	20	0	0	1	0	0	0x409	0x409	0x140001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	84	20	1	1	1	0	0	0x409	0x409	0xc0140001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	85	22	0	0	1	0	0	0x409	0x409	0x160001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	85	22	1	1	1	0	0	0x409	0x409	0xc0160001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	66	48	0	0	1	0	0	0x409	0x409	0x300001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	66	48	1	1	1	0	0	0x409	0x409	0xc0300001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	69	18	0	0	1	0	0	0x409	0x409	0x120001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	69	18	1	1	1	0	0	0x409	0x409	0xc0120001
2016-04-21afilippov	0x1	0x10196	chrome.e		3220	0	0	40	80	0	0	1	0	1	0x409	0x409	0x1500001

Рисунок 14 — Содержимое .csv-файла, в который записывается собранная информация о взаимодействии пользователя с клавиатурой компьютера

Файл, содержащий запись активности пользователя при работе с клавиатурой (.csv), содержит следующие основные поля:

- Время прихода события в систему;
- Имя (идентификатор) пользователя;
- Идентификатор текущей сессии работы за компьютером;
- Имя процесса, с которым ведется работа;
- PID процесса, с которым ведется работа;
- Код рассматриваемой (нажатой / отжатой) клавиши;
- Метка: нажатие или отжатие клавиши произошло;
- Текущий язык раскладки клавиатуры.

```
{
  "Version": "3.2.4.0",
  "Bitness": 32,
  "Options": {
    "provocationMode": 0,
    "restartTimer": 0,
    "internalSessionID": 0,
    "timeUTC": 0
  },
  "OS": "Windows Server: 6.1.7601 Service Pack 1",
  "ComputerName": "JINN",
  "IPAddress": "158.250.19.10",
  "UserID": "kazachuk",
  "StartTime": "2017-09-26T11:28:42.441",
  "SessionID": 1,
  "Processor": {
    "frequencyMhz": 2999,
    "architectureType": 9,
    "oemID": 9,
    "count": 2,
    "type": 8664,
    "activemask": 3
  },
  "Display": {
    "width": 1280,
    "height": 1024,
    "hmmSize": 452,
    "vmmSize": 361,
    "numMonitors": 1
  },
  "Keyboard": {
    "type": 7,
    "subType": 0,
    "functionKeysType": 12,
    "repeatDelay": 1,
    "repeatSpeed": 31
  },
  "InputDevices": {
    "Keyboard0": "acpi#pnp0303#4&226211b3&0#{884b96c3-56ef-11d1-bc8c-00a0c91405dd}",
    "Keyboard1": "root#rdp_kbd#0000#{884b96c3-56ef-11d1-bc8c-00a0c91405dd}",
  }
}
```

Рисунок 15 — Содержимое JSON-файла, хранящего информацию об аппаратно-программной конфигурации клиентской машины

Данный программный компонент реализован на ЯП C++.

#### 4.2.2.2 Программный компонент обработки данных клавиатурного почерка пользователей

Данный программный компонент входит в состав модуля консолидации агента консолидации, а также модуля классификации агента мониторинга. Его основными заданиями являются:

- Задание объединения данных (Data Task), представляющее собой объединение исходных данных для конкретного пользователя из различных сессий (полученных из соответствующих .csv-файлов) и дальнейшую фильтрацию данных, не попадающих в рассматриваемый временной интервал (данные за который необходимо проанализировать), задающийся в конфигурационном файле данного задания;

- Задание обработки объединенных данных (Data Process Task). Основной задачей задания обработки объединенных данных является чтение результатов выполнения задания объединения данных, вызов процессора обработки данных (Data Processor) и запись обработанных данных в результирующий .csv-файл.

Процессор обработки объединенных данных выполняет фильтрацию данных: убирает непарные события нажатия/отжатия клавиш, продолжительную серию событий удерживания клавиш, а также события, соответствующие работе пользователя с клавиатурными тренажерами.

Данный программный компонент реализован на ЯП Python 3.5 совместно с модулями numpy, pandas, user-agents, statsmodels, позволяющими существенно упростить код программы и ускорить ее работу за счет встроенных методов распараллеливания.

#### 4.2.2.3 Программный компонент построения структуры векторов признаков

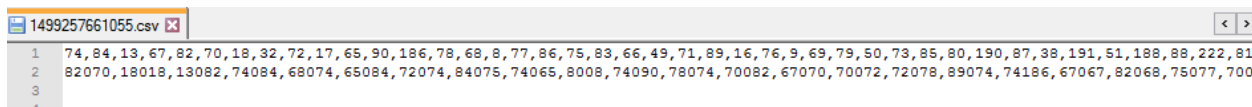
Данный программный компонент входит в состав модуля построения индивидуальных моделей поведения пользователей и состоит из задания построения структуры векторов признаков (Structure Task). Основной задачей данного задания является чтение исходных обработанных данных, вызов процессора построения структуры векторов признаков (Structure Processor), а также запись результирующего файла структуры в соответствующий .csv-файл. Процессор построения структуры векторов признаков выявляет наиболее часто используемые пользователем одиночные клавиши клавиатуры и комбинации клавиш. Количество отбираемых клавиш задаётся при создании компонента и сохраняется в конфигурационном файле.

Результирующий .csv-файл имеет следующую структуру (см. Рисунок 16):

- В первой строке упорядочены по частоте встречаемости коды наиболее часто встречаемых при работе рассматриваемого пользователя клавиш (параметр – их максимальное число указывается при создании процессора построения структуры признакового пространства). На первом месте стоит код наиболее часто встречаемой клавиши.
- Во второй строке упорядочены по частоте встречаемости коды наиболее часто встречаемых при работе рассматриваемого пользователя диграфов (параметр – их максимальное число указывается при создании процессора построения структуры признакового пространства). На первом месте стоит код наиболее часто встречаемого диграфа. Код каждого диграфа



представляет собой разделенную нулем («0») пару кодов двух клавиш, составляющих данный диграф. На первом месте в данной паре стоит первая клавиша данного диграфа.



1	74, 84, 13, 67, 82, 70, 18, 32, 72, 17, 65, 90, 186, 78, 68, 8, 77, 86, 75, 83, 66, 49, 71, 89, 16, 76, 9, 69, 79, 50, 73, 85, 80, 190, 87, 38, 191, 51, 188, 88, 222, 81
2	82070, 18018, 13082, 74084, 68074, 65084, 72074, 84075, 74065, 8008, 74090, 78074, 70082, 67070, 70072, 72078, 89074, 74186, 67067, 82068, 75077, 700

Рисунок 16 — Построенная структура признакового пространства

Данный программный компонент реализован на ЯП Python 3.5 совместно с модулями numpy, pandas, user-agents, statsmodels, позволяющими существенно упростить код программы и ускорить ее работу за счет встроенных методов распараллеливания.

#### 4.2.2.4 Программный компонент построения векторов признаков пользователя

Данный программный компонент входит в состав модуля построения индивидуальных моделей поведения пользователей, а также модуля классификации агента мониторинга и модуля анализа клавиатурного почерка пользователей в отложенном режиме, и состоит из задания построения векторов признаков (Features Task). Основной задачей данного задания является чтение исходных обработанных данных, файла построенной структуры векторов признаков, вызов процессора построения векторов признаков пользователя (Features Extractor) и запись построенных векторов признаков в результирующий .csv-файл. Процессор построения векторов признаков выполняет разбиение входных данных на временные окна и расчет признаков для каждого временного окна. Параметрами данного процессора являются настройки разбиения исходных обработанных данных на временные окна. Поддерживаются возможности расчета перекрывающихся между собой временных окон, а также учета пауз в действиях пользователя и учета смены активных процессов при осуществлении разбиения на временные окна. Разбиение на временные окна может производиться как при превышении лимита на количество событий в окне, так и при превышении лимита на временную продолжительность окна. Для каждого временного окна осуществляется расчет следующих характеристических признаков:

- Среднее время удержания клавиши во временном окне;
- Среднее время между отпусканием первой и отпусканием второй клавиши во временном окне;
- Среднее время между нажатием первой и отпусканием второй клавиши во временном окне;

- Среднее время удержания группы клавиш во временном окне;
- Частота набора текста пользователем во временном окне.

Содержимое файла рассчитанных векторов признаков одной из сессий работы пользователя с клавиатурой выглядит следующим образом (см. Рисунок 17):

```
2016-04-22 08:55:43.860000,2016-04-22 08:57:52.167000,500,"[147.0]",0,"[120.0, 87.0, 84.0, 70.0, 80.0
2016-04-22 08:56:27.434000,2016-04-22 08:57:57.543000,324,0,0,"[84.0, 414.0, 109.0, 131.0, 178.0, 123
2016-04-22 08:59:26.732000,2016-04-22 09:01:22.802000,338,0,0,"[124.0, 106.0, 109.0, 110.0, 95.0, 87.
2016-04-22 10:26:42.667000,2016-04-22 10:28:26.706000,500,0,0,"[88.0, 91.0, 80.0, 77.0, 133.0, 98.0,
2016-04-22 10:27:24.871000,2016-04-22 10:28:41.362000,386,0,0,"[128.0, 73.0, 113.0, 102.0, 77.0, 82.0
2016-04-22 10:30:47.085000,2016-04-22 10:32:30.540000,500,0,0,"[102.0, 84.0, 91.0, 117.0, 87.0, 153.0
2016-04-22 10:31:41.760000,2016-04-22 10:33:30.230000,500,0,0,"[98.0, 98.0, 84.0, 106.0, 66.0, 91.0,
2016-04-22 10:32:30.547000,2016-04-22 10:34:14.930000,500,0,0,"[87.0, 84.0, 105.0, 164.0, 87.0, 98.0,
2016-04-22 10:33:30.292000,2016-04-22 10:34:52.578000,324,0,0,"[58.0, 59.0, 87.0, 130.0, 131.0, 109.0
2016-04-22 10:50:42.591000,2016-04-22 10:53:15.231000,500,"[197.0]",0,"[120.0, 95.0, 106.0, 31.0, 112
2016-04-22 10:52:20.923000,2016-04-22 10:53:53.456000,500,0,0,"[91.0, 102.0, 109.0, 141.0, 91.0, 87.0
2016-04-22 10:53:15.309000,2016-04-22 10:54:32.690000,500,0,0,"[109.0, 91.0, 87.0, 91.0, 76.0, 99.0,
2016-04-22 10:53:54.311000,2016-04-22 10:55:12.815000,500,0,0,"[99.0, 99.0, 105.0, 91.0, 73.0, 117.0,
2016-04-22 10:54:33.480000,2016-04-22 10:55:44.610000,500,0,0,"[109.0, 91.0, 91.0, 109.0, 108.0, 83.0
2016-04-22 10:55:13.026000,2016-04-22 10:56:08.212000,322,0,0,"[94.0, 99.0, 113.0, 87.0, 16.0, 83.0,
2016-04-22 11:28:10.162000,2016-04-22 11:30:13.968000,500,0,0,"[94.0, 215.0, 110.0, 102.0, 95.0, 89.0
2016-04-22 11:29:10.854000,2016-04-22 11:30:39.097000,500,0,0,"[89.0, 98.0, 105.0, 102.0, 102.0, 91.0
2016-04-22 11:30:14.056000,2016-04-22 11:31:07.149000,500,0,0,"[77.0, 99.0, 98.0, 95.0, 84.0, 98.0, 9
2016-04-22 11:30:39.254000,2016-04-22 11:31:44.681000,500,0,0,"[87.0, 97.0, 102.0, 80.0, 80.0, 69.0,
2016-04-22 11:31:07.265000,2016-04-22 11:32:30.397000,500,0,0,"[80.0, 84.0, 107.0, 91.0, 88.0, 111.0,
2016-04-22 11:31:44.772000,2016-04-22 11:33:13.990000,500,0,0,"[88.0, 111.0, 134.0, 120.0, 91.0, 100.
2016-04-22 11:57:45.490000,2016-04-22 12:00:37.618000,300,0,0,"[184.0, 98.0, 156.0, 109.0, 75.0, 28.0
2016-04-22 12:47:45.743000,2016-04-22 12:51:39.649000,500,0,0,"[83.0, 72.0, 94.0, 133.0, 101.0, 95.0,
2016-04-22 12:49:21.331000,2016-04-22 12:54:31.648000,500,0,0,"[98.0, 17.0, 95.0, 95.0, 87.0, 110.0,
2016-04-22 12:51:39.740000,2016-04-22 12:55:00.205000,500,0,0,"[84.0, 98.0, 80.0, 91.0, 24.0, 24.0, 1
2016-04-22 12:54:31.710000,2016-04-22 12:57:09.503000,500,0,0,"[91.0, 92.0, 97.0, 102.0, 106.0, 102.0
2016-04-22 12:55:00.208000,2016-04-22 12:58:26.271000,500,0,0,"[87.0, 87.0, 70.0, 58.0, 126.0, 98.0,
2016-04-22 12:57:09.612000,2016-04-22 12:59:22.514000,500,0,0,"[126.0, 98.0, 80.0, 84.0, 73.0, 91.0,
2016-04-22 12:58:26.373000,2016-04-22 12:59:55.864000,500,0,0,"[98.0, 105.0, 84.0, 93.0, 102.0, 104.0
2016-04-22 12:59:22.543000,2016-04-22 13:00:34.186000,500,0,0,"[104.0, 84.0, 95.0, 134.0, 81.0, 81.0,
2016-04-22 12:59:55.884000,2016-04-22 13:00:56.806000,420,0,0,"[87.0, 120.0, 99.0, 106.0, 73.0, 89.0,
2016-04-22 13:02:46.033000,2016-04-22 13:05:46.948000,498,0,0,"[95.0, 93.0, 91.0, 65.0, 70.0, 83.0, 7
2016-04-22 15:40:18.884000,2016-04-22 15:43:06.665000,484,0,0,"[39.0, 26.0, 121.0, 95.0, 233.0, 250.0
```

Рисунок 17 — Содержимое одного из файлов с построенными векторами признаков

Здесь мы видим 33 построенных вектора признаков (по одному на каждое временное окно). Для каждого вектора признаков вначале записывается время начала временного окна, затем его конец, далее записывается количество событий клавиатуры, попавших в данное временное окно. Далее через запятую записываются все рассчитанные вышеперечисленные признаки, характеризующие динамику работы пользователя с клавиатурой. Признаки, рассчитанные как статистики, представляют собой одно число. Признаки, представляющие собой временные ряды (то есть так называемые составные признаки), записываются в виде массива в кавычках и квадратных скобках. Данный внутренний формат используется для последующей обработки полученных признаков. Как мы видим, на данном рисунке изображены только начала данных векторов признаков ввиду их большой размерности.

Данный программный компонент реализован на ЯП Python 3.5 совместно с модулями numpy, pandas, user-agents, statsmodels, позволяющими существенно упростить код программы и ускорить ее работу за счет встроенных методов распараллеливания.

#### 4.2.2.5 Программный компонент обработки векторов признаков пользователя

Данный программный компонент входит в состав модуля построения индивидуальных моделей поведения пользователей, а также модуля классификации агента мониторинга и модуля анализа клавиатурного почерка пользователей в отложенном режиме, и состоит из задания обработки векторов признаков (Features Process Task). Основной задачей данного задания является чтение построенных векторов признаков, вызов необходимого процессора обработки векторов признаков пользователя (Features Processor) и запись обработанных векторов признаков в результирующий .csv-файл. Существуют шесть реализаций данного процессора: усреднение составных признаков (Mean Features Processor), выбор наиболее или наименее стабильных признаков на основе критерия Колмогорова-Смирнова (Stability Features Processor), отбор признаков методом главных компонент (PCA Features Processor), а также дискретизация по квантилям (Quantiles Discretization Features Processor), стандартизация значений признаков (Standard Scaler Features Processor) и конвейер обработки векторов признаков (Features Chain Processor). Основной задачей процессора Features Chain Processor является объединение нескольких процессоров обработки векторов признаков в один конвейер. Конвейер состоит из стадий, которые выполняются последовательно. Каждая стадия представляет собой набор отдельных трансформаций, каждая из которых представлена процессором обработки признаков, а также перечнем имен признаков, для которых должно быть применено указанное преобразование. Для отобранных (процессорами Stability Features Processor либо PCA Features Processor) признаков производится усреднение их значений в рамках каждого временного окна (процессором Mean Features Processor). Далее производится обработка полученных значений признаков процессорами дискретизации либо стандартизации. Полученные векторы признаков записываются в результирующий .csv-файл. Идентификаторы используемых отобранных признаков, а также используемые параметры алгоритмов обработки признаков записываются в соответствующий конфигурационный файл и используются для дальнейшей классификации новых тестовых данных.

Данный программный компонент реализован на ЯП Python 3.5 совместно с модулями numpy, pandas, user-agents, scipy, scikit-learn, statsmodels, позволяющими

существенно упростить код программы, ускорить ее работу за счет встроенных методов распараллеливания и предоставляющими широкие возможности для расчета различных статистик.

#### 4.2.2.6 Программный компонент построения модели пользователя

Данный программный компонент входит в состав модуля построения индивидуальных моделей поведения пользователей и состоит из задания построения модели пользователя (Model Task). Основной задачей данного задания является чтение обработанных векторов признаков обучающего набора данных, обучение модели пользователя с помощью процессора построения модели и классификации (Classifier), а также запись построенной модели в результирующий .pkl-файл. Основной задачей данного процессора является использование обработанных векторов признаков из обучающей выборки для обучения модели (и последующей классификации). Существуют различные реализации данного процессора: SVM Classifier (соответствующий методу машинного обучения SVM (SVC)), Kernel PCA Classifier (соответствующий методу машинного обучения Kernel PCA), Fuzzy Classifier (соответствующий методу машинного обучения Fuzzy), RNN Classifier (соответствующий методу машинного обучения RNN [100]) и ESFC Classifier (соответствующий методу машинного обучения ESFC). Дополнительно, поддерживаются перестройка и обновление построенной ранее (устаревшей) модели рассматриваемого пользователя.

Данный программный компонент реализован на ЯП Python 3.5 совместно с модулями numpy, pandas, user-agents, scipy, scikit-learn, statsmodels, позволяющими существенно упростить код программы, ускорить ее работу за счет встроенных методов распараллеливания и предоставляющими широкие возможности для расчета различных статистик и анализа данных.

#### 4.2.2.7 Программный компонент классификации новых данных

Данный программный компонент входит в состав модулей классификации агента мониторинга и анализа клавиатурного почерка пользователей в отложенном режиме, и состоит из задания классификации новых данных (Classification Task). Основной задачей данного задания является чтение тестовых векторов признаков пользователей, данные которых необходимо проанализировать, вызов процессора Classifier для проведения классификации новых данных с использованием обученной модели легитимного

пользователя, а также запись результатов классификации в виде рассчитанной меры аномальности и изображения построенной ROC-кривой в результирующие файлы.

Данный программный компонент реализован на ЯП Python 3.5 совместно с модулями numpy, pandas, user-agents, scipy, scikit-learn, statsmodels, позволяющими существенно упростить код программы, ускорить ее работу за счет встроенных методов распараллеливания и предоставляющими широкие возможности для расчета различных статистик и анализа данных.

#### 4.2.2.8 Программный компонент визуализации

Данный программный компонент входит в состав АРМ аналитика и состоит из задания визуализации (Visualization Task). Основной задачей данного задания является визуализация процессов сбора данных, построения векторов признаков и классификации новых данных. Для непосредственной визуализации используется процессор визуализации (Visualization Processor). Основной задачей данного процессора является визуализация одного из трёх видов заданий соответственно: задания сбора данных, задания построения векторов признаков и задания классификации. В случае визуализации задания сбора данных, строится график, отображающий количество событий, поступающих за определенный промежуток времени (этот промежуток может регулироваться изменением значения элемента «Масштаб» соответствующего графического интерфейса). В случае визуализации задания построения векторов признаков помимо задания сбора данных, по которому было построено данное задание построения векторов признаков, также визуализируется число событий, собранных до построения соответствующего вектора признаков. В случае же визуализации задания классификации, помимо заданий сбора данных и построения векторов признаков, визуализируется также степень принадлежности построенных векторов признаков истинному пользователю, по которому было создано задание построения модели.

Данный программный компонент реализован на ЯП Python 3.5 совместно с модулями numpy, pandas, user-agents, matplotlib, позволяющими существенно упростить код программы и ускорить ее работу за счет встроенных методов распараллеливания.

#### 4.2.2.9 Программный компонент оценки аномальности поведения пользователя на основе анализа целых сессий работы за компьютером

Данный программный компонент входит в состав АРМ аналитика и состоит из задания оценки аномальности поведения пользователя на основе анализа целых сессий

работы за компьютером (Evaluation Task). Основной задачей данного задания является чтение результатов классификации легитимного и тестового пользователей, полученных за целые сессии их работы за компьютером, вызов процессора Evaluation Processor для оценки аномальности поведения тестового пользователя с использованием t-статистики Уэлша [98], а также запись полученных результатов оценки аномальности в виде рассчитанной меры аномальности и изображения графика аномальности поведения пользователя в результирующие файлы. Данное задание применимо как к данным (результатам классификации), полученным из модуля анализа клавиатурного почерка пользователей в отложенном режиме, так и к данным (результатам классификации), полученным в режиме, близком к режиму реального времени (данные показатели передаются в АРМ аналитика из единого сетевого хранилища).

Данный программный компонент реализован на ЯП Python 3.5 совместно с модулями numpy, pandas, user-agents, scipy, scikit-learn, statsmodels, позволяющими существенно упростить код программы, ускорить ее работу за счет встроенных методов распараллеливания и предоставляющими широкие возможности для расчета различных статистик.

#### **4.2.3 Пример использования**

Продemonстрируем работу разработанного ЭО ПК на примере процессов поэтапного построения модели и дальнейшей классификации пользователей на основе анализа их клавиатурного почерка в отложенном режиме.

Графический интерфейс разработанного приложения представлен на Рисунке 18. Здесь мы видим следующие области: область созданных заданий и процессоров; область выполняющихся в данный момент заданий; область вывода информационных сообщений о работе системы; панель управляющих кнопок для создания заданий, процессоров и очередей заданий, а также панель для фильтрации отображаемых в данный момент заданий и процессоров.

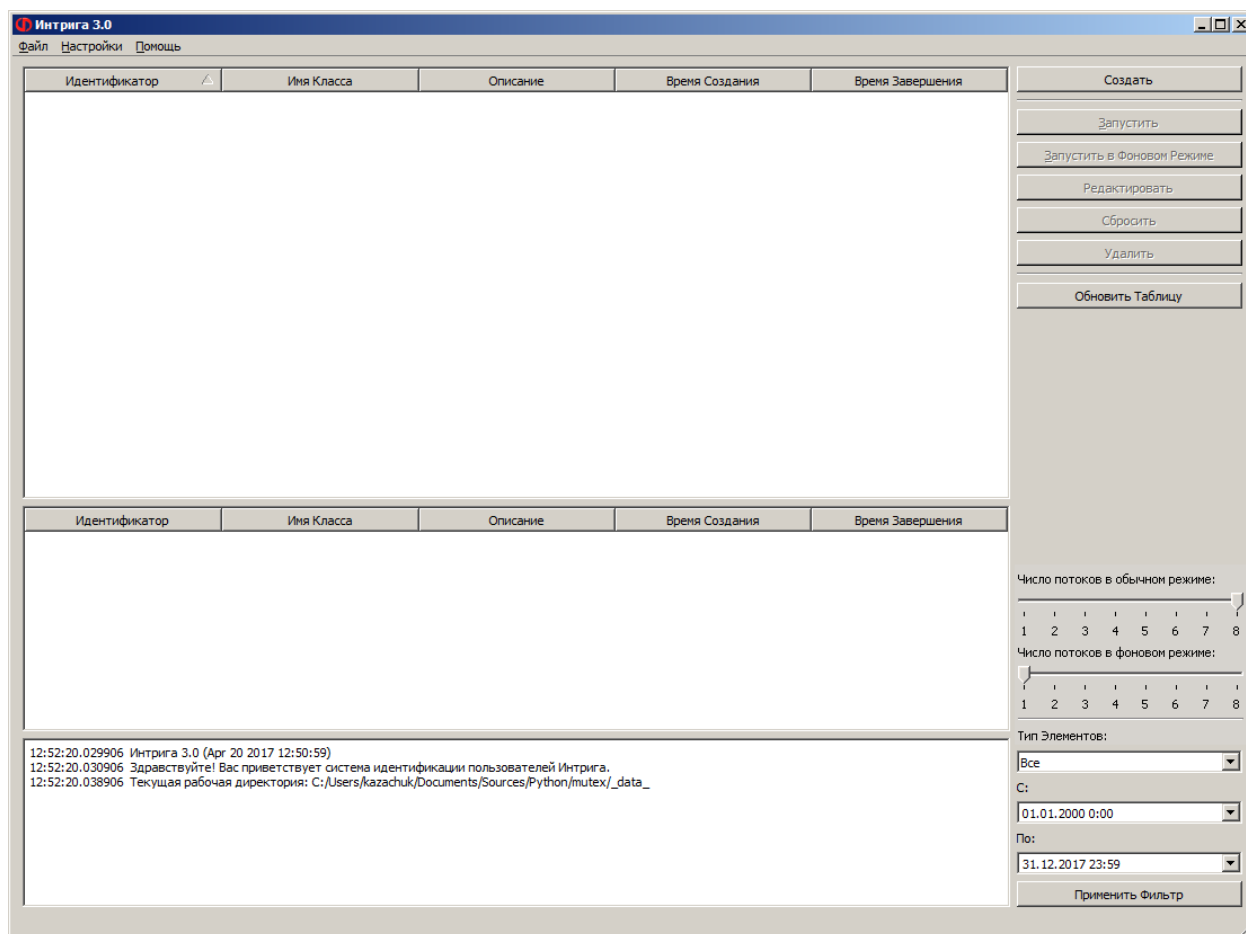


Рисунок 18 — Графический интерфейс разработанного ЭО ПК

Вначале необходимо создать задание объединения данных динамики работы пользователя с клавиатурой (Data Task). Для этого необходимо нажать на кнопку «Создать», выбрать в появившемся окне пункт «Data Task» и нажать ОК. Далее, в новом появившемся окне необходимо задать соответствующие параметры данного задания (см. Рисунок 19).

**Создать Задание Объединения Данных**

Идентификатор:  
1509437369753

Описание:  
Задание объединения данных для пользователя User22

Пользователи: Новый пользователь, User21, User22, User23

Машины: Новая машина, Без машины

Устройство ввода: keyboard

☒ Начальный момент интервала рассматриваемых событий:  
01.01.2000 00:00:00

☒ Конечный момент интервала рассматриваемых событий:  
20.04.2017 13:22:22

Дни недели: Пн, Вт, Ср, Чт, Пт, Сб, Вс

▶ Временные интервалы

☐ Имена процессов (через запятую):

☐ Директория хука: Выбрать...

☒ Частота копирования данных (мс): 1000

По умолчанию

Ок Отмена

Рисунок 19 — Создание задания объединения данных

Далее созданное задание отобразится в поле созданных заданий и процессоров (см. Рисунок 20).

**Интрига 3.0**

Файл Настройки Помощь

Идентификатор	Имя Класа	Описание	Время Создания	Время Завершения
1492682742032	DataTask	Задание объединения данных для пользователя User22	2017-04-20 13:34:52	None

Создать

Запустить

Запустить в Фоновом Режиме

Редактировать

Сбросить

Удалить

Обновить Таблицу



Рисунок 20 — Появление задания объединения данных в списке созданных заданий и процессоров

Для запуска данного задания, необходимо его выделить, нажав на него один раз левой кнопкой мыши, и нажать на кнопку «Запустить». В ходе работы данного задания в рабочую директорию запишутся соответствующие данному заданию результирующие и конфигурационные файлы. По завершении работы данного задания, в информационное поле будет выведено соответствующее сообщение о результатах его выполнения (см. Рисунок 21).

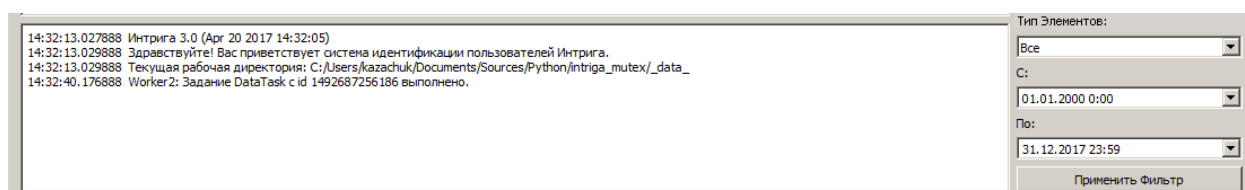


Рисунок 21 — Вывод сообщения об окончании выполнения задания

Следующим этапом является создание процессора обработки данных (Keyboard Data Processor), а также создание и запуск задания обработки объединенных данных (Data Process Task). При создании процессора обработки объединенных данных необходимо не забыть указать используемые опции предобработки. При создании задания Data Process Task необходимо выбрать соответствующее ему задание объединения данных, созданное ранее, выбрать созданный ранее процессор обработки данных и нажать ОК (см. Рисунок 22).

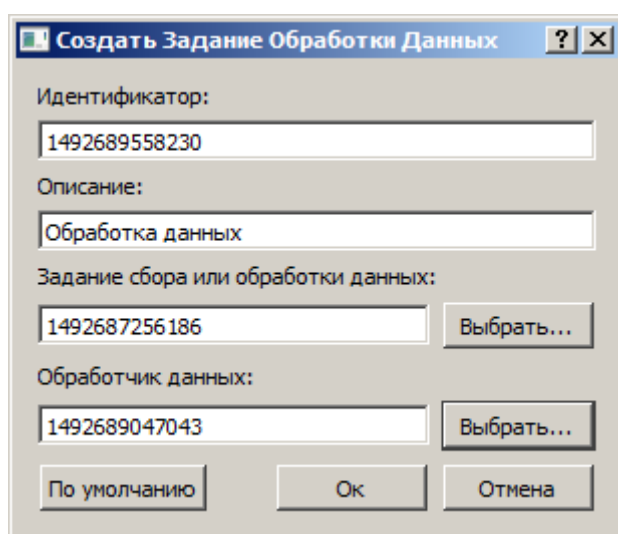


Рисунок 22 — Создание задания обработки объединенных данных

Для построения структуры векторов признаков, характеризующих динамику работы пользователя с клавиатурой, необходимо вначале создать процессор построения структуры векторов признаков («Structure Processor») и задать необходимые настройки отбора используемых для анализа клавиатурных клавиш (см. Рисунок 23):

Создать Процессор Построения Структуры Признаков

Идентификатор:  
1492693619797

Описание:  
Процессор Построения Структуры признаков

Число популярных клавиш:  
50

Число популярных диграфов:  
100

Число популярных триграфов:  
0

Число популярных триграфов с пропусками:  
0

Фильтр событий или окон:  
Нет

Необходимые клавиши для фильтра:  
Добавить...

Запрещённые клавиши для фильтра:  
Добавить...

По умолчанию    Ок    Отмена

Рисунок 23 — Создание процессора построения структуры векторов признаков

Далее необходимо создать и выполнить задание построения структуры векторов признаков (Structure Task). При создании данного задания необходимо указать соответствующие ему задание обработки объединенных данных, а также процессор построения структуры векторов признаков, созданные ранее.

Для вычисления признаков, характеризующих динамику работы пользователя с клавиатурой, вначале необходимо создать процессор построения векторов признаков («Keyboard Feature Extractor»), выбрав при этом параметры разбиения последовательности событий на временные окна (см. Рисунок 24).

**Создать Процессор Извлечения Признаков по Клавиатуре**

Идентификатор:  
1492769018882

Описание:  
Процессор извлечения признаков по клавиатуре

Режим для разбиения на окна:  
SIZE

Минимальное время для разбиения на окна (в сек.):  
240

Максимальное время для разбиения на окна (в сек.):  
420

Минимальный размер окна:  
300

Максимальный размер окна:  
500

Величина паузы при разбиении (в мс):  
80000

Процент перекрытия:  
0,50000

☐ Минимальная интенсивность (ударов в минуту):  
1

☐ Максимальная интенсивность (ударов в минуту):  
1

☒ Параметр PAUSE\_SLICE

Имя выбранной провокации:

По умолчанию      Ок      Отмена

Рисунок 24 — Создание процессора построения векторов признаков

Далее необходимо создать и выполнить задание построения векторов признаков («Feature Task»). При создании данного задания необходимо указать соответствующие ему задание обработки объединенных данных, а также процессор построения векторов признаков и файл, содержащий построенную структуру признакового пространства (полученный в результате выполнения задания Structure Task), созданные ранее.

Следующим этапом является обработка построенных векторов признаков. Для проведения данной обработки, вначале необходимо создать соответствующие процессоры: процессор дискретизации признаков по квантилям («Quantiles Discretization

Features Processor»), процессор усреднения составных признаков («Mean Features Processor»), процессор отбора наиболее или наименее стабильных признаков («Stability Features Processor»), указав при этом необходимые настройки данных процессоров. Далее необходимо создать конвейер обработки векторов признаков («Features Chain Processor»), указав применение построенного процессора дискретизации признаков к числовым признаками, применение построенного процессора стабильности признаков к списковым признаками, а также применение построенных процессоров усреднения и дискретизации признаков к стабильным признаками (см. Рисунок 25).

**Создать Конвейер Обработки Признаков**

Идентификатор:  
1492772671457

Описание:  
Конвейер обработки признаков

Добавить Трансформацию      Удалить Трансформацию

☐ Применять к Стабильным Признакам  
☐ Указать Номера Колонок(через Запятую):  
 \_\_\_\_\_

Список Процессоров Обработки Признаков в Порядке Выполнения:  
 1492771120039     

---

Трансформация №2

☒ Применять к Списковым Признакам  
☐ Применять к Числовым Признакам  
☐ Применять к Стабильным Признакам  
☐ Указать Номера Колонок(через Запятую):  
 \_\_\_\_\_

Список Процессоров Обработки Признаков в Порядке Выполнения:  
 1492771828236     

---

Трансформация №3

☐ Применять к Списковым Признакам  
☐ Применять к Числовым Признакам  
☒ Применять к Стабильным Признакам  
☐ Указать Номера Колонок(через Запятую):  
 \_\_\_\_\_

Список Процессоров Обработки Признаков в Порядке Выполнения:  
 1492771548967; 1492771120039

Рисунок 25 — Создание конвейера обработки векторов признаков

Далее необходимо создать и запустить задание обработки векторов признаков («Features Process Task»), указав при его создании используемые задание построения векторов признаков и конвейер обработки векторов признаков, созданные ранее.

Для построения одноклассовой модели пользователя по динамике его работы с клавиатурой, вначале необходимо создать процессор, соответствующий используемому классификатору. Например, для выбора классификатора Fuzzy необходимо создать процессор «Fuzzy Classifier» и задать настройки данного классификатора (см. Рисунок 26).

Создать Классификатор Fuzzy

Идентификатор:  
1493115596181

Описание:  
FuzzyClassifier

М:  
1,50000

Максимальное число итераций:  
1000

Epsilon:  
0,00001

К:  
0,10000

Kernel:  
rbf

Gamma:  
auto

☐ Random\_state:  
None

Порог аутентификации:  
0,50000

По умолчанию    Ок    Отмена

Рисунок 26 — Создание процессора, соответствующего классификатору Fuzzy

Далее необходимо создать и запустить задание построения модели пользователя по динамике его работы с клавиатурой компьютера, указав при его создании соответствующие задание обработки векторов признаков и используемый процессор для классификатора, созданные ранее, а также необходимый процент от имеющегося

количества векторов признаков пользователя для формирования валидационной (holdout) выборки.

Рассмотрим процесс классификации произвольного тестового пользователя на основе построенной вышеописанным способом легитимной модели клавиатурного почерка.

Для начала необходимо объединить данные динамики работы тестируемого пользователя с клавиатурой и предобработать их. Для этого необходимо создать и выполнить задания объединения и обработки объединенных данных (Data Task и Data Process Task). Механизм создания данных заданий полностью аналогичен созданию данных заданий для легитимного пользователя. Стоит обратить внимание, что все необходимые процессоры мы создали до построения модели легитимного пользователя, и при классификации данных тестируемого пользователя нет необходимости создавать процессоры заново, необходимо только создать и выполнить нужные задания. При создании и выполнении нужных заданий необходимо указать созданные ранее (до построения модели легитимного пользователя) процессоры.

Следующим этапом является создание и выполнение задания построения векторов признаков тестируемого пользователя (Feature Task). При его создании необходимо не забыть указать файл со структурой построенного признакового пространства для легитимного пользователя.

Далее необходимо создать и выполнить задание обработки векторов признаков тестируемого пользователя (Feature Process Task). При его создании необходимо в качестве задания, обученного для применения обработчика векторов признаков, указать Feature Process Task легитимного пользователя.

Следующим этапом является создание и выполнение задания классификации тестируемого пользователя (Classification Task). При его создании необходимо указать задание построение модели легитимного пользователя (Model Task), а также задание расчета векторов признаков тестируемого пользователя (Features Process Task), созданные ранее.

Содержимое полей графического интерфейса ЭО ПК после выполнения описанной последовательности действий представлено на Рисунке 27.

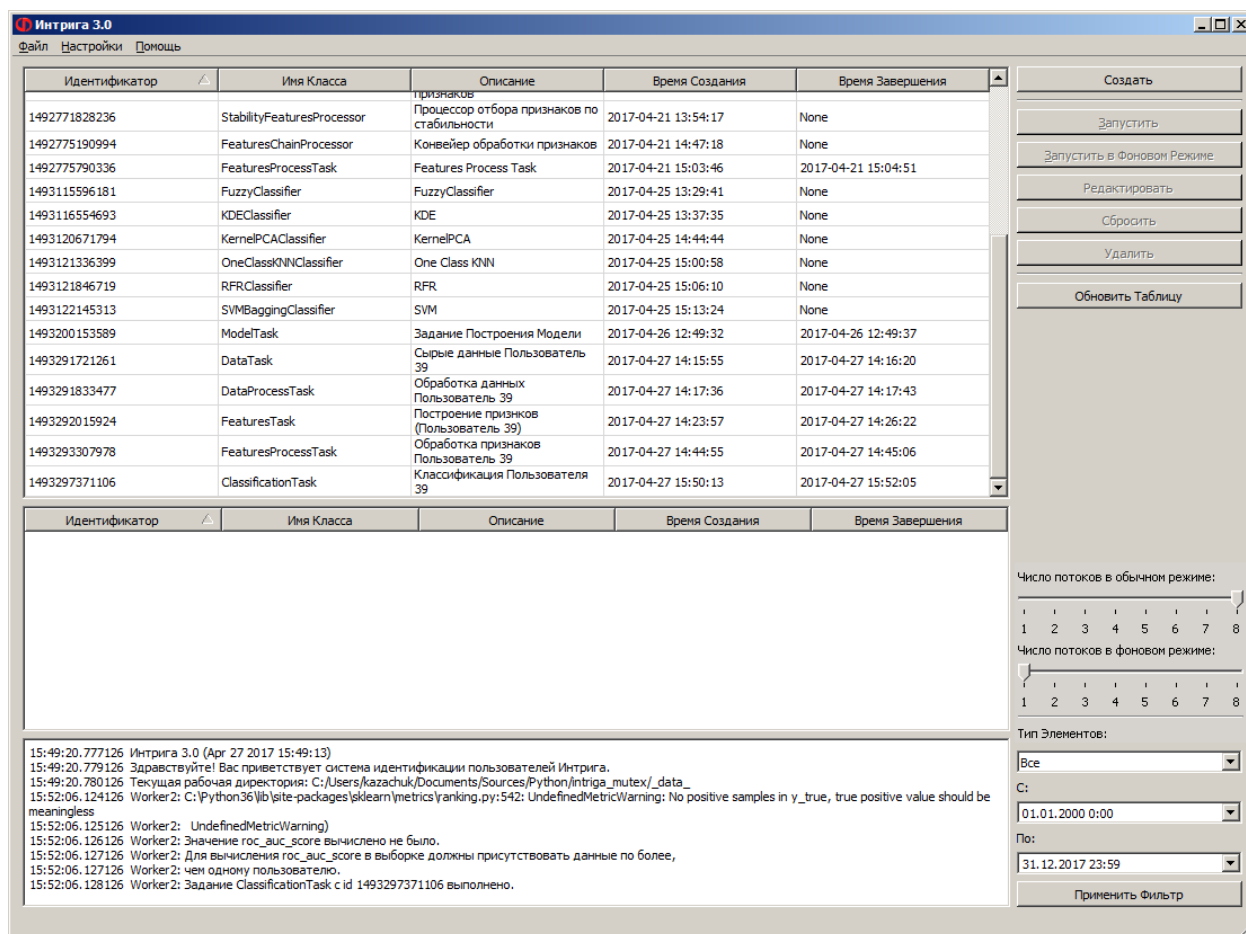


Рисунок 27 — Содержимое полей графического интерфейса после выполнения вышеописанной последовательности действий

В результате выполнения задания классификации тестируемого пользователя мы получим .csv-файл, содержащий характеристики тестируемых векторов признаков тестируемого пользователя и результаты их классификации (см. Рисунок 28).

```

1  start_time,end_time,size,res,predictions
2  2015-08-12 15:26:17.713000,2015-08-12 15:28:17.851000,500,-0.45531618050924694,No
3  2015-08-12 15:27:10.808000,2015-08-12 15:29:33.722000,500,-0.6404757880276497,No
4  2015-08-12 15:28:17.897000,2015-08-12 15:32:51.017000,500,-1.1272078299367996,No
5  2015-08-12 15:29:34.233000,2015-08-12 15:33:47.975000,500,-1.0654721426269862,No
6  2015-08-12 15:32:51.130000,2015-08-12 15:34:48.171000,390,-0.8420056908986773,No
7  2015-08-12 15:39:52.722000,2015-08-12 15:42:19.347000,500,-0.6060078041785806,No
8  2015-08-12 15:40:59.808000,2015-08-12 15:43:07.885000,500,-0.731993622285153,No
9  2015-08-12 15:42:19.500000,2015-08-12 15:44:10.271000,500,-0.8254763215995422,No
10 2015-08-12 15:43:08.494000,2015-08-12 15:46:16.300000,500,-0.2980500543907689,No
11 2015-08-12 15:44:10.586000,2015-08-12 15:47:09.198000,500,-0.8743371789738161,No
12 2015-08-12 15:46:16.602000,2015-08-12 15:48:10.580000,500,-1.089837678534641,No
13 2015-08-12 15:47:09.409000,2015-08-12 15:49:30.463000,500,-0.45548031641531384,No
14 2015-08-12 15:48:12.296000,2015-08-12 15:51:13.493000,500,-0.7037946260324326,No
15 2015-08-12 15:49:32.290000,2015-08-12 15:53:34.794000,500,-0.5074093857631254,No
16 2015-08-12 15:51:13.508000,2015-08-12 15:54:23.536000,500,-0.9035268920607767,No
17 2015-08-12 15:53:34.872000,2015-08-12 15:55:11.990000,500,-0.5854903866580388,No
18 2015-08-12 15:54:23.584000,2015-08-12 15:56:02.401000,500,-1.0348112325020975,No
19 2015-08-12 15:55:12.076000,2015-08-12 15:56:51.320000,500,-1.1355278673234177,No
20 2015-08-12 15:56:02.420000,2015-08-12 15:57:38.269000,500,-0.8234447175759126,No
21 2015-08-12 15:56:51.490000,2015-08-12 15:58:29.117000,500,-0.9536009995778162,No
22 2015-08-12 15:57:38.699000,2015-08-12 16:00:04.950000,474,-1.0278323899243524,No

```

Рисунок 28 — Содержимое файла с результатами классификации тестируемого пользователя

Структура данного файла определяется следующими колонками:

- 1) Start\_time – время прихода первого события в рассматриваемом векторе признаков;
- 2) End\_Time – время прихода последнего события в рассматриваемом векторе признаков;
- 3) Size – размер временного окна, по которому был построен рассматриваемый вектор признаков;
- 4) Res – результат классификации (величина отклика классификатора) для рассматриваемого вектора признаков;
- 5) Predictions – вердикт классификатора (легитимен ли рассматриваемый пользователь).

Помимо .csv-файла с полученными результатами классификации, в соответствующем окне мы также сможем видеть график схожести данных тестируемого пользователя с легитимным (см. Рисунок 29).

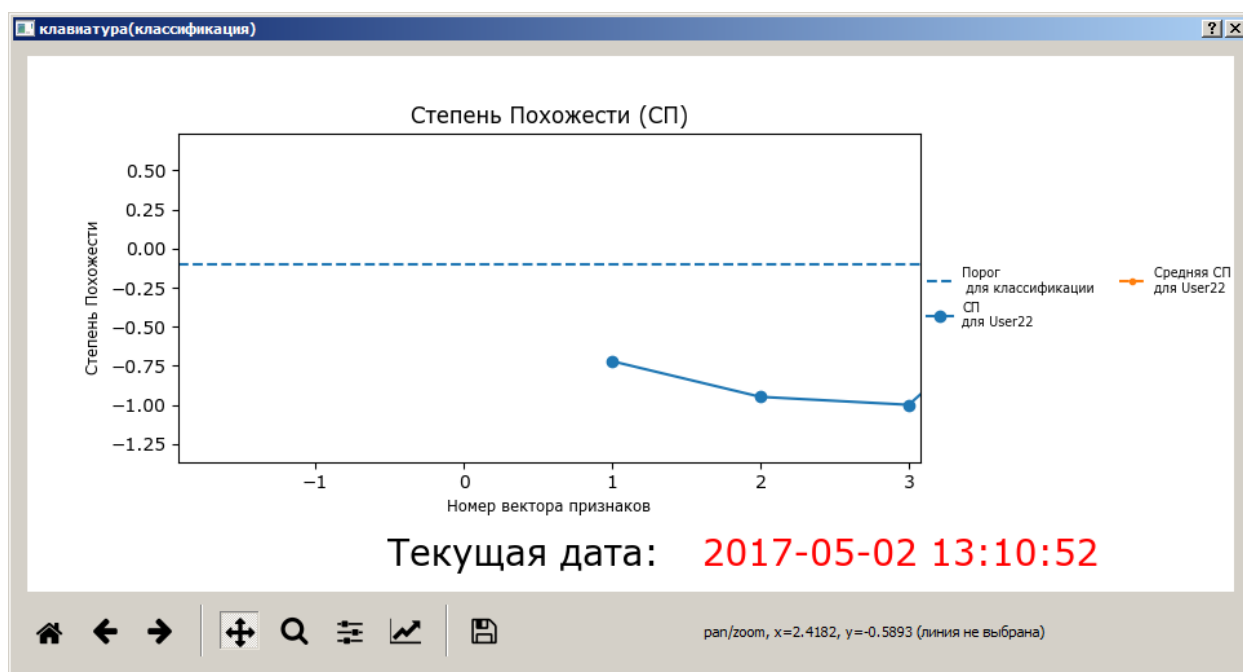


Рисунок 29 — График схожести данных тестируемого пользователя с легитимным



### 4.3 Экспериментальное исследование

В разделах 2–3 данной диссертационной работы было проведено исследование качества работы разработанных алгоритмов применительно к наборам данным, содержащим данные динамики работы 10–20 пользователей с клавиатурой компьютера. Для подтверждения высокого качества работы предложенных алгоритмов, было проведено дополнительное экспериментальное исследование на наборе данных Villani [38] (Набор данных №1), содержащем данные динамики работы 144 пользователей с клавиатурой компьютера, собранные в фоновом режиме. В среднем, каждый пользователь проработал за компьютером порядка одного рабочего дня. Для каждого действия пользователя записывался код клавиши, тип действия (нажатие или отпускание) и время совершения действия. Для проведения экспериментов, тестовые данные были переведены в формат, используемый в разработанном ЭО ПК (см. Раздел 4.2.2.1). Для построения векторов признаков использовались значения параметров, подобранные на предыдущем этапе экспериментальных исследований (см. Раздел 2.7):

- **Локальные данные:**
  - **Размер окна: (300 – 500) событий;**
  - **Процент перекрытия: 0.01;**
  - **Количество квантилей для дискретизации: 7;**
  - **Пауза при разбиении на временные окна: 40 секунд;**
  - **Количество рассматриваемых наиболее часто используемых одиночных клавиш: 50;**
  - **Количество рассматриваемых наиболее часто используемых диграфов: 100;**
  - **Использование группового признака (среднего времени удержания группы клавиш во временном окне);**
  - **Отбор признаков, уровень стабильности которых выше 0.5**

Для построения модели пользователя использовались классификаторы SVC, Kernal PCA, Fuzzy, ESFC и RNN (репликаторная нейронная сеть, [100]). Подбор оптимальных значений метапараметров алгоритмов машинного обучения осуществлялся разработанным в данной диссертационной работе методом (см. Раздел 3.4). Имеющиеся данные по каждому пользователю в равных долях были разбиты на три части: обучающую, валидационную и тестовую выборки. Валидационная выборка использовалась для подбора оптимальных значений метапараметров одноклассовых

классификаторов, а также для оценки аномальности поведения пользователя на основе анализа целых сессий работы за компьютером с использованием t-статистики Уэлша (см. Раздел 3.5). В качестве порогового значения на количество векторов обучающей выборки было установлено ограничение в 25 векторов признаков. Для оценки качества классификации использовалось медианное значение величины ROC AUC в сочетании с межквартильным разбросом (усредненные по всем пользователям). Исследовалось качество работы алгоритмов при классификации как отдельных событий (отдельных векторов признаков), так и целых сессий работы пользователей за компьютером.

Результаты экспериментов при классификации отдельных событий представлены в Таблице 23.

Таблица 23 — Результаты проведенных экспериментов на наборе данных Villani (классификация отдельных событий)

	Стандартизация признаков	Отбор признаков по стабильности, Стандартизация признаков	Дискретизация признаков по квантилям	Отбор признаков по стабильности, Дискретизация признаков по квантилям
SVC	0.8195±0.0778	0.8744±0.0722	0.8745±0.0715	0.9262±0.0676
Fuzzy	0.8293±0.0777	0.8749±0.0725	0.8763±0.0707	0.9306±0.0650
Kernel PCA	0.8213±0.0777	0.8746±0.0721	0.8761±0.0709	0.9315±0.0647
ESFC	0.8306±0.0775	0.8905±0.0715	0.8925±0.0628	<b>0.9486±0.0521</b>
RNN	0.8197±0.0779	0.8747±0.0723	0.8751±0.0714	0.9263±0.0655

Наилучший результат распознавания (Медианное значение ROC AUC = 0.9486, Разброс ROC AUC = 0.0521) был получен при использовании классификатора ESFC со следующими значениями метапараметров, подобранными разработанным в данной работе алгоритмом:

- Ширина ядра = 0.006;
- Процент исключений = 5%;
- Степень нечеткости = 1.5.

Далее результаты проведенной повекторной классификации (полученные при использовании найденной оптимальной композиции алгоритмов – отбора признаков по уровню их стабильности и дискретизации наиболее стабильных признаков по квантилям)

использовались для оценки аномальности поведения пользователя на основе анализа целых сессий работы за компьютером с использованием t-статистики Уэлша. Для проверки качества работы данного метода в качестве метрики также использовалось медианное значение площади под ROC-кривой (ROC AUC) в сочетании с межквартильным разбросом (усредненные по всем пользователям). При этом, в качестве откликов для построения ROC-кривой использовались значения p-value, полученные в результате расчета t-статистики Уэлша для каждой сессии (сеанса работы за компьютером) тестируемого пользователя и валидационной выборки легитимного пользователя.

Результаты экспериментов при классификации целых сессий работы пользователей за компьютером представлены в Таблице 24.

Таблица 24 — Результаты проведенных экспериментов на наборе данных Villani (классификация целых сессий)

Метод построения модели пользователя	Результат
SVC	0.9414±0.0521
Fuzzy	0.9509±0.0473
Kernel PCA	0.9513±0.0453
ESFC	<b>0.9683±0.0301</b>
RNN	0.9417±0.0523

В результате проведенных экспериментов было получено, что в случае анализа целых сессий работы пользователей за компьютером, наилучший результат распознавания также достигается при использовании классификатора ESFC: медианное значение показателя ROC AUC в сочетании с межквартильным разбросом составляет порядка **0.9683±0.0301**, что свидетельствует о высоком качестве работы разработанных алгоритмов.

На основании полученных результатов можно сделать следующие выводы:

- Предложенная комбинация разработанных алгоритмов (отбор признаков по уровню их стабильности с использованием критерия Колмогорова-Смирнова, дискретизация по квантилям наиболее стабильных признаков, построение модели пользователя с использованием метода ESFC и разработанного алгоритма подбора оптимальных значений метапараметров одноклассовых методов машинного обучения) показывает наилучшее качество работы на всех рассмотренных четырех наборах тестовых данных

(выборки содержат данные клавиатурного почерка от 10 до 140 пользователей) и позволяет улучшить качество работы базовых алгоритмов (ROC AUC) порядка на 9–12%;

- С помощью предложенного метода оценки аномальности поведения пользователя на основе анализа целых сессий работы за компьютером удастся достичь достаточно высокого качества распознавания: медианное значение показателя ROC AUC достигает порядка **0.93–0.97** на всех рассмотренных наборах тестовых данных;
- Для всех рассмотренных методов построения модели пользователя классификация по сессиям дала более точную оценку, чем классификация по отдельным событиям (векторам признаков).

#### **4.4 Оценка производительности**

Для оценки производительности разработанного ЭО ПК была проведена дополнительная серия экспериментальных исследований. Рассматривалось среднее время выполнения каждого из основных компонентов ЭО ПК, а также максимальная нагрузка ЦП и максимальный объем оперативной памяти, потребляемые при работе системы (усредненные по всем пользователям).

Эксперименты проходили на рабочей станции, имеющей следующие аппаратные и программные характеристики: программные характеристики – ОС Windows Server 2008 R2 Enterprise (64bit), Python 3.5.2; аппаратные характеристики: процессор Intel(R) Xeon(R) CPU E5504, частота 2 ГГц, 2 ядра; оперативная память 24 Гбайт; дисковый накопитель HDD, 697 Гбайт.

Отметим, что одной из главных особенностей систем динамической аутентификации является необходимость их работы в режиме, близком к режиму реального времени. Сбор данных и аутентификация пользователей должны происходить в фоновом режиме незаметно для пользователей и не нагружать систему: для пользователя не должны быть видны никакие дополнительные задержки при работе с компьютером. Для проверки незаметности работы компонентов сбора и аутентификации пользователей, были привлечены 10 независимых экспертов, на рабочие компьютеры которых было установлено разработанное ПО. Каждый из экспертов проработал за своим компьютером порядка одного рабочего дня и не заметил никаких дополнительных задержек в работе компьютера: реакция системы не отличалась от работы при выключенных компонентах

ЭО ПК. Также дополнительный анализ показал, что все клавиатурные события были обработаны своевременно, и вся информация о них была записана в необходимые файлы в полном объеме.

Для оценки производительности основных компонентов разработанного ЭО ПК использовался Набор данных 2, содержащий данные динамики работы 20 пользователей с клавиатурой, собранные в фоновом режиме их работы за компьютером. В среднем, каждый пользователь проработал за компьютером порядка 10 часов. Общий объем собранных данных составил порядка 130 МБайт.

Результаты проведенного экспериментального исследования (усредненные по всем пользователям) приведены в Таблице 25. Отметим, что выполнение задач в ЭО ПК осуществлялось в последовательном режиме. При использовании параллельного режима работы, общее время выполнения экспериментов сокращается примерно в  $0.75 \cdot N$  раз, где  $N$  – число используемых процессов. В программных компонентах использовалась полученная оптимальная комбинация разработанных алгоритмов (отбор признаков по уровню их стабильности с использованием критерия Колмогорова-Смирнова, дискретизация по квантилям наиболее стабильных признаков, построение модели пользователя с использованием метода ESFC и разработанного алгоритма подбора оптимальных значений метапараметров одноклассовых методов машинного обучения).

Таблица 25 — Характеристики производительности разработанного ЭО ПК

Название программного компонента	Среднее время работы	Максимальная нагрузка ЦП (%)	Максимальный объем потребляемой ОП (МБайт)
Программный компонент сбора данных клавиатурного почерка пользователей	Средний объем собранных данных: 6399 КБ; Среднее время работы: 10 часов	1	1
Программный компонент обработки данных клавиатурного	Средний объем входных данных: 6399 КБ; Среднее время	5	6

<b>Название программного компонента</b>	<b>Среднее время работы</b>	<b>Максимальная нагрузка ЦП (%)</b>	<b>Максимальный объем потребляемой ОП (МБайт)</b>
почерка пользователей	работы: 9 сек		
Программный компонент построения структуры векторов признаков	Средний объем входных данных: 5320 КБ; Среднее время работы: 21 сек	5	6.25
Программный компонент построения векторов признака пользователя	Средний объем входных данных: 5322 КБ; Среднее время работы: 300 сек	6	6.75
Программный компонент обработки векторов признаков пользователя	Средний объем входных данных: 389 КБ; Среднее количество векторов признаков: 97; Среднее время работы: 13 сек	6	11
Программный компонент построения модели пользователя	Среднее количество векторов признаков: 33; Среднее время работы: 0.04 сек	8	9.85
Программный компонент классификации новых данных	Среднее количество векторов признаков для классификации: 33; Среднее время работы: 0.12 сек	6	9.75

Название программного компонента	Среднее время работы	Максимальная нагрузка ЦП (%)	Максимальный объем потребляемой ОП (МБайт)
Программный компонент оценки аномальности поведения пользователя на основе анализа целых сессий работы за компьютером	Среднее количество векторов признаков валидационной выборки: 33; Среднее количество векторов признаков тестовой выборки: 33; Среднее время работы: 0.004 сек	6	9.5

Из приведенных в Таблице 25 данных можно сделать следующие выводы:

- Разработанный ЭО ПК обладает достаточно высокой производительностью:
  - Все поступающие в систему клавиатурные события обрабатываются верно, своевременно и в полном объеме;
  - Все программные компоненты, включая наиболее критичные для пользователя компоненты, которые устанавливаются непосредственно на рабочие места пользователей и работа которых должна быть внешне незаметна, затрачивают незначительное количество аппаратных ресурсов компьютера и не влияют на скорость работы других приложений: в среднем, на обработку одного вектора признаков (включающего информацию о порядка 300–500 клавиатурных событиях) уходит порядка 3.5 секунд и 0.11 МБайт ОП, что свидетельствует о возможности применения разработанного ЭО ПК на практике.

## 4.5 Выводы

В данном разделе была проведена разработка экспериментального образца программного комплекса (ЭО ПК) динамической аутентификации пользователей по динамике их работы с клавиатурой персонального компьютера (ноутбука), в основе

работы которого лежат предложенные в данной работе алгоритмы. Были достигнутые следующие результаты:

- Был разработан ЭО ПК динамической аутентификации пользователей по динамике их работы с клавиатурой персонального компьютера (ноутбука), обладающий высокой производительностью, устойчивостью к возможным ошибкам во входных данных, поддерживающий режим многозадачности и реализующий три следующих базовых сценария функционирования:
  - Сбор поведенческой информации о взаимодействии пользователей с клавиатурой компьютера;
  - Построение индивидуальных моделей поведения пользователей;
  - Применение индивидуальных моделей поведения пользователей.
- Была подробно описана программная реализация разработанного ЭО ПК (описаны архитектура системы, принцип работы программных компонентов, описан графический интерфейс и приведен пошаговый пример использования системы). Поскольку разработанный ЭО ПК представляет собой набор взаимосвязанных между собой программных агентов, каждый из которых выполняет собственную логическую задачу, данный ЭО ПК обладает свойствами масштабируемости (возможностью распределять программные компоненты по разным физическим машинам) и расширяемости (возможностью добавлять либо заменять отдельные программные модули и компоненты).
- Проведено дополнительное экспериментальное исследование работы разработанного ЭО ПК на большом объеме данных (данных клавиатурного почерка 144 пользователей), подтвердившее высокое качество работы разработанных алгоритмов (порядка 0.90–0.95 ROC AUC в случае классификации отдельных векторов признаков и порядка 0.97 ROC AUC в случае классификации целых сессий работы пользователей за компьютером).



## 5 ЗАКЛЮЧЕНИЕ

Основные результаты диссертационной работы заключаются в следующем:

1. Предложен подход к подготовке данных, описывающих клавиатурный почерк пользователя, включающий в себя способ построения признакового пространства и подход к дальнейшей обработке признаков на основе дискретизации их по квантилям. Сокращение размерности признакового пространства производится путем отбора наиболее значимых признаков с использованием критерия Колмогорова-Смирнова. Экспериментально установлено, что данный подход позволяет построить пространство стабильных по времени признаковых характеристик;
2. Разработан нечеткий метод выявления аномалий в данных на основе эллиптической кластеризации (ESFC) в RKHS, строящий в пространстве высокой размерности эллиптические области с оптимальным центром для выявления аномалий. Подбор оптимальных значений метапараметров данного алгоритма осуществляется собственно разработанным методом, строящим стабильные к смене тестового набора данных одноклассовые модели без использования информации о данных нелегитимного класса. Оценка аномальности поведения пользователя производится как за короткий, так и за продолжительный период работы – с использованием разработанного метода оценки аномальности поведения пользователей на основе анализа целых сессий работы за компьютером с использованием t-статистики Уэлша. По результатам экспериментов, метод ESFC превзошел качество распознавания существующих алгоритмов при классификации как отдельных векторов признаков, так и целых сессий работы пользователей за компьютером;
3. Разработана архитектура, реализован и апробирован экспериментальный образец мультиагентного программного комплекса, использующий предложенный комплекс алгоритмов для обнаружения аномального поведения пользователей по особенностям работы с клавиатурой компьютера. Проведенные на его основе экспериментальные исследования подтвердили качество и обосновали достоверность полученных результатов.

Данные результаты опубликованы в 7 печатных работах [14 – 20].

Разработанный экспериментальный образец программного комплекса прошел апробацию в рамках НИР «Разработка технологий биометрической идентификации пользователя по признакам, проявляющимся при использовании устройств ввода данных персональных ЭВМ» (Номер договора №01-04/15 от 08 апреля 2015 г), 2015–2017 гг.

Полученные результаты диссертационной работы могут послужить основой для построения перспективных современных систем информационной безопасности, которые будут включать в себя средства анализа динамики работы пользователей с клавиатурой компьютера. При этом, могут использоваться как все разработанные модули, так и отдельные из них (например, модули сбора данных о динамике работы пользователей с клавиатурой персонального компьютера / ноутбука).

## 6 СПИСОК ЛИТЕРАТУРЫ

- 1 Руководящий документ. Защита от несанкционированного доступа к информации. Термины и определения. Гостехкомиссия России, 30 марта 1992 г. [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2019. — Режим доступа: <https://fstec.ru/component/attachments/download/298>. — 08.08.2019.
- 2 Cao K., Jain A. K. Hacking mobile phones using 2D printed fingerprints // Technical Report. – 2016.
- 3 ГОСТ Р 52633.0-2006. Защита информации. Техника защиты информации. Требования к средствам высоконадежной биометрической аутентификации [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2019. — Режим доступа: <http://docs.cntd.ru/document/1200048922>. — 08.08.2019.
- 4 ГОСТ Р 54412-2011/ISO/IEC/TR 24741:2007. Информационные технологии (ИТ). Биометрия. Обучающая программа по биометрии [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2019. — Режим доступа: <http://docs.cntd.ru/document/1200094221>. — 08.08.2019.
- 5 ГОСТ ISO/IEC 24713-1-2013 Информационные технологии (ИТ). Биометрические профили для взаимодействия и обмена данными. Часть 1. Общая архитектура биометрической системы и биометрические профили [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2019. — Режим доступа: <http://docs.cntd.ru/document/1200107284>. — 08.08.2019.
- 6 Казанцев И. С. Анализ клавиатурного почерка в процессах аутентификации, идентификации и обнаружения подмены оператора // Молодой ученый. — 2016. — №9. — С. 167-169 [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2019. — Режим доступа: <https://moluch.ru/archive/113/28900/>. — 08.08.2019.
- 7 Машечкин И.В., Петровский М.И., Попов И.С., Терехин А.Н., Глазкова В.В., Курынин Р.В., Головин С.И., Царёв Д.В., Казачук М.А. Отчет о прикладных научных исследованиях (промежуточный) по теме «Теоретические исследования (2-ой очереди) поставленных перед ПНИ задач». — М., 2015.
- 8 Машечкин И.В., Петровский М.И., Попов И.С., Терехин А.Н., Саликов П.М., Горохов О.Е., Никифоров Д.А., Глазкова В.В., Казачук М.А. Отчет о прикладных научных исследованиях (заключительный) по теме «Обобщение и оценка результатов исследований (этап 5)». — М., 2016.

9 Машечкин И.В., Попов И.С., Терехин А.Н., Петровский М.И., Царёв Д.В., Корчагин А.Ю., Глазкова В.В., Саликов П.М., Казачук М.А., Горохов О.Е., Никифоров Д.А. Отчет о прикладных научных исследованиях (промежуточный) по теме «Теоретические исследования (3-ей очереди) поставленных перед ПНИ задач». — М., 2015.

10 Машечкин И.В., Петровский М.И., Попов И.С., Царёв Д.В., Терехин А.Н., Глазкова В.В., Никифоров Д.А., Горохов О.Е., Саликов П.М., Казачук М.А. Отчет о прикладных научных исследованиях (промежуточный) по теме «Экспериментальные исследования (этап 4) поставленных перед ПНИ задач». — М., 2016.

11 Машечкин И.В., Терехин А.Н., Петровский М.И., Глазкова В.В., Попов И.С., Казачук М.А., Ковальчук А.А., Закляков Р.Д., Орпанен И.С. Отчет о научно-исследовательской работе «Разработка технологий биометрической идентификации пользователя по признакам, проявляющимся при использовании устройств ввода данных персональных ЭВМ» (шифр «Интрига-М»). Номер договора №01-04/15 от 08 апреля 2015 г. Этап 1. — М., 2015.

12 Машечкин И.В., Терехин А.Н., Петровский М.И., Глазкова В.В., Попов И.С., Казачук М.А., Ковальчук А.А., Закляков Р.Д., Орпанен И.С., Горохов О.Е. Отчет о научно-исследовательской работе «Разработка технологий биометрической идентификации пользователя по признакам, проявляющимся при использовании устройств ввода данных персональных ЭВМ» (шифр «Интрига-М»). Номер договора №01-04/15 от 08 апреля 2015 г. Этап 2. — М., 2016.

13 Машечкин И.В., Терехин А.Н., Петровский М.И., Попов И.С., Казачук М.А., Ковальчук А.А., Закляков Р.Д., Горохов О.Е. Отчет о научно-исследовательской работе «Разработка технологий биометрической идентификации пользователя по признакам, проявляющимся при использовании устройств ввода данных персональных ЭВМ» (шифр «Интрига-М»). Номер договора №01-04/15 от 08 апреля 2015 г. Этап 3. — М., 2017.

14 Kazachuk M., Kovalchuk A., Mashechkin I., Orpanen I., Petrovskiy M., Popov I., Zakliakov R. One-Class Models for Continuous Authentication Based on Keystroke Dynamics //International Conference on Intelligent Data Engineering and Automated Learning. – Springer, Cham, 2016. – С. 416-425.

15 Kazachuk M., Petrovskiy M., Mashechkin I., Gorokhov O. Novelty Detection Using Elliptical Fuzzy Clustering in a Reproducing Kernel Hilbert Space //International Conference on Intelligent Data Engineering and Automated Learning. –Springer, Cham, 2018. – С. 221-232.

- 16 Методы поиска исключений в потоках сложноструктурированных данных / М. А. Казачук, М. И. Петровский, И. В. Машечкин, О. Е. Горохов //Вестник Московского университета. Серия 15: Вычислительная математика и кибернетика. – 2019. – № 3. – С. 17-28.
- 17 Казачук М. А., Машечкин И. В., Петровский М. И., Терехин А.Н., Попов И.С., Закляков Р.Д., Горохов О.Е. Методы активной аутентификации пользователей по особенностям работы с клавиатурой персонального компьютера //«Ломоносовские чтения»: Научная конференция. – Москва: МАКС Пресс, 2017. – С. 112-112.
- 18 Казачук М. А., Машечкин И. В., Петровский М. И. Исследование и разработка методов активной аутентификации пользователей по динамике их работы с клавиатурой персонального компьютера //«Тихоновские чтения»: Научная конференция. – Москва: МАКС Пресс, 2017. – С. 79-80.
- 19 Казачук М. А., Петровский М. И., Машечкин И. В. Нечеткий метод выявления аномалий в данных на основе эллиптической кластеризации в RKHS //«Тихоновские чтения»: Научная конференция. — Москва: МАКС Пресс, 2018. – С. 91-91.
- 20 Казачук М. А., Петровский М. И., Машечкин И. В. Применение нечеткой кластеризации в задаче динамической аутентификации пользователей по клавиатурному почерку //«Ломоносовские чтения»: Научная конференция. – Москва: МАКС Пресс, 2019. – С. 64-65.
- 21 BehavioSec. Integrate Behavior Biometrics Authentication [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2019. — Режим доступа: <https://www.behaviosec.com/>. — 08.08.2019.
- 22 KeyTrac. Keyboard biometrics made simple for you [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2019. — Режим доступа: <https://www.keytrac.net/>. — 08.08.2019.
- 23 KeystrokeID [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2019. — Режим доступа: <https://www.idcontrol.com/keystrokeid>. — 08.08.2019.
- 24 ScoutAnalytics [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2019. — Режим доступа: <https://my.scoutanalytics.com>. — 08.08.2019.
- 25 Пат. US8332932B2 US. Keystroke dynamics authentication techniques. Inventor: Mechthild: R. Kellas-DicksYvonne J. Stark. Original Assignee: Scout Analytics Inc. Priority date: 2007-12-07.
- 26 TypingDNA. Recognize people by the way they type [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2019. — Режим доступа: <https://www.typingdna.com/>. — 08.08.2019.

- 27 KeystrokeDNA. Biometric authentication as a service [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2019. — Режим доступа: <https://keystrokedna.com/>. — 08.08.2019.
- 28 Biocatch. CONTINUOUS AUTHENTICATION [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2019. — Режим доступа: <https://www.biocatch.com/continuous-authentication-solutions>. — 08.08.2019.
- 29 BioTracker. Stop Threats in Seconds with BioTracker Defend [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2019. — Режим доступа: <https://www.plurilock.com/products/biotracker-defend-edr/>. — 08.08.2019.
- 30 Intensity Analytics. Protecting your identity is just a start. Own your identity [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2019. — Режим доступа: <https://www.intensityanalytics.com/Default.aspx>. — 08.08.2019.
- 31 Symantec.VIP. Secure Authentication Anywhere [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2019. — Режим доступа: <https://vip.symantec.com/>. — 08.08.2019.
- 32 NuData Security. Mastercard. Imagine a world that protects you and your customers [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2019. — Режим доступа: <https://nudatasecurity.com/>. — 08.08.2019.
- 33 NoPassword. NoPassword Authentication [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2019. — Режим доступа: <https://www2.nopassword.com/authentication/>. — 08.08.2019.
- 34 Alsolami E. et al. User-representative feature selection for keystroke dynamics. – 2011.
- 35 Ceker H., Upadhyaya S. Enhanced recognition of keystroke dynamics using Gaussian mixture models //Military Communications Conference, MILCOM 2015-2015 IEEE. – IEEE, 2015. – С. 1305-1310.
- 36 Everitt R. A. J., McOwan P. W. Java-based internet biometric authentication system //IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2003. – Т. 25. – №. 9. – С. 1166-1172.
- 37 Monaco J. V. et al. Developing a keystroke biometric system for continual authentication of computer users //Intelligence and Security Informatics Conference (EISIC), 2012 European. – IEEE, 2012. – С. 210-216.
- 38 Tappert C. C., Villani M., Cha S. H. Keystroke biometric identification and authentication on long-text input //Behavioral biometrics for human identification: Intelligent applications. – IGI Global, 2010. – С. 342-367.
- 39 Bello L. et al. Collection and publication of a fixed text keystroke dynamics dataset //XVI Congreso Argentino de Ciencias de la Computación. – 2010.

- 40 Silva L. A. O. Behavioural biometrics in the World Wide Web : дис. – 2014.
- 41 Liu J. et al. The beihang keystroke dynamics authentication system //arXiv preprint arXiv:1310.4485. – 2013.
- 42 Çeker H., Upadhyaya S. User authentication with keystroke dynamics in long-text data //Biometrics Theory, Applications and Systems (BTAS), 2016 IEEE 8th International Conference on. – IEEE, 2016. – C. 1-6.
- 43 Kang P., Cho S. Keystroke dynamics-based user authentication using long and free text strings from various input devices //Information Sciences. – 2015. – Т. 308. – C. 72-93.
- 44 Namin, A.S.: Cyberspace security use keystroke dynamics //Ph.D. thesis, Texas Tech University – 2015.
- 45 Bailey K. O., Okolica J. S., Peterson G. L. User identification and authentication using multi-modal behavioral biometrics //Computers & Security. – 2014. – Т. 43. – C. 77-89.
- 46 Bailey K. O. Computer based behavioral biometric authentication via multi-modal fusion. – AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH GRADUATE SCHOOL OF ENGINEERING AND MANAGEMENT, 2013. – №. AFIT-ENG-13-M-04.
- 47 Bakelman N. et al. Continual keystroke biometric authentication on short bursts of keyboard input //Proceedings of Student-Faculty Research Day, CSIS, Pace University. – 2012.
- 48 Babu B. M., Bhanu M. S. Prevention of insider attacks by integrating behavior analysis with risk based access control model to protect cloud //Procedia Computer Science. – 2015. – Т. 54. – C. 157-166.
- 49 Morales A. et al. Keystroke dynamics recognition based on personal data: A comparative experimental evaluation implementing reproducible research //Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on. – IEEE, 2015. – C. 1-6.
- 50 Ponkshe R. V., Chole V. AUTHENTICATION ENHANCEMENT BASED ON MOUSE AND KEYSTROKE FEATURES. – 2015.
- 51 Об утверждении Классификатора программ для электронных вычислительных машин и баз данных (с изменениями на 1 апреля 2016 года). [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2019. — Режим доступа: <http://docs.cntd.ru/document/420337007>. — 09.08.2019.
- 52 Mikowski M., Powell J. Single page web applications: JavaScript end-to-end. – Manning Publications Co., 2013.
- 53 Smith D. NPAPI Plugin Perspectives and the Oracle JRE //Retrieved April. – 2015. – Т. 25.

- 54 Yin J. et al. WebC: toward a portable framework for deploying legacy code in web browsers //Science China Information Sciences. – 2015. – Т. 58. – №. 7. – С. 1-15.
- 55 Pour G. Understanding software component technologies: JavaBeans and ActiveX //Proceedings Technology of Object-Oriented Languages and Systems. TOOLS 29 (Cat. No. PR00275). – IEEE, 1999. – С. 398-398.
- 56 Tilkov S., Vinoski S. Node.js: Using JavaScript to build high-performance network programs //IEEE Internet Computing. – 2010. – Т. 14. – №. 6. – С. 80-83.
- 57 Killourhy K. S., Maxion R. A. Comparing anomaly-detection algorithms for keystroke dynamics //Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP international conference on. – IEEE, 2009. – С. 125-134.
- 58 Kabir M. N. et al. On the development of a web extension for text authentication on Google Chrome //2019 International Conference on Electrical, Computer and Communication Engineering (ECCE). – IEEE, 2019. – С. 1-5.
- 59 Быков А. В., Пьянков С. В. Web-картографирование: учеб. пособие //Перм. гос. нац. исслед. университет. – 2015. – Т. 110.
- 60 Balagurusamy E. Programming with JAVA. – McGraw-Hill Education, 2019. – Т. 6.
- 61 Lagartos I., Redondo J. M., Ortin F. Efficient runtime metaprogramming services for Java //Journal of Systems and Software. – 2019. – Т. 153. – С. 220-237.
- 62 Teh P. S., Teoh A. B. J., Yue S. A survey of keystroke dynamics biometrics //The Scientific World Journal. – 2013. – Т. 2013.
- 63 Witno S. et al. Monitoring Computer Activities with Cloud to Device Messaging (C2DM) //Tech-E. – 2018. – Т. 1. – №. 2. – С. 35-42.
- 64 Alsultan A., Warwick K. Keystroke dynamics authentication: a survey of free-text methods //International Journal of Computer Science Issues (IJCSI). – 2013. – Т. 10. – №. 4. – С. 1.
- 65 Deng Y., Zhong Y. Keystroke dynamics user authentication using advanced machine learning methods //Recent Advances in User Authentication Using Keystroke Dynamics Biometrics, GCSR. – 2015. – Т. 2. – С. 23-40.
- 66 Saevanee H. et al. Continuous user authentication using multi-modal biometrics //Computers & Security. – 2015. – Т. 53. – С. 234-246.
- 67 Ahmed A. A. E., Traore I., Ahmed A. Digital Fingerprinting Based on Keystroke Dynamics //HAISA. – 2008. – С. 94-104.



- 68 Kim J., Kim H., Kang P. Keystroke dynamics-based user authentication using freely typed text based on user-adaptive feature extraction and novelty detection //Applied Soft Computing. – 2018. – Т. 62. – С. 1077-1087.
- 69 Idrus S. Z. S. et al. Soft biometrics for keystroke dynamics //International Conference Image Analysis and Recognition. – Springer, Berlin, Heidelberg, 2013. – С. 11-18.
- 70 Kim J., Kang P. Recurrent neural network-based user authentication for freely typed keystroke data //arXiv preprint arXiv:1806.06190. – 2018.
- 71 Bhattasali T. et al. Modular logic of authentication using dynamic keystroke pattern analysis //AIP Conference Proceedings. – AIP Publishing, 2016. – Т. 1738. – №. 1. – С. 180012.
- 72 Shanmugapriya D., Padmavathi G. An efficient feature selection technique for user authentication using keystroke dynamics //IJCSNS International Journal of Computer Science and Network Security. – 2011. – Т. 11. – №. 10. – С. 191-195.
- 73 Jolliffe I. Principal component analysis. – Springer Berlin Heidelberg, 2011. – С. 1094-1096.
- 74 Stefan D., Shu X., Yao D. D. Robustness of keystroke-dynamics based biometrics against synthetic forgeries //computers & security. – 2012. – Т. 31. – №. 1. – С. 109-121.
- 75 Распознавание: Генетический алгоритм [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2019. — Режим доступа: [http://www.machinelearning.ru/wiki/index.php?title=Генетический\\_алгоритм](http://www.machinelearning.ru/wiki/index.php?title=Генетический_алгоритм). — 08.08.2019.
- 76 Курейчик В. М., Кажаров А. А. Алгоритмы эволюционного роевого интеллекта в решении задачи разбиения графа //Проблемы разработки перспективных микро-и наноэлектронных систем (МЭС). – 2012. – №. 1. – С. 237-242.
- 77 Мурзин Б. П., Светличная В. А. Использование алгоритма муравьиной колонии для определения оптимального маршрута доставки грузов. – 2011.
- 78 Мещеряков Р. В., Ходашинский И. А., Гусакова Е. Н. Оценка информативного признакового пространства для системы обнаружения вторжений //Известия Южного федерального университета. Технические науки. – 2013. – №. 12 (149).
- 79 Kanimozhi M., Kanimozhi A. Implementing Neural Network in Keystroke Dynamics for a Better Biometric Authentication System.
- 80 Yu E., Cho S. GA-SVM wrapper approach for feature subset selection in keystroke dynamics identity verification //Neural Networks, 2003. Proceedings of the International Joint Conference on. – IEEE, 2003. – Т. 3. – С. 2253-2257.
- 81 Yong Z., Dun-wei G., Wan-qiu Z. Feature selection of unreliable data using an improved multi-objective PSO algorithm //Neurocomputing. – 2016. – Т. 171. – С. 1281-1290.

- 82 Abdulkader S. N., Atia A., Mostafa M. S. M. Authentication systems: Principles and threats //Computer and Information Science. – 2015. – Т. 8. – №. 3. – С. 155.
- 83 Zhong Y., Deng Y. A survey on keystroke dynamics biometrics: approaches, advances, and evaluations //Recent Advances in User Authentication Using Keystroke Dynamics Biometrics. Science Gate Publishing. – 2015. – С. 1-22.
- 84 Bours P., Mondal S. Performance evaluation of continuous authentication systems //IET Biometrics. – 2015. – Т. 4. – №. 4. – С. 220-226.
- 85 Alsultan A., Warwick K., Wei H. Improving the performance of free-text keystroke dynamics authentication by fusion //Applied Soft Computing. – 2018. – Т. 70. – С. 1024-1033.
- 86 Giot R. et al. Unconstrained keystroke dynamics authentication with shared secret //computers & security. – 2011. – Т. 30. – №. 6. – С. 427-445.
- 87 Giot R., El-Abed M., Rosenberger C. Greyc keystroke: a benchmark for keystroke dynamics biometric systems //Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on. – IEEE, 2009. – С. 1-6.
- 88 Распознавание: Критерий Колмогорова-Смирнова [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2019. — Режим доступа: [http://www.machinelearning.ru/wiki/index.php?title=Критерий\\_Колмогорова-Смирнова](http://www.machinelearning.ru/wiki/index.php?title=Критерий_Колмогорова-Смирнова). — 08.08.2019.
- 89 Chuda D., Kratky P., Tvarozek J. Mouse Clicks Can Recognize Web Page Visitors! //Proceedings of the 24th International Conference on World Wide Web Companion. – International World Wide Web Conferences Steering Committee, 2015. – С. 21-22.
- 90 Ngai E. W. T. et al. The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature //Decision support systems. – 2011. – Т. 50. – №. 3. – С. 559-569.
- 91 Petrovskiy M., Tsarev D., Pospelova I. Pattern Based Information Retrieval Approach to Discover Extremist Information on the Internet //International Conference on Mining Intelligence and Knowledge Exploration. – Springer, Cham, 2017. – С. 240-249.
- 92 Петровский М.И. Исследование и разработка алгоритмов поиска исключений в системах интеллектуального анализа данных: дис. канд. физ.-мат. наук: 05.13.11. – МГУ им. М.В. Ломоносова, Москва, 2003 - 145 с.
- 93 Ben-Hur A. et al. Support vector clustering //Journal of machine learning research. – 2001. – Т. 2. – №. Dec. – С. 125-137.
- 94 Scholkopf B. et al. Support vector method for novelty detection //Advances in neural information processing systems. – 2000. – С. 582-588

- 95 Hoffmann H. Kernel PCA for novelty detection //Pattern recognition. – 2007. – Т. 40. – №. 3. – С. 863-874.
- 96 Petrovskiy M. A fuzzy kernel-based method for real-time network intrusion detection //International Workshop on Innovative Internet Community Systems. – Springer, Berlin, Heidelberg, 2003. – С. 189-200.
- 97 Bezdek J. C. et al. Local convergence analysis of a grouped variable version of coordinate descent //Journal of Optimization Theory and Applications. – 1987. – Т. 54. – №. 3. – С. 471-477.
- 98 Welch B. L. The generalization of student's' problem when several different population variances are involved //Biometrika. – 1947. – Т. 34. – №. 1/2. – С. 28-35.
- 99 Царев Д.В. Методы и программные средства анализа поведения пользователей при работе с текстовыми данными для решения задач информационной безопасности: дис. канд. физ.-мат. наук: 05.13.11. – МГУ им. М.В. Ломоносова, Москва, 2017 - 143 с.
- 100 Hawkins S. et al. Outlier detection using replicator neural networks //International Conference on Data Warehousing and Knowledge Discovery. – Springer, Berlin, Heidelberg, 2002. – С. 170-180.