МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ М.В.ЛОМОНОСОВА
(МГУ)

№ госрегистрации 114100140107

УТВЕРЖДАЮ
Проректор - начальник
Управления научной политики и
организации научных исследований
МГУ, д.ф.-м.н.
_____ А.А.Федянин
«30» июня 2015 г.

ОТЧЕТ
О ПРИКЛАДНЫХ НАУЧНЫХ ИССЛЕДОВАНИЯХ

Исследование и разработка инновационной технологии построения программных средств обеспечения компьютерной безопасности, основанных на использовании методов машинного обучения и математической статистики для анализа данных поведенческой биометрии пользователей при работе в рамках стандартного человеко-машинного интерфейса, для решения задач активной аутентификации и идентификации пользователей, обнаружения внутренних вторжений и предотвращения попыток хищения конфиденциальной информации

по теме:
ТЕОРЕТИЧЕСКИЕ ИССЛЕДОВАНИЯ (2-ОЙ ОЧЕРЕДИ)
ПОСТАВЛЕННЫХ ПЕРЕД ПНИ ЗАДАЧ

(промежуточный)

Этап 2

ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014-2020 годы»

Соглашение о предоставлении субсидии от 27.06.2014 № 14.604.21.0056

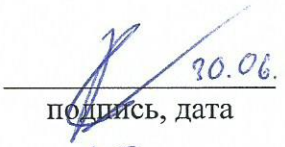
Руководитель проекта,
профессор, д.ф.-м.н. _____ 30.06.2015 _____ И.В.Машечкин
подпись, дата

Москва 2015

СПИСОК ИСПОЛНИТЕЛЕЙ

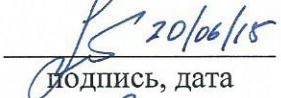
Руководитель проекта
д.ф.-м.н., профессор  И.В. Машечкин (введение, заключение)
подпись, дата

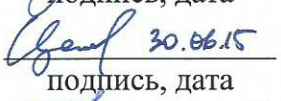
Исполнители темы

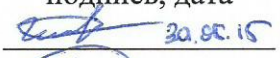
к.ф.м.н., доцент  М.И. Петровский (разделы 4,5, заключение)
подпись, дата

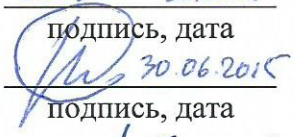
к.ф.м.н., ассистент  В.В. Глазкова (разделы 4, 6)
подпись, дата

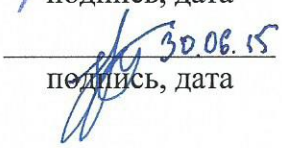
м.н.с.  И.С. Попов (раздел 4, 5,7)
подпись, дата

к.ф.м.н., доцент  А.Н. Терехин (раздел 2,3)
подпись, дата

математик  Д.В. Царёв (введение, раздел 1, 6)
подпись, дата

математик  С.И. Головин (разделы 4, 5)
подпись, дата

программист  А.Ю. Корчагин (введение, раздел 1)
подпись, дата

Нормоконтролер
ассистент  Р.В. Курынин (разделы 2, 3, заключение)
подпись, дата

РЕФЕРАТ

Отчет 193 с., 1 ч., 62 рис., 36 табл., 62 источника, 1 прил.

КОМПЬЮТЕРНАЯ БЕЗОПАСНОСТЬ, ПОВЕДЕНЧЕСКАЯ БИОМЕТРИЯ, АКТИВНАЯ АУТЕНТИФИКАЦИЯ, ИДЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЕЙ, ДИНАМИКА РАБОТЫ С КЛАВИАТУРОЙ И МЫШЬЮ, ОБНАРУЖЕНИЕ ВНУТРЕННИХ ВТОРЖЕНИЙ, ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ, МАШИННОЕ ОБУЧЕНИЕ, МОДЕЛИРОВАНИЕ ПОВЕДЕНИЯ ПОЛЬЗОВАТЕЛЯ.

Объектом исследования являются методы анализа данных поведенческой биометрии пользователей при работе в рамках стандартного человеко-машинного интерфейса, для решения задач компьютерной безопасности, включая задачи активной аутентификации и идентификации пользователей, обнаружения внутренних вторжений и предотвращения попыток хищения конфиденциальной информации.

Цель работы — исследование и разработка комплекса научных решений, направленных на создание программных средств анализа индивидуальных особенностей поведения пользователей компьютерных систем (поведенческой биометрии) при работе в рамках стандартного человеко-машинного интерфейса, с целью создания инновационной технологии построения систем компьютерной безопасности.

В рамках настоящих ПНИ проводились работы, соответствующие второму этапу «Теоретические исследования (2-ой очереди) поставленных перед ПНИ задач».

В отчете содержится информация о проведенных теоретических исследованиях 2-ой очереди по разработке структур данных, методов сбора, предобработки, хранения и управления для поведенческой биометрической информации об особенностях работы со стандартными устройствами ввода-вывода (клавиатура, мышь, монитор) и работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы. Помимо этого, проведена разработка методов машинного обучения и математической статистики для построения и применения поведенческих моделей на основе биометрической информации об особенностях работы со стандартными устройствами ввода-вывода (клавиатура, мышь, монитор) и об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы, а также на основе поведенческой биометрии работы пользователя с текстовыми данными.

СОДЕРЖАНИЕ

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящем отчете о НИР применяют следующие термины с соответствующими определениями.

ТЗ	Техническое задание на выполнение прикладных научных исследований (Приложение 1 к Соглашению № 14.604.21.0056 о предоставлении субсидии от 27.06.2014).
Контент документа	Содержимое документа.
ИБ	Информационная безопасность.
ИАД	Интеллектуальный анализ данных.
ППК	Прикладной программный комплекс.
ЭО ПК	Экспериментальный образец программного комплекса.
ОВБС	Отчет о проведенных работах на этапе № 2 Плана-графика по Соглашению о предоставлении субсидии № 14.604.21.0056 от 27.06.2014г., выполненных за счет внебюджетных средств.

ВВЕДЕНИЕ

КОМПЬЮТЕРНАЯ БЕЗОПАСНОСТЬ, ПОВЕДЕНЧЕСКАЯ БИОМЕТРИЯ, АКТИВНАЯ АУТЕНТИФИКАЦИЯ, ИДЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЕЙ, ДИНАМИКА РАБОТЫ С КЛАВИАТУРОЙ И МЫШЬЮ, ОБНАРУЖЕНИЕ ВНУТРЕННИХ ВТОРЖЕНИЙ, ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ, МАШИННОЕ ОБУЧЕНИЕ, МОДЕЛИРОВАНИЕ ПОВЕДЕНИЯ ПОЛЬЗОВАТЕЛЯ.

В цели настоящего этапа входят теоретические исследования (2-ой очереди) поставленных перед ПНИ задач по аутентификации без использования секретной информации и постоянной фоновой идентификации пользователей, раннего обнаружения внутренних вторжений, а также раннего обнаружения попыток хищения конфиденциальной информации на основе поведенческой биометрической информации об особенностях работы пользователей со стандартными устройствами ввода-вывода, с информационными и вычислительными ресурсами защищаемой компьютерной системы, с текстовой информацией различных типов.

В задачи настоящего этапа исследований входят следующие: разработка структур данных, методов сбора, предобработки, хранения и управления для поведенческой биометрической информации об особенностях работы со стандартными устройствами ввода-вывода (клавиатура, мышь, монитор) и работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы; разработка методов машинного обучения и математической статистики для построения и применения поведенческих моделей на основе биометрической информации об особенностях работы со стандартными устройствами ввода-вывода (клавиатура, мышь, монитор) и об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы, а также на основе поведенческой биометрии работы пользователя с текстовыми данными. Помимо этого, в задачи настоящего этапа входят работы, финансируемые за счет средств внебюджетных источников, по формированию наборов экспериментальных данных.

Работы настоящего этапа проводятся на основе результатов, полученных на предыдущем этапе настоящих ПНИ. В промежуточном отчете по проведенным работам в рамках первого этапа ПНИ «Выбор направления исследований. Теоретические исследования (1-ой очереди) поставленных перед ПНИ задач» [1] содержится информация о проведенных патентных исследованиях и выполненном аналитическом обзоре методов биометрической

аутентификации пользователей с использованием машинного обучения и интеллектуального анализа данных для решения поставленных в ПНИ задач, сделаны выводы о наиболее перспективных и актуальных подходах. Обосновано и выбрано направление исследований моделей представления биометрической информации в части структур и состава данных, методов сбора, предобработки, хранения и управления; обосновано и выбрано направление исследований алгоритмов и методов машинного обучения и математической статистики для построения и применения поведенческих моделей; сформулированы уточненные функциональные требования к реализации программных компонентов. Проведены теоретические исследования 1-ой очереди по разработке структур данных, методов сбора, предобработки, хранения и управления для поведенческой информации об особенностях работы пользователя с текстовой информацией, а также методов языково-независимой предобработки собираемой текстовой информации.

1 Разработка методов машинного обучения и математической статистики для построения и применения поведенческих моделей на основе поведенческой биометрии работы пользователя с текстовыми данными

Согласно требованиям п.3.10 ТЗ в ходе выполнения ПНИ должны быть разработаны методы машинного обучения и математической статистики для построения и применения поведенческих моделей с целью решения задач постоянной фоновой идентификации пользователей и раннего обнаружения попыток хищения конфиденциальной информации на основе поведенческой биометрии работы пользователя с текстовыми данными.

Настоящий раздел посвящён теоретическим исследованиям решения указанных задач на основе проведённого аналитического обзора, осуществлённого выбора направления исследований и предложенного подхода, представленных в отчёте за предыдущий этап настоящих ПНИ (см. подразделы 1.3 и 2.3 в [1], соответственно).

В качестве направления исследований в части возможности построения и применения поведенческих моделей на основе поведенческой биометрии работы пользователя с текстовыми данными был выбран предложенный коллективом авторов подход к анализу работы пользователя с текстовой информацией, основанный на тематическом моделировании (или тематическом анализе) сложившихся в прошлом тенденций работы (поведения) пользователя с текстовым контентом различных категорий (в том числе конфиденциальных). Тематическое моделирование работы пользователя предполагает определение основных тематик его текстового контента и расчёт соответствующих им весов в заданные интервалы времени. На основе предложенных оценок отклонения поведения работы пользователя с контентом от его обычного поведения осуществляется идентификация данного пользователя и раннее обнаружение попыток хищения конфиденциальной информации. Далее в следующих подразделах описаны разработанные методы машинного обучения и математической статистики для решения указанных задач.

1.1 Задача постоянной фоновой идентификации пользователей

Решение *задачи постоянной фоновой идентификации пользователей* заключается в оценке достоверности того, что пользователь, работающий с защищаемой компьютерной системой, является действительно тем, от имени кого он авторизовался. В отличие от аутентификации в настоящей задаче не подразумевается применение явных процедур проверки, требующих интерактивных действий от пользователя. В настоящем подразделе рассматриваются методы машинного обучения и математической статистики для построения и применения поведенческих моделей с целью решения данной задачи на основе данных о работе пользователей с текстовой информацией.

В основе подхода к решению задачи идентификации, предложенного коллективом авторов на этапе выбора направления исследований, лежит тематическое моделирование текстовых данных, с которыми работал пользователь за заданное модельное время, где под модельным временем понимается временной интервал, в течение которого собиралась информация о характерной работе данного пользователя с текстовым контентом. Тематический анализ модельного времени позволяет получать характеристики, описывающие сложившиеся в прошлом (в рамках модельного времени) тенденции работы пользователя с текстовыми данными.

Для применения тематического анализа модельное время разбивается на последовательно измеренные через некоторые промежутки времени интервалы. Например, в качестве промежутка времени (шага) может быть выбран час, день (рисунок 1), а также время, за которое происходит заданное число событий [2, 3]. Таким образом, выбираемые промежутки времени не обязательно должны быть равны между собой. С помощью тематического моделирования, применённого к текстовому контенту временных интервалов модельного времени, выделяются основные тематики пользователя и соответствующие им веса в каждом временном интервале модельного времени.

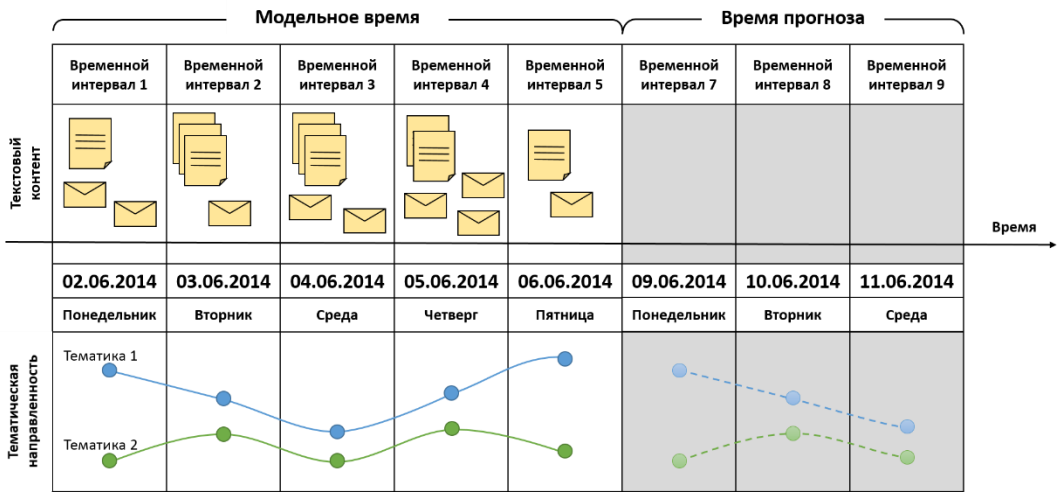


Рисунок 1 — Формирование временных рядов тематической направленности пользователя.

Веса тематик во временном интервале характеризуют тематическую направленность пользователя, на их основе формируются временные ряды изменения его тематической направленности для каждой из тематик (рисунок 1). Далее по сформированным временным рядам строятся прогнозы (рисунок 1). На основе значений отклонений тематической направленности от спрогнозированных данных определяются временные интервалы работы с контентом, не свойственные для данного пользователя. Таким образом, вычисляемые отклонения тематической направленности являются оценками достоверности того, что пользователь, работающий с защищаемой компьютерной системой, является действительно тем, от имени кого он авторизовался, тем самым решается задача идентификации пользователей в постановке, приведённой в начале настоящего подраздела.

Ниже настоящий подраздел организован следующим образом. В пункте 1.1.1 приведено описание процедуры выделения тематических характеристик из текстовых данных пользователя и формирования тематических временных рядов. В пункте 1.1.2 рассматриваются популярные подходы прогнозирования временных рядов и представлен собственный оригинальный метод прогнозирования на основе ортонормированной неотрицательной матричной факторизации. Пункт 1.1.3 посвящен экспериментальному исследованию предложенного подхода идентификации пользователя на примере реальной корпоративной переписки, содержащейся в наборе экспериментальных данных НТЭД1 (см. подраздел 1.3 ОВБС), сформированном на основе набора электронных писем Enron [4].

1.1.1 Выделение тематических характеристик из текстовых данных пользователя

Исходя из выбранного направления исследований (см. подраздел 2.3 в [1]) и работ авторов [3, 5–9] в качестве методов тематического моделирования были выбраны методы, основанные на неотрицательной матричной факторизации. Методы неотрицательной матричной факторизации работают с векторным представлением текста типа «мешок слов» (англ. «bag-of-words») [10]. В нашем случае в качестве текстов выступают текстовые данные каждого временного интервала. Далее под термином временной интервал в зависимости от контекста будет пониматься либо совокупность текстовых данных анализируемого пользователя за рассматриваемый временной интервал, либо непосредственно интервал времени.

Формально опишем модель построения тематических временных рядов для n временных интервалов модельного времени. Каждый временной интервал j ($1 \le j \le n$) отображается в числовой вектор $A_j=[a_{1,j}, a_{2,j}, \ldots, a_{m,j}]^T$ фиксированной размерности m, где m — число признаков текстовых данный за модельное время, а i-ая компонента вектора определяет вес i-го признака в j-ом временном интервале.

В качестве признаков в модели «мешка слов» используются лексемы, входящие в текст, а размерность признакового пространства равна размерности словаря лексем. Под лексемами в общем случае понимаются все различные слова текста. Однако обычно применяются некоторые меры по предварительной обработке текста с целью получения более «информативного» признакового пространства [10]: удаление стоп-слов, приведение слов к нормализованной форме (стемминг) и т.д. Цель предварительной обработки текста — оставить только те признаки, которые наиболее информативны, т.е. наиболее сильно характеризуют текст. К тому же уменьшение анализируемых признаков приводит к уменьшению использования вычислительных ресурсов. В интеллектуальном анализе текстовых данных для обозначения признака текста принято использовать термин «терм».

Вес i-го терма в векторном представлении j-го временного интервала определяется как $a_{i,j}=L_{i,j}G_i$. $L_{i,j}$ — локальный вес терма i во временном интервале j, G_i — глобальный вес терма i во всех временных интервалах. Т.к. для вычисления отклонений от спрогнозированных значений будут использоваться новые временные интервалы, не вошедшие в модельное время, то заранее определить использование того или иного терма в будущих временных интервалах невозможно, поэтому использование глобального веса исключается. В ходе экспериментов, проводимых в пункте 1.1.3, наилучшие результаты

были получены при использовании логарифмического веса в качестве локального: $L_{i,j} = 1 + \log(t_{i,j})$, где $t_{i,j}$ — число появлений терма i во временном интервале j [9, 10].

Таким образом, текстовый контент пользователя за модельное время представляется в виде числовой матрицы, строки которой соответствуют термам, а столбцы текстам каждого временного интервала. Объединение термов в тематики и представление временных интервалов в пространстве тематик осуществляется путём применения к данной матрице неотрицательной матричной факторизации.

Матрица модельных временных интервалов $A \in \Re^{m \times n}$, где m — число различных термов, n — число временных интервалов. Элементы матрицы A принимают неотрицательные значения, т.к. являются весами соответствующих термов во временных интервалах. Тогда цель неотрицательной матричной факторизации состоит в нахождении матриц $W_k \in \Re^{m \times k}$ и $H_k \in \Re^{k \times n}$ с неотрицательными элементами, которые минимизируют целевую функцию (1) [11]:

$$f(W_k, H_k) = \frac{1}{2}\|A - W_k H_k\|_F^2, \; k << min(m,n). \qquad (1)$$

Матрица $W_k=[w_{ij}]$ задает отображение пространства k тематик в пространство m термов, матрица $H_k=[h_{ij}]$ соответствует представлению временных интервалов в пространстве тематик, т.е. элемент h_{ij} соответствует представлению i-ой тематики в j-ом временном интервале. В связи с тем, что элементы матрицы H_k неотрицательны, то их можно рассматривать как вклад (вес) тематики во временной интервал. Чем больше значение элемента h_{ij} по сравнению с другими элементами j-го временного интервала, тем более характерна i-ая тематика для текста данного временного интервала. На этом свойстве основаны алгоритмы кластеризации, использующие неотрицательную матричную факторизацию [12, 13]. Аналогично и для матрицы W_k, чем больше значение элемента w_{ij} по сравнению с другими элементами j-го столбца (j-ой тематике), тем более характерен i-ый терм для данной тематики [14].

Исходя из описанных свойств неотрицательной матричной факторизации, временной ряд изменения каждой из выделенных k тематик формируется из значений элементов соответствующей строки матрицы H_k (рисунок 2).

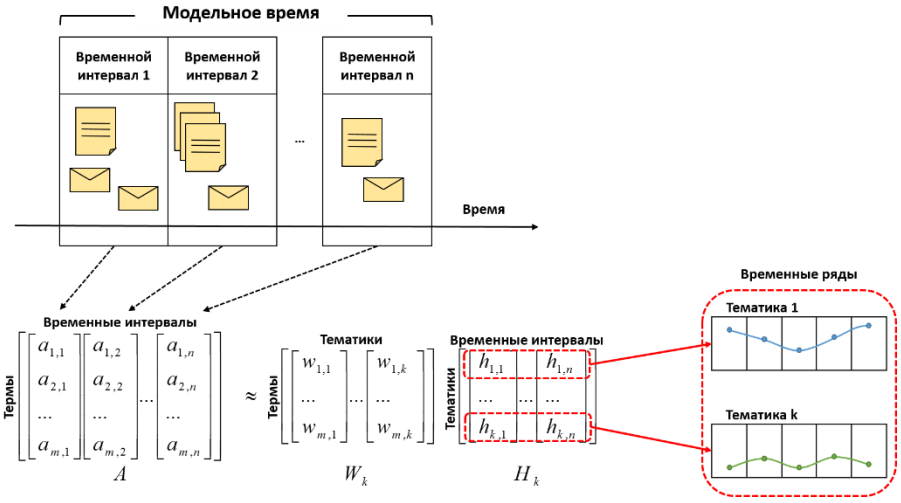


Рисунок 2 — Построение временных рядов на основе неотрицательной матричной факторизации.

Существуют различные методы реализации неотрицательной матричной факторизации [11, 13, 14, 15]. Однако для реализации предлагаемого подхода, необходимо иметь возможность отображать новые, не вошедшие в модельное время, временные интервалы в уже построенное пространство тематик. Для реализации данного функционала необходимо наложить дополнительное условие ортонормированности матрицы W_k: $W_k^T \cdot W_k = I$. Тогда для отображения матрицы временных интервалов времени прогноза A_{new} в пространство тематик модельного времени достаточно A_{new} слева умножить на W_k^T: $W_k^T \cdot A_{new} = H_{k\_new}$.

Авторы исследовали применение множества популярных алгоритмов, реализующих ортонормированную неотрицательную матричную факторизацию [11, 13, 15], для использования в предлагаемом подходе. На основе экспериментальных исследований был выбран алгоритм минимизации целевой функции $f(W_k, H_k) = \frac{1}{2}\|A - W_k H_k\|_F^2 + \frac{\alpha}{2}\|W_k^T W_k - I\|_F^2$, описанный в [11] и позволяющий задавать баланс между точностью приближения исходной матрицы и ортонормированностью получаемых тематик с помощью параметра α.

1. Элементы матриц $W^1 \in \Re_+^{m \times k}$ и $H^1 \in \Re_+^{k \times n}$ инициализируются случайными неотрицательными числами;
2. В цикле *p* раз выполняются итерационные формулы (2), (3) для вычисления матриц *W* и *H*:

   а. $$H_{b,j}^{p+1} = H_{b,j}^{p} \frac{((W^p)^T A)_{b,j}}{((W^p)^T W^p H^p)_{b,j}}, \forall b, j : 1 \le b \le k, 1 \le j \le n \quad , \qquad (2)$$

b. $$W_{i,a}^{p+1} = W_{i,a}^{p} \frac{(A(H^{p+1})^T + \alpha W^p)_{i,a}}{(W^p H^{p+1}(H^{p+1})^T + \alpha W^p (W^p)^T W^p)_{i,a}}, \forall i,a : 1 \le i \le m, 1 \le a \le k \quad . \quad (3)$$

Для построения прогнозов k тематических временных рядов, заданных матрицей H_k, в предлагаемом подходе идентификации пользователя применялись методы, описанные в пункте 1.1.2. После построения прогнозов тематических рядов вычисляются отклонения тематической направленности пользователя за время прогноза от спрогнозированных значений (рисунок 3). Вычисленные отклонения используются для идентификации временных интервалов с несвойственной тематической направленностью пользователя.

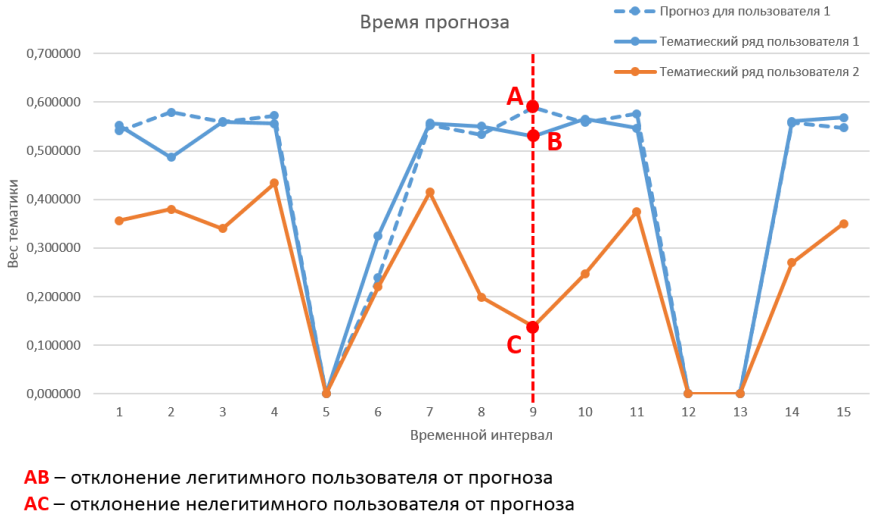


Рисунок 3 — Пример отклонений тематических временных рядов от прогноза.

Отдельно отметим, что матрица W_k, построенная по модельным временным интервалам пользователя, описывает основные тематики пользовательского контента и служит для отображения любых текстовых данных (не обязательно временных интервалов) в пространство тематик данного пользователя, т.е. матрица W_k «характеризует» пользователя с точки зрения его тематических предпочтений в контенте. Поэтому для обозначения матрицы W_k нами также будет использоваться термин тематический «портрет» пользователя. Получаем, что в общем случае временные интервалы одного пользователя можно проецировать в тематическое пространство другого пользователя, и на основе полученных представлений также анализировать временные ряды. Данная возможность позволяет определять временные интервалы, когда один пользователь активно интересовался материалами характерными для другого пользователя. Также возможно сформировать подобную матрицу W_k вообще не на основе временных интервалов пользователей, а на основе заранее сформированного набора документов организации (тренировочный набор),

чьи тематики представляют наибольший интерес для анализа работы пользователей, получив таким образом тематический «портрет» набора документов и уже на его основе строить временные ряды для пользователей.

1.1.2 Применение методов прогнозирования временных рядов для задачи идентификации пользователей

Прогнозирование временных рядов заключается в построении модели для предсказания будущих событий, основываясь на известных событиях прошлого [2]. Для построения прогноза временных рядов использовались следующие модели: линейная модель авторегрессии, авторегрессионная модель дерева и предложенная оригинальная модель на основе ортонормированной неотрицательной матричной факторизации [3].

1.1.2.1 Линейная модель авторегрессии и авторегрессионная модель дерева

В линейной модели авторегрессии временных рядов значение временного ряда в данный момент линейно зависит от предыдущих значений этого же ряда. Формально авторегрессионную модель порядка *p*, которую обычно обозначают, как AR(*p*), определяют следующим образом:

1. $X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t$, где φ_i — параметры авторегрессионной модели, c — константа (для простоты константу, как правило, опускают), ε_t — белый шум;
2. Форма записи с помощью оператора задержки L ($Lx_t = x_{t-1}$): $c + \varepsilon_t = \left(1 - \sum_{i=1}^{p} \varphi_i L_i\right) X_t$.

Авторегрессионная модель дерева (англ. AutoRegressive Tree Model, ART) — модель дерева принятия решений, в «листьях» которого располагаются авторегрессионные модели (AR). Для реализации данной модели в рамках настоящих ПНИ используется алгоритм ARTXP, разработанный Microsoft, который основан на их реализации алгоритма дерева принятия решений. Алгоритм ARTXP устанавливает соотношение между переменным количеством предыдущих элементов и каждым текущим элементом, для которого выполняется прогноз [16].

Отметим основные особенности алгоритма Microsoft ARTXP [17]:

– алгоритм ARTXP поддерживает учёт корреляций между несколькими анализируемыми рядами, т.е. поддерживает перекрестное прогнозирование;

– алгоритм ARTXP используется для прогнозирования ближайших значений временного ряда (порядка 5-7 временных шагов) и даёт существенно менее точный долгосрочный прогноз.

1.1.2.1 Модель прогнозирования на основе ортонормированной неотрицательной матричной факторизации

Анализируемый временной ряд можно представить в виде вектора, элементами которого являются рассчитанные значения в соответствующих временных точках, например, строки матрицы H_k (пункт 1.1.1). Для того чтобы представить вектор временного ряда в матричной форме введём понятие порядка модели, т.е. количество подряд идущих значений временного ряда по которым определяются основные взаимосвязи, аналогично линейной модели авторегрессии. Тогда вектор временного ряда размерности n при заданном порядке модели p можно представить в виде матрицы размерности $p\times(n\text{-}p+1)$, чьи столбцы соответствуют всевозможным подпоследовательностям длины p подряд идущих временных точек анализируемого ряда (рисунок 4).

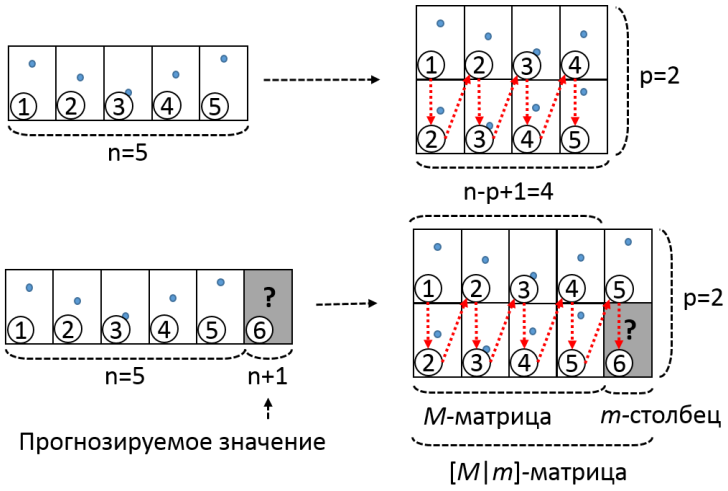


Рисунок 4 — Отображение вектора временного ряда в матрицу.

Задача прогнозирования следующего (n+1)-го значения (рисунок 4) сводится к задаче аппроксимации пропущенных значений в матрице временного ряда (матрица $[M|m]$) на основе уже заполненных значений (матрица M) — так называемая задача подстановки пропущенных значений (англ. Missing Value Imputation) [18–20].

Наиболее распространенным методами матричного разложения к решению задачи подстановки пропущенных значений являются подходы на основе сингулярного разложения (англ. Singular Value Decomposition, SVD) [18–20]. Однако, авторами была исследована

возможность применения методов неотрицательной матричной факторизации для решения задачи подстановки пропущенных значений [18], на основе полученных результатов далее будет описан алгоритм подстановки пропущенных значений с использованием ортонормированной неотрицательной матричной факторизации.

Рассмотрим ортонормированную неотрицательную матричную факторизацию матрицы M: $M \approx M_k = Wm_k \cdot Hm_k$, $Wm_k^T \cdot Wm_k = I$, где M_k является аппроксимацией исходной матрицы M, k — число «латентных» признаков, при этом $k << min(p, n-p+1)$ [8, 18, 21]. Матрица Wm_k задает отображение между пространством «латентных» признаков размерности k и пространством позиций элементов (от 1 до p) в сформированных подпоследовательностях длины p. Таким образом, выделенные «латентные» признаки содержат только наиболее значимую информацию о взаимосвязях между позициями элементов среди всех подпоследовательностей. Каждый столбец матрицы Hm_k описывает соответствующую подпоследовательность в виде вектор-столбца с весами соответствующих базисных «латентных» признаков.

Было предложено вычисление пропущенных значений в матрице $[M|m]$ на основе значений матрицы M следующим итерационным способом (по аналогии с решением в [19]):

- *Шаг 0*. Рассчитать ортонормированную неотрицательную матричную факторизацию матрицы M (с изначально заполненными элементами): $M \approx M_k = Wm_k \cdot Hm_k$, $Wm_k^T \cdot Wm_k = I$.
- *Шаг 1*. Инициализировать пропущенное значения в столбце m. Традиционно в литературе пропущенные значения инициализируются средним значением по столбцу или всему вектору временного ряда. Если известны данные о сезонности временного ряда, то при расчёте средних значений можно учитывать и шаг сезонности. На выходе получается полностью заполненный столбец m^i, где $i = 0$.
- *Шаг 2*. Рассчитать аппроксимацию столбца m с помощью полученной на *шаге 0* модели ортонормированной неотрицательной матричной факторизацию (ONMF-модель): $m_{approx} = (Wm_k \cdot Wm_k^T) \cdot m^i$. После чего сформировать m^{i+1} путем замены пропущенного значения в исходном столбце m соответствующим полученным значением в m_{approx}.
- *Шаг 3*. До тех пор пока значение $\left\|m^i - m^{i+1}\right\| / \left\|m^i\right\|$ меньше заданного порога (как правило, значение порога берут равным 10^{-6}), установить $i = i+1$ и перейти на шаг 2. На практике обычно достаточно 5-6 шагов для сходимости алгоритма.

Заметим, что в приведённом алгоритме вместо ортонормированной неотрицательной матричной факторизации можно использовать и традиционное сингулярное разложение [19].

При прогнозировании многомерных временных рядов можно их прогнозировать как по отдельности, так и с поддержкой перекрестного прогнозирования. Для этого совокупность из m временных рядов нужно объединить в одну матрицу (рисунок 5). Тогда получаемые «латентные» признаки (матрица Wm_k) будут содержать наиболее значимую информацию о взаимосвязях между позициями элементов всех временных рядов среди всех подпоследовательностей, таким образом будет учитываться влияние всех анализируемых временных рядов друг на друга.

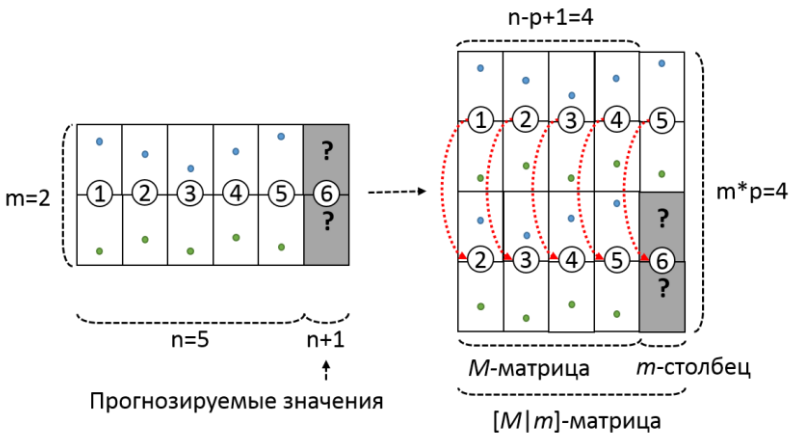


Рисунок 5 — Формирование матрицы прогнозирования для совокупности временных рядов.

1.1.3 Экспериментальные исследования

Первоочередной задачей при проведении экспериментальных исследований предлагаемого подхода идентификации пользователя являлся выбор тестового набора данных. Исходя из формулировки решаемой задачи были предъявлены следующие критерии к набору экспериментальных данных:

– текстовая информация из корпоративной среды;
– возможность сопоставления текстовых данных с пользователями;
– возможность определения времени обработки текстовых данных.

На основе сформулированных выше требований для экспериментального исследования предлагаемого подхода был выбран набор НТЭД1 (см. подраздел 1.3 ОВБС), сформированный на основе набора электронных писем Enron [4].

Для экспериментального исследования предлагаемого подхода из набора НТЭД1 были выбраны три сотрудника с наибольшим количеством писем: «dasovich-j», «kaminski-v» и «kean-s». В качестве ограничения по времени было выбрано первое полугодие 2001 года, т.к. в это время велась самая активная переписка данных пользователей. Выбранные шесть

месяцев были разбиты на пересекающиеся временные интервалы (экспериментальные диапазоны, ЭД) по пять недель с шагом одна неделя, при этом первые четыре недели каждого интервала использовались в качестве модельного времени, а следующая неделя для прогноза. Таким образом всё рассматриваемое время было разбито на 21 экспериментальный диапазон. Каждый ЭД состоит из модельного времени и времени прогноза. В таблице 1 приведена статистика распределения общих писем между выбранными пользователями. Из представленной в таблице 1 статистике распределения общих писем между выбранными пользователями видно, что пользователь «kean-s» имеет существенное количество (более 25%) общих писем с «dasovich-j», что должно сказаться на качестве идентификации.

Таблица 1 — Статистика распределения общих писем между пользователями.

Письма пользователя	Общие письма с пользователем		
	dasovich-j	kaminski-v	kean-s
dasovich-j	11480	56	1730
kaminski-v	16	8757	36
kean-s	2628	91	10043

Для демонстрации предлагаемого подхода идентификации работы пользователя рассматривалась следующая задача бинарной классификации: требуется отделить временные интервалы работы с письмами пользователя «dasovich-j» от временных интервалов работы пользователей «kaminski-v» и «kean-s» на каждой неделе прогноза каждого из 21-го экспериментального диапазона.

Для каждого ЭД производилась следующая процедура, состоящая из 5 шагов:

1. Для каждого пользователя в качестве шага временной точки выбирался не фиксированный шаг времени, а время, за которое пользователь успевал обработать 50 писем. Таким образом, каждая точка временных рядов пользователей представляет конкатенацию из 50 их писем (рисунок 6). Текстовые данные в наборе НТЭД1 являются англоязычными, поэтому для формирования словаря термов использовались такие методы предварительной обработки текста, как удаление стоп-слов и приведение слов к нормализованной форме на основе семантической сети WordNet [22]. Для вычисления весов термов использовался только локальный логарифмический вес. Векторы временных интервалов нормализовались по евклидовой норме.
2. К сформированной матрице модельных временных интервалов (точек) пользователя «dasovich-j» применяется тематическое моделирование на основе ортонормированной

неотрицательной матричной факторизации для получения матрицы «портрета» пользователя (W_k) и матрицы представления временных интервалов в пространстве тематик (H_k). В итоге для пользователя «dasovich-j» получаем k тематических временных рядов для модельного времени (в проводимых экспериментах k=3).

3. Отображение векторов временных точек времени прогноза для всех пользователей осуществлялось с использованием матрицы «портрета» пользователя «dasovich-j» (W_k). Таким образом, были получены реальные тематические данные всех пользователей для временных точек прогноза (рисунок 6).
4. С помощью каждого метода прогнозирования (пункт 1.1.2) строились прогнозы на основе тематических временных рядов пользователя «dasovich-j» за модельное время (рисунок 6). Далее для обозначения методов прогнозирования используются сокращения: метод линейной авторегрессии — AR, метод на основе авторегрессионной модели дерева решений — MS_ARTXP, предложенный метод прогнозирования на основе ортонормированной неотрицательной матричной факторизации — ONMF.
5. Для каждого метода прогнозирования рассчитывалась оценка отклонения каждой временной точки времени прогноза всех пользователей от спрогнозированных значений. В качестве оценки отклонения временной точки от прогноза использовалась *абсолютная оценка* — сумма по всем k тематикам абсолютного отклонения реальных значений весов тематик от спрогнозированных (рисунок 3).

Рисунок 6 — Временные ряды анализируемых пользователей для одной из тематик.

После проведения вышеописанной процедуры с каждым из 21-го ЭД для каждого метода прогнозирования получаем, что всем прогнозируемым временным точкам всех пользователей (96 точек прогноза для каждого пользователя, всего 288) сопоставлены их оценки отклонения.

Фиксируя значение порога допустимого отклонения от прогноза пользователя «dasovich-j» получаем бинарную классификацию для всех прогнозируемых временных интервалов. Для оценки качества классификации обычно используют ROC-кривые — графическая характеристика качества бинарного классификатора, зависимость доли верных положительных классификаций от доли ложных положительных классификаций при варьировании порога решающего правила (оценки отклонения) [23]. Для сравнения нескольких моделей классификации будем использовать значение AUC (англ. Area Under Curve), которое вычисляется как площадь под ROC-кривой и является агрегированной характеристикой качества классификации, не зависящей от соотношения цен ошибок [23]. Чем больше значение AUC, тем «лучше» модель классификации. Полученные ROC-кривые и значения AUC для рассмотренных методов классификации приведены на рисунке 7 и таблице 2.

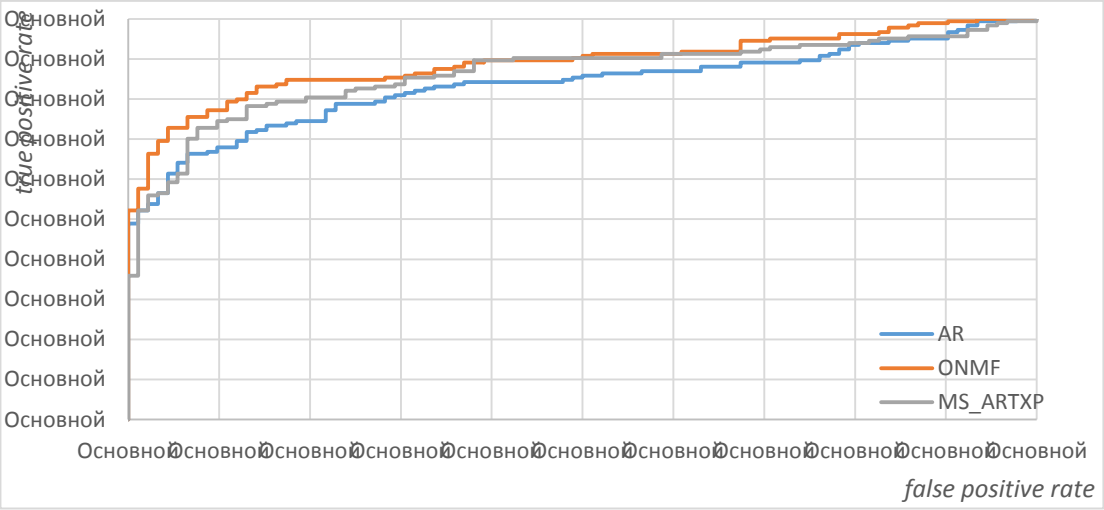


Рисунок 7 — ROC-кривые методов прогнозирования при абсолютной оценки отклонения.

Таблица 2 — Значения AUC для методов прогнозирования и подходов расчёта оценки отклонения.

	AR	MS_ARTXP	ONMF
AUC	0,835539	0,864367	0,889887

Из приведённых данных следует, что:

– Предложенный подход идентификации работы пользователя на основе отклонений его тематической направленности от спрогнозированных данных показывает высокое качество идентификации даже при использовании стандартных методов прогнозирования.

– Разработанный метод прогнозирования временных рядов, основанный на ортонормированной неотрицательной матричной факторизации, показал высокое качество прогнозирования и свою применимость в рассмотренном подходе идентификации работы пользователя.

1.2 Задача раннего обнаружения попыток хищения конфиденциальной информации

Решением *задачи раннего обнаружения попыток хищения конфиденциальной информации* является выявление фактов аномального или подозрительного поведения инсайдеров (авторизованных легальных пользователей или нарушителей, авторизовавшихся под чужим именем), которые могут предшествовать или непосредственно являться частью организации попытки хищения конфиденциальной информации. Это утверждение обусловлено тем, что в большинстве случаев непосредственно хищению конфиденциальной информации предшествует аномальное (хотя возможно и разрешенное) поведение пользователя, т.е. пользователь еще до кражи информации начинает совершать действия, не характерные для его предыдущей активности.

Основное отличие задачи раннего обнаружения попыток хищения конфиденциальной информации от задачи идентификации пользователей заключается в том, что требуется определять не интервалы времени с несвойственной для пользователя работой с текстовыми данными, а непосредственно сами факты работы с документами несвойственного контента. Т.е. каждая точка формируемых тематических временных рядов в задаче идентификации пользователей соответствует *совокупности* текстовых документов (за выбранный интервал времени), а в задаче раннего обнаружения попыток хищения конфиденциальной информации — отдельному документу (другими словами, интервал времени выбирается таким образом, чтобы он содержал ровно один документ).

В связи с указанной спецификой было получено, что зачастую сформированные тематические временные ряды для задачи раннего обнаружения попыток хищения конфиденциальной информации являются плохо прогнозируемыми. Это обусловлено тем, что пользователь последовательно может работать с документами, относящимися к различным тематикам (при этом данные тематики являются характерными для пользователя), вследствие чего при переходе от одной точки временного ряда к последующей возникает сильное изменение тематической направленности (рисунок 8).

Поэтому невозможно применить предложенный в подразделе 1.1 подход идентификации пользователей, в котором используется прогнозирование тематических временных рядов.

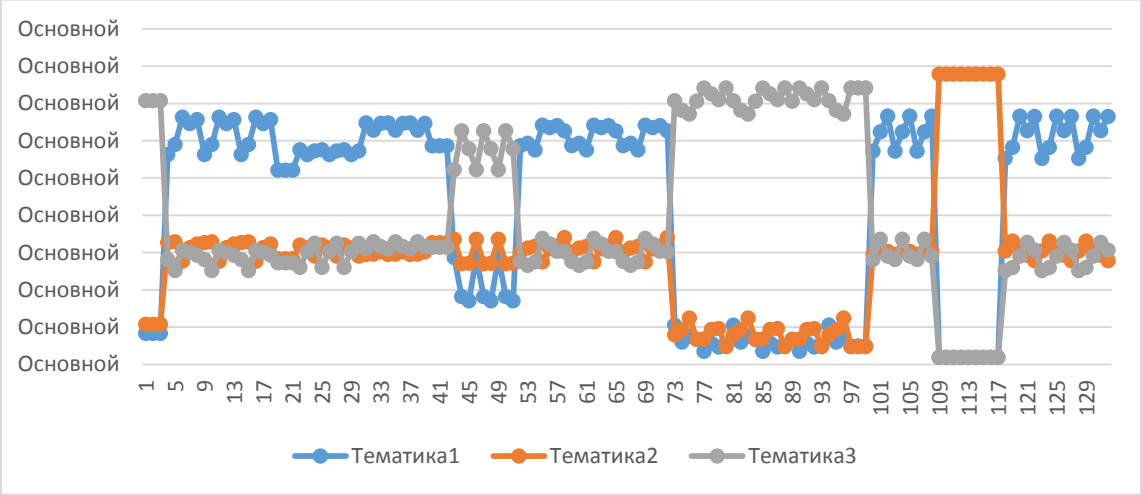


Рисунок 8 — Тематические временные ряды для трёх тематик за модельное время пользователя в задаче раннего обнаружения попыток хищения конфиденциальной информации.

Для решения указанной проблемы была предложена модификация тематической поведенческой модели, используемой в задаче идентификации пользователей, заключающаяся в оценке общего присутствия характерных тематик пользователя (а не тематик по отдельности) в каждой точке временного ряда.

Ниже настоящий подраздел организован следующим образом. В пункте 1.2.1 приведено описание методов построения и применение поведенческой модели с целью решения задачи раннего обнаружения попыток хищения конфиденциальной информации. Пункт 1.2.2 посвящен экспериментальному исследованию предложенного подхода раннего обнаружения попыток хищения конфиденциальной информации на примере работы пользователей с реальными корпоративными документами, содержащимися в наборе экспериментальных данных НТЭД2 (см. подраздел 1.3 ОВБС).

1.2.1 Построение и применение поведенческой модели с целью решения задачи раннего обнаружения попыток хищения конфиденциальной информации

В задаче идентификации пользователей поведенческая модель состоит из совокупности временных рядов и их прогнозов для каждой выделенной тематики. Для решения задачи раннего обнаружения попыток хищения конфиденциальной информации

требуется разработать метод вычисления оценки принадлежности документа (точки временного ряда) ко всем характерным для данного пользователя тематикам.

Аналогично задачи идентификации пользователя первоначально формируется временной ряд для модельного времени пользователя. Каждая точка временного ряда соответствует отдельному документу, обрабатываемому пользователем. Далее к матрице документов временного ряда A применяется тематическое моделирование на основе ортонормированной неотрицательной матричной факторизации (см. пункт 1.1.1). Результатом тематического моделирования являются:

- матрица «портрета» пользователя W_k, требующаяся для отображения новых (не вошедших в модельное время) документов в пространство тематик;
- матрица представления документов модельного времени в пространстве тематик H_k.

Таким образом любой текстовый документ может быть представлен в пространстве тематик пользователя в виде числового вектора $h = [h_1, \ldots, h_k]$, фиксированной размерности k, где k — число выделенных тематик пользователя за модельное время, а i-ая компонента вектора ($1 \leq i \leq k$) определяет вес i-ой тематике в рассматриваемом документе. Тем самым, чем больше элементы вектора h, тем сильнее текст соответствующего документа характеризуется тематиками данного пользователя. Исходя из этого для вычисления общей оценки принадлежности документа к тематикам данного пользователя было предложено использовать норму вектора документа, представленного в пространстве тематик. Были исследованы возможности применение следующих норм вектора:

- Сумма элементов вектора (из-за использования ортонормированной неотрицательной матричной факторизации элементы векторного представления документа в пространстве тематик неотрицательны, поэтому нет необходимости в расчете абсолютной величины (модуля) элементов).
- Евклидова норма.
- Максимум из элементов вектора.

На рисунке 9 продемонстрированы вычисленные оценки принадлежности документов к тематикам анализируемого пользователя для примера тематических временных рядов, представленных на рисунке 8.

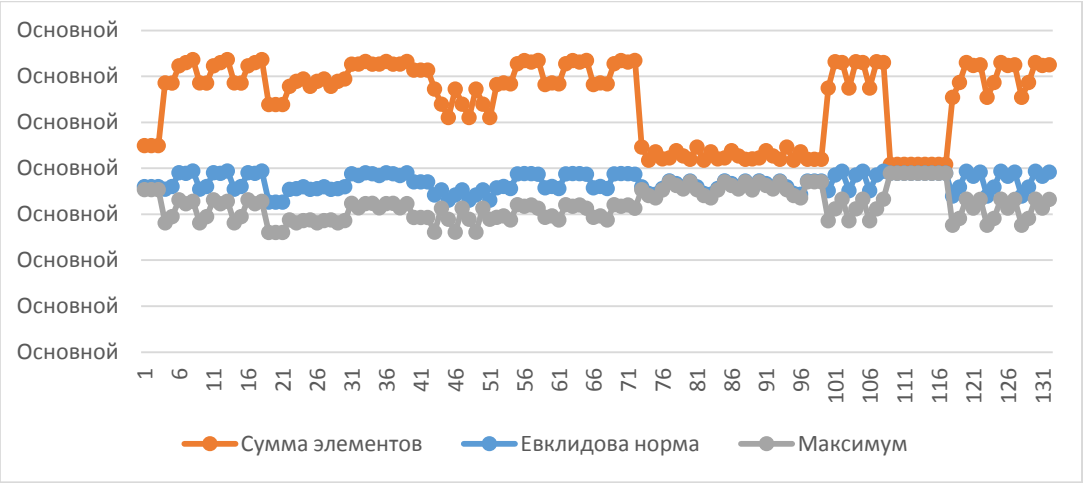


Рисунок 9 — Оценки принадлежности документов к тематикам.

На основе значений оценок принадлежности документов к тематикам определяются документы, не свойственные для данного пользователя. Соответственно, чем ниже оценка документа, тем более несвойственен документ пользователю. Таким образом осуществляется выявление фактов аномального или подозрительного поведения пользователя, которые могут предшествовать или непосредственно являться частью организации попытки хищения конфиденциальной информации.

В предложенном подходе к решению задачи раннего обнаружения попыток хищения конфиденциальной информации так же, как и при решении задачи идентификации пользователей, используется тематическое моделирование для определения тематической направленности пользователя. Однако, вычисление оценки аномальности поведения пользователя при работе с текстовыми данными осуществляется исходя из векторного представления данных в пространстве тематик характерных для пользователя. Тем самым, в отличие от задачи идентификации, не требуется выполнять дополнительную процедуру построения прогнозов дальнейшей тематической направленности пользователя. Кроме того, в предложенном подходе вообще не требуется упорядочивать документы по времени.

1.2.2 Экспериментальные исследования

Первоочередной задачей при проведении экспериментальных исследований предлагаемого подхода раннего обнаружения попыток хищения конфиденциальной информации являлся выбор тестового набора данных. Исходя из формулировки решаемой задачи были предъявлены следующие критерии к набору экспериментальных данных:

– текстовые документы из корпоративной среды;

- возможность сопоставления текстовых документов с пользователями;
- возможность определения времени обработки текстовых документов.

На основе сформулированных выше требований для экспериментального исследования предлагаемого подхода был выбран набор НТЭД2 (см. подраздел 1.3 ОВБС), содержащий текстовые документы, обрабатываемые по средствам электронной почты. За основу данного набора была взята версия набора электронных писем Enron, содержащая прикреплённые к письмам файлы [24].

Для экспериментального исследования предлагаемого подхода использовались все данные набора НТЭД2, т.е. текстовые документы 15 сотрудников за 2000 и 2001 годы. В таблице 3 приведены характеристики набора НТЭД2. Для каждого из 15 пользователей выбранные два года были разбиты на пересекающиеся временные интервалы (экспериментальные диапазоны, ЭД) по пять недель с шагом одна неделя, при этом первые четыре недели каждого интервала использовались в качестве модельного времени, а следующая неделя для прогноза. При формировании экспериментальных диапазонов для каждого пользователя накладывалось ограничение на минимальное число документов, содержащихся в модельном времени данного пользователя. В рамках настоящих исследований требовалась представленность не менее 20-ти документов в модельном времени пользователя. Таким образом было получено 147 экспериментальных диапазонов.

Таблица 3 — Характеристики набора НТЭД2.

Номер	Имя пользователя	Число текстовых документов за 2000 и 2001 года
1	chris_germany	51
2	daren_farmer	261
3	darron_c_giron	40
4	gerald_nemec	2516
5	john_lavorato	108
6	kate_symes	70
7	louise_kitchen	243
8	mark_taylor	1182
9	matthew_lenhart	6
10	phillip_m_love	31
11	richard_sanders	868
12	richard_shapiro	707

Продолжение таблицы 3

13	sally_beck	603
14	sara_shackleton	2926
15	vkaminski	616
Итого	**15 пользователей**	**10228 текстовых документов**

Для демонстрации предлагаемого подхода раннего обнаружения попыток хищения конфиденциальной информации рассматривалась следующая задача бинарной классификации: для документов времени прогноза требуется отделить документы пользователя, для которого строится модель поведения на модельном времени, от документов остальных пользователей.

Для каждого ЭД производилась следующая процедура, состоящая из 4 шагов:

1. Для текущего анализируемого пользователя, т.е. пользователя для которого строится модель поведения (матрица «портрета» пользователя W_k), определяется набор документов модельного времени, с которыми работал пользователь, а затем формируется матричное представление полученного набора. Текстовые данные в наборе НТЭД2 являются англоязычными, поэтому для формирования словаря термов использовались такие методы предварительной обработки текста, как удаление стоп-слов и приведение слов к нормализованной форме на основе семантической сети WordNet [22]. Для вычисления весов термов использовался только локальный логарифмический вес. Векторы временных интервалов нормализовались по евклидовой норме.
2. К сформированной матрице модельных документов пользователя применяется тематическое моделирование на основе ортонормированной неотрицательной матричной факторизации для получения матрицы «портрета» пользователя (W_k) и матрицы представления модельных документов в пространстве тематик (H_k). В проводимых экспериментах число тематик выбиралось равным 10% от размера словаря термов, но не более 10.
3. Отображение документов времени прогноза для всех пользователей в тематическое пространство осуществлялось с использованием матрицы «портрета» анализируемого пользователя W_k (пункт 1.1.1) Таким образом были получены векторные представления классифицируемых документов в пространстве тематик анализируемого пользователя.
4. Для каждой из рассматриваемых норм вектора (пункт 1.2.1) рассчитывалась оценка принадлежности классифицируемых документов к тематикам анализируемого пользователя.

После проведения вышеописанной процедуры для каждого из 147 ЭД получаем, что документам времени прогноза всех пользователей присвоены оценки их принадлежности к тематикам анализируемого пользователя. На основе данных оценок для каждого ЭД было вычислено значение AUC (англ. Area Under Curve), которое является агрегированной характеристикой качества классификации (см. пункт 1.1.3). Таким образом, было получено 147 значений AUC. Для оценки полученного множества значений AUC использовались устойчивые (робастные) оценки центральной тенденции (медиана) и разброса (интерквартильный размах, ИКР) [25, 26]. Интерквартильным размахом (англ. Interquartile Range) называется разность между третьим и первым квартилями множества значений AUC. Полученные значения медиан и интерквартильных размахов для норм вектора приведены в таблице 4.

Таблица 4 — Значения медиан и интерквартильных размахов.

Норма	Медиана	Интерквартильный размах
Сумма элементов	0,835	0,328
Евклидова норма	0,84	0,35
Максимум	0,846	0,314

Из приведённых данных следует, что предложенный подход к решению задачи раннего обнаружения попыток хищения конфиденциальной информации на основе оценки принадлежности документов к тематикам пользователя показывает высокое качество выявление фактов работы пользователя с несвойственными для него документами. Применение различных стандартных норм вектора для расчёта данных оценок привело к схожим высоким результатам. Однако, на используемом наборе данных НТЭД2 применение нормы максимума (вычисление максимального элемента вектора) оказалось наилучшим.

1.3 Выводы

В данном разделе производилась разработка методов машинного обучения и математической статистики для построения и применения поведенческих моделей с целью решения задач постоянной фоновой идентификации пользователей и раннего обнаружения попыток хищения конфиденциальной информации на основе поведенческой биометрии работы пользователя с текстовыми данными.

В качестве направления исследований в части возможности построения и применения поведенческих моделей на основе поведенческой биометрии работы пользователя с текстовыми данными был выбран предложенный коллективом авторов подход к анализу работы пользователя с текстовой информацией, основанный на тематическом моделировании (или тематическом анализе) сложившихся в прошлом тенденций работы (поведения) пользователя с текстовым контентом различных категорий (в том числе конфиденциальных). Тематическое моделирование работы пользователя предполагает определение основных тематик его текстового контента и расчёт соответствующих им весов в заданные интервалы времени. На основе предложенных оценок отклонения поведения работы пользователя с контентом от его обычного поведения осуществляется идентификация данного пользователя и раннее обнаружение попыток хищения конфиденциальной информации.

Для решения задачи постоянной фоновой идентификации пользователей были разработаны следующие методы:

- метод расчёта оценки отклонения тематической направленности пользователя от спрогнозированных данных, который показал высокое качество идентификации даже при использовании стандартных методов прогнозирования;
- метод прогнозирования временных рядов, основанный на ортонормированной неотрицательной матричной факторизации, который показал высокое качество прогнозирования и свою применимость в рассмотренном подходе идентификации работы пользователя.

Для решения задачи раннего обнаружения попыток хищения конфиденциальной информации был разработан метод расчёта оценки принадлежности текстовых данных к характерным тематикам пользователя, который показал высокое качество выявление фактов работы пользователя с несвойственными для него документами.

Проведённые экспериментальные исследования предложенных методов для решения задач постоянной фоновой идентификации пользователей и раннего обнаружения попыток хищения конфиденциальной информации на сформированных наборах экспериментальных данных НТЭД1 и НТЭД2 подтвердили полученные выводы.

Результаты, полученные в рамках настоящего раздела, подтверждают соответствие требованиям пп. 2.1.7, 3.10, 4.1.1.2 подпункт 3) ТЗ.

На основе полученных результатов в ходе проведения работ настоящего раздела были подготовлены следующие публикации:

- I.V. Mashechkin, M.I. Petrovskiy, D.S. Popov, D.V. Tsarev. Applying text mining methods for data loss prevention. Programming and Computer Software, 41(1):23–30, 2015. DOI: 10.1134/S0361768815010041. (Связь со статьей приводится в пункте 1.1.1).
- И.В. Машечкин, М.И. Петровский, Д.В. Царёв. Применение методов интеллектуального анализа текстовой информации для предотвращения утечек данных. *Программирование*, (1):32–43, 2015. (Связь со статьей приводится в пункте 1.1.1).
- В.Ю. Королев, А.Ю. Корчагин, И.В. Машечкин, М.И. Петровский, Д.В. Царёв. Применение временных рядов в задаче фоновой идентификации пользователей на основе анализа их работы с текстовыми данными. Труды Института системного программирования РАН (электронный журнал), 27(1):151–172, 2015. DOI: 10.15514/ISPRAS-2015-27(1)-8. (Связь со статьей приводится в подразделе 1.1).

2 Разработка структур данных, методов сбора, предобработки, хранения и управления для поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы

Согласно п.3.6 ТЗ должны быть разработаны структуры данных, методы сбора, предобработки, хранения и управления для поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы.

У коллектива авторов имеется научный задел в области разработки систем сбора, обработки и анализа информации из разнородных источников [27, 28, 29].

При решении указанной задачи необходимо также учитывать требования ТЗ к реализации методов сбора, предобработки, хранения и управления для поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы, указанные в п.4.1.2.3.1 и предписывающие обеспечение сбора и обработки следующих типов фактов работы пользователей:

- факты работы с локальными файлами;
- факты работы с внешними носителями;
- факты работы с разделяемыми сетевыми ресурсами;
- факты работы с удаленными сервисами и приложениями по сетевым протоколам TCP/IP;
- факты входа и выхода в/из системы;
- факты локального и удаленного запуска и установки приложений и сервисов;
- факты изменения программной конфигурации системы;
- факты подключения дополнительного оборудования.

Таким образом, необходимо решить задачу обеспечения наблюдения (или мониторинга) за действиями пользователей и процессов в рамках операционной системы. Как было показано в аналитическом обзоре, представленном в отчёте за предыдущий этап настоящих ПНИ (см. подраздел 1.2 в [1]), традиционно решение данной задачи строится на

использовании технологии «управления информацией и событиями безопасности» (Security Information and Event Management, SIEM), которая, в свою очередь, базируется на сборе и обработке элементарных событий, фиксируемых в ходе деятельности пользователей/процессов в журналах (logs) различных приложений и операционных систем, работающих в рамках компьютерной системы. Многие современные операционные системы обладают штатной системой аудита, позволяющей унифицированным способом собирать данные о действиях пользователей. Достоинствами такого метода являются простота механизмов сбора (нет необходимости в написании драйверов низкого уровня), возможность тонкой настройки аудита на конкретные события и информационные объекты (можно обойтись без написания фильтров событий). Недостатками встроенной подсистемы сбора аудита является большое количество типов событий, поступающих из разнородных источников, недостаточно полная информация о работе с объектами, находящимися на других машинах, сложность механизмов перекодировки и анализа событий штатного аудита. Кроме того, с помощью штатной системы аудита могут возникать сложности со сбором данных о действиях пользователей, обладающими административными правами, поскольку штатный аудит может быть ими отключен. С учетом указанных недостатков возникает потребность в расширении штатного функционала системы аудита, речь о котором пойдет ниже в настоящем разделе.

Не ограничивая общности рассуждений и принимая во внимание требование п.4.1.2.9 ТЗ дальнейшая разработка структур данных, методов сбора, предобработки, хранения и управления для поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы будет вестись применительно к автоматизированным местам пользователей (АРМ), функционирующим под управлением ОС семейства Microsoft Windows.

Для удовлетворения выделенных требований в рамках ОС Microsoft Windows информацию о запуске процессов, работе с реестром, сеансах работы на АРМ, установке программ, а также информацию об отказах и ошибках предлагается собирать с помощью встроенного аудита ОС, так как информация данного типа описывается штатными средствами достаточно подробно. Проблема уничтожения данных штатного аудита может быть решена, во-первых, упреждающим чтением вновь собранных данных, предшествующим потенциальному удалению этих данных, а во-вторых, контролем включенности аудита на АРМ. Для сбора остальных типов активности предлагается

разработать собственный функционал мониторинга, так как стандартный аудит для данного вида активности либо не существует, либо не полон.

Существует несколько подходов к анализу собранной информации: на АРМ пользователя, централизованно, или же на основе комбинации двух подходов. Коллективом участников предлагается схема с использованием централизованной обработки, поскольку ее необходимость обусловлена требованиями п.4.1.2.4.2 ТЗ, а также следующими факторами: записи о фактах активности одного и того же объекта могут быть расположены на разных АРМ; одним из вариантов анализа является сравнение поведения одного объекта с поведением других объектов того же типа. Консолидация данных и их централизованный анализ позволяют производить агрегацию и корреляцию (см. раздел 1.2 в [1]), что существенно повышает качество и «наглядность» результатов анализа. Наличие централизованной консолидации данных позволяет обеспечить дополнительный функционал, такой как сбор общей базы событий, что может быть использовано для расследования происшествий или отложенного дополнительного анализа. Предлагаемый подход несет с собой и ряд проблем, связанных с производительностью и защищенностью, типичных для централизованных систем. Поэтому встает вопрос в обеспечении эффективности механизмов консолидации данных и средств их защиты.

С другой стороны, выше отмечались недостатки штатной системы аудита деятельности пользователей, поэтому возникает задача предобработки элементарных событий. Так, собранные элементарные события не всегда описывают законченные действия пользователей, не всегда описывают работу с каждым из ресурсов по отдельности и не всегда содержат необходимые параметры для обработки. Для анализа параметров работы пользователей с ресурсами представляет интерес не только отдельные события, но и их множества и последовательности, описывающее некоторое совокупное действие, например, формирование процесса и суммарный объем данных, переданный этим процессом. Таким образом, требуется проведение агрегации элементарных событий в новые более «информативные» события.

Также в некоторых случаях для улучшения качества и наглядности анализа необходимо уточнять или дополнять набор атрибутов собираемых событий. Например, заменять системные идентификаторы пользователей именем пользователя, определять имена родительских процессов, заменять IP-адреса на имена хостов, а номера портов на их общепринятые названия и т.п.

Важной задачей является фильтрация собираемых значений. Задача фильтрации состоит в отсечении ненужных событий, и, если производить фильтрацию на стороне АРМ,

это позволяет существенно снизить нагрузку на механизмы централизованной консолидации. Фильтрация событий выполняется непосредственно после чтения событий из журнала, что позволяет минимизировать загрузку АРМ, так как ненужные события отбрасываются сразу и не участвуют на последующих этапах обработки.

Таким образом, под предварительной обработкой событий будет пониматься:

- фильтрация прочитанных из журналов событий данных с целью отсечения ненужных для анализа записей журналов;
- расширение набора атрибутов прочитанных событий с целью уточнения характеристик событий значениями, которые могут быть получены только на АРМ.

Подводя промежуточные итоги, можно выделить следующий круг технологических задач:

- мониторинг данных;
- предобработка данных;
- консолидация данных.

Как отмечалось в разделе 3 в [1], для решения подобного класса задач управления информацией из множества различных источников [30] обычно применяется мультиагентный подход, заключающийся в установке *программных агентов* на каждый источник. *Программный агент* — это автономный процесс, способный реагировать на среду исполнения и вызывать изменения в среде исполнения, возможно, в кооперации с пользователями или другими агентами. Свойство, которое делает агента чем-то большим, чем процесс, — это способность функционировать автономно, агент сам контролирует свои действия.

В случае мониторинга работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы, мультиагентная система должна выполнять следующие задачи:

- Собственно, *мониторинг работы с информационными и вычислительными ресурсами*. Получение элементарных событий из различных источников (журналов).
- *Предварительная обработка событий*. Фильтрация событий с последующей нормализацией и унификацией (преобразование представления событий безопасности в унифицированный формат).
- *Сохранение предобработанных событий* в защищенном локальном хранилище.

– *Консолидация* собранных и предобработанных *данных* путем передачи их в центральное хранилище (которое также можно рассматривать в качестве специального агента — *сервера консолидации*).

По существу, агенты на АРМ пользователей выполняют сбор, обработку и передачу данных, такие агенты в [30] отнесены к типу *информационных* — управление информацией из множества различных источников, в том числе и физически разных. В дальнейшем мы будем называть эти агенты *агентами сбора*. Учитывая требования п.4.1.2.3.1 ТЗ агент сбора должен осуществлять сбор и обработку данных в фоновом режиме, т.е. основными требованиям к нему являются производительность и «незаметность» для пользователя АРМ.

В задачи *сервера консолидации* входит получение информации от агентов сбора и ее размещение в специализированном хранилище, а также обеспечение управления настройками агентов сбора. Основной особенностью работы сервера консолидации является необходимость параллельного получения данных от большого количества агентов (тысячи и даже десятки тысяч), что объясняется масштабами современных сетей. Поэтому возникает необходимость реализации:

– эффективного представления данных на агентах сбора для их последующей передачи на сервер консолидации;
– механизмов распределения нагрузки на сеть (стратегий передачи данных).

Ниже в настоящем разделе на примере агента сбора и сервера консолидации будут разработаны структуры данных, методы сбора, предобработки, хранения и управления для поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы.

2.1 Мониторинг работы с информационными и вычислительными ресурсами

Для обеспечения сбора и обработки всех типов фактов, указанных выше, задача мониторинга работы с информационными и вычислительными ресурсами композиционно разбивается на следующие:

– мониторинг на основе штатных средств аудита;
– мониторинг файловой системы;
– мониторинг работы с сетью;
– мониторинг установки и удаления технических средств;

– сбор записей журналов;
– контроль аудита на АРМ пользователей.

2.1.1 Мониторинг на основе штатных средств аудита

В качестве основы сбора данных использовались журналы ОС Microsoft Windows. Детальную информацию об организации системных журналов ОС Microsoft Windows можно найти в [31]. Основные журналы и идентификаторы событий в операционных системах Microsoft Windows одинаковы. Для фиксации событий безопасности используется журнал Security («Безопасность»), из которого предлагается извлекать события, указанные в таблице 5.

Таблица 5 — Обрабатываемые идентификаторы событий журнала Security.

Идентификатор события	Описание
512	запуск операционной системы Windows
513	запуск выключения Windows
528	успешное начало сеанса работы
538	завершение сеанса работы
551	начало выполнения завершения сеанса работы
592	создание нового процесса
593	окончание работы процесса

В журнале System («Система») собираются системные события, из которых предлагается использовать события с идентификаторами, приведенными в таблице 6.

Таблица 6 — Обрабатываемые идентификаторы событий журнала System.

Идентификатор события	Описание
6005	событие запуска Event log

В журнале Application («Приложение») собираются события прикладного уровня, из которых предлагается использовать события с идентификаторами, приведенными в таблице 7.

Таблица 7 — Обрабатываемые идентификаторы событий журнала Application.

Идентификатор события	Описание
1033	установка программы
1034	удаление программы

Указанные во всех трех журналах типы событий предлагается использовать в качестве исходных данных для формирования фактов работы пользователей с информационными и вычислительными ресурсами защищаемой компьютерной системы на следующем шаге по предобработки этих исходных данных (см. подраздел 2.2).

2.1.2 Мониторинг файловой системы

Мониторинг файловой системы предлагается осуществлять по аналогии с подходом, изложенным в подразделе 3.1 в [1].

В ОС Windows используются специальные структуры данных ядра, называемые IRP-пакетами (англ. *I/O Request Packet* — пакет запроса ввода/вывода), для обеспечения обмена данными между приложениями и драйвером, а также между драйвером и драйвером. Таким образом, обращение к файлам — это фактически формирование соответствующих IRP и посылка их драйверам файловой системы [32]. Операции быстрого ввода/вывода (англ. *Fast I/O*), специально предназначенные для быстрого синхронного ввода/вывода в кэшируемых файлах, мы не учитываем, т.к. они служат для передачи данных непосредственно между пользовательскими буферами и системным кэшем в обход файловой системы и стеков драйверов устройств [33].

Фильтрация IRP — это общий и универсальный механизм, его используют при разработке антивирусов, файловых архиваторов, файлового шифрования и т.д. Для реализации фильтрации IRP есть документированные возможности — написание драйвера и присоединение его к стеку драйверов файловой системы. Начиная с Windows XP SP2, возможно написание драйверов – *минифильтров* ФС [34], предназначенных специально для мониторинга (и фильтрации) IRP-пакетов ФС. Важной особенностью минифильтров является поддержка двунаправленного небуферизированного канала обмена сообщениями между драйвером и приложениями пользовательского режима, в качестве которых обычно используют службы Windows [35]. Общий механизм мониторинга IRP изображён на рисунке 10.

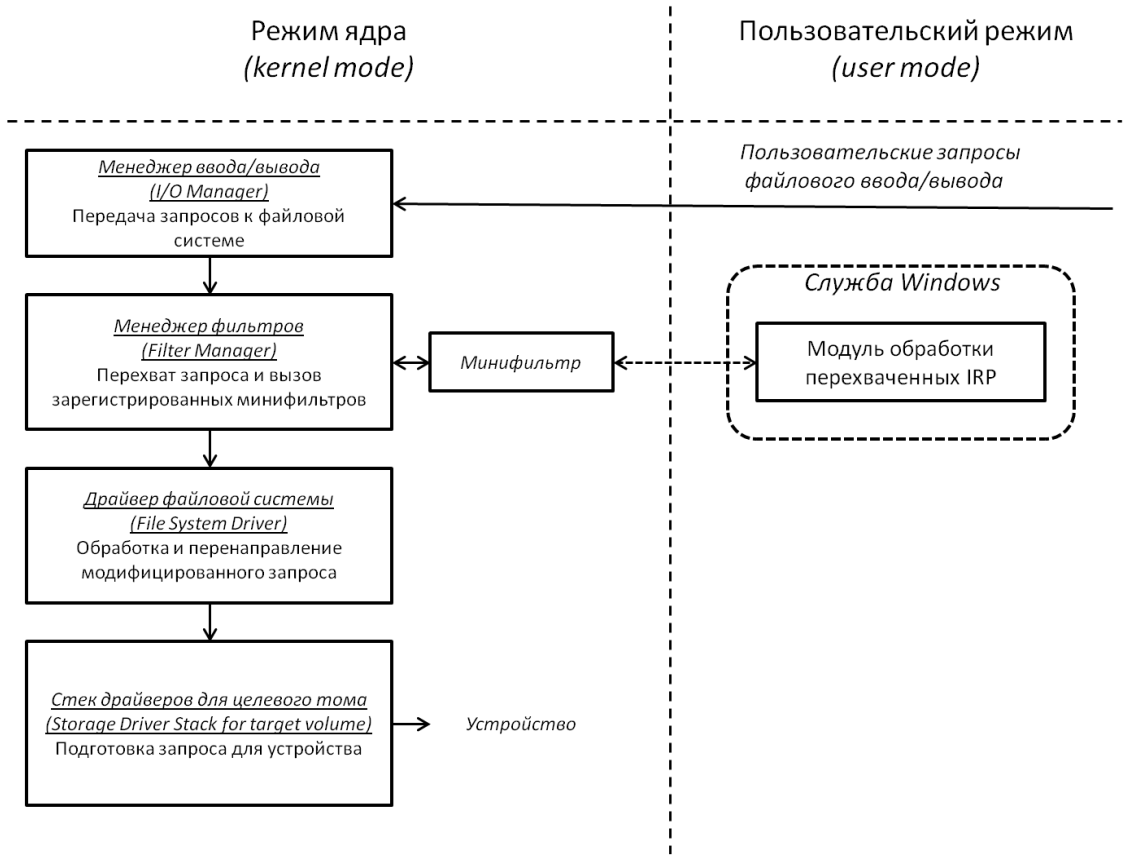


Рисунок 10 — Мониторинг IRP [1].

Для мониторинга файловой системы требуется разработать драйвер-минифильтр ФС. Для сбора аудита необходимо выполнять обработку, как минимум, IRP-пакетов IRP_MJ_CREATE, IRP_MJ_CLEANUP, IRP_MJ_CLOSE, IRP_MJ_READ, IRP_MJ_WRITE, IRP_MJ_SET_INFORMATION.

2.1.3 Мониторинг работы с сетью

Для сбора параметров работы с сетью используется технология LSP (Layered Service Provider, англ. многоуровневый поставщик услуг) — технология Windows sockets версии 2.0, позволяющая пользователю подключать собственные DLL-библиотеки для обработки вызовов Winsock API. Суть технологии состоит в том, что любое обращение к Winsock API будет передано по цепочке всем зарегистрированным модулям LSP. Каждый из этих модулей может модифицировать принимаемые/передаваемые данные и/или адреса, либо просто получить параметры запроса. Технология LSP обычно используется для подсчёта и ограничения трафика, антивирусной защиты, регулирования скорости закачки и приоритетов, а также для организации фильтрации контента.

Для использования технологии LSP с целью подсчета трафика требуется создать специальную динамически подгружаемую библиотеку (DLL), которую затем

зарегистрировать в качестве модуля LSP. В задачи данной DLL будет входить подсчет параметров сетевого обращения и сохранение собранных данных в журнал операционной системы, откуда данные будут прочитаны агентом сбора.

2.1.4 Мониторинг установки и удаления технических средств

В ОС Microsoft Windows предусмотрен набор API для работы с установленными в операционной системе устройствами. Данный набор включает функции для получения текущего списка устройств, для получения имен и других данных устройств, а также функции, позволяющие отслеживать изменение конфигурации. Мониторинг фактов изменения состава устройств предлагается реализовать в агенте сбора на основе существующих в API методов. Для этого необходимо рассмотреть два случая:

– конфигурация обновляется, когда агент выключен (например, выключенное АРМ);
– конфигурация изменяется, во время работы агента.

Для решения задачи отслеживая изменения конфигурации, произошедшего в момент, когда агент был выключен, реализуется следующий алгоритм:

1. Агент в момент старта получает с помощью соответствующих методов текущий список устройств, установленных в операционной системе.
2. Агент считывает из файла записанный ранее список устройств и сравнивает его со вновь созданным списком, все найденные изменения заносятся в системный журнал. Если файла со списком устройств нет, то ничего не заносится.
3. Агент записывает в файл новый список устройств и продолжает свою работу.
4. Агент при обнаружении изменения состава устройств определяет и записывает в файл новый список устройств.

Таким образом, во время своей работы агент поддерживает актуальный список устройств, а если конфигурация была изменена на выключенном АРМ, то агент обнаружит данный факт при включении.

Для задачи мониторинга изменения конфигурации АРМ во время работы агента такая схема не подойдет, так как, во-первых, задача определения списка всех устройств требовательна к ресурсам, а, во-вторых, если периодически опрашивать списки устройств, возможна ситуация, когда устройство будет добавлено и удалено между циклами опроса, т.е. факт подключения не будет обнаружен. Для решения задачи осуществляется «подписывание» на сообщения операционной системы об изменении аппаратной конфигурации. В момент прихода сообщения об изменении конфигурации в системном

журнале фиксируется произведенная операция и имя устройства, и обновляется файл со списком текущей конфигурации.

2.1.4.1 Получение текущей конфигурации

Первая задача, которую необходимо решить, — это получение списка устройств. Устройства подразделяются на классы: например, класс видеоустройств, принтеров, модемов, клавиатур и т.д. Любое устройство должно принадлежать какому-либо классу. Каждый класс идентифицируется своим GUID-ом (глобальный уникальный идентификатор). GUID — это 128-битная запись типа: {C06136A2-43EA-4F43-AF06-7413D07E28B7}. Для получения полного списка устройств сначала надо получить список классов. Для получения списка классов используется функция `CM_Enumerate_Classes`:

```
CMAPI CONFIGRET WINAPI
  CM_Enumerate_Classes(
    IN ULONG ulClassIndex,// индекс класса
    OUT LPGUID ClassGuid,// указатель GUID класса
    IN ULONG ulFlags //не используется
    );
```

Для перечисления всех значений требуется в цикле вызывать функцию, начиная с индекса 0. Если функция вернула значение `CR_NO_SUCH_VALUE`, значит, список завершен. Вторым параметром должен быть указатель на переменную `TGUID`, в которую будет сохранён GUID класса. Получение информации о классе осуществляет функция `SetupDiGetClassDescription`:

```
WINSETUPAPI BOOL WINAPI
  SetupDiGetClassDescription(
    IN LPGUID ClassGuid,//GUID класса
    OUT PTSTR ClassDescription,//строка
    IN DWORD ClassDescriptionSize,//размер строки
    OUT PDWORD RequiredSize OPTIONAL//требуемый размер
    );
```

Вторым параметром должен идти указатель на буфер, в который будет сохранена строка с именем класса. Третьим параметром должен идти размер передаваемого буфера. Если указанного буфера не хватит, то требуемый размер будет сохранён в переменной, указатель на которую передается четвёртым параметром.

После получения списка классов требуется получить список устройств, принадлежащих некоторому классу. Для этого используется функция SetupDiGetClassDevs:

```
HDEVINFO
  SetupDiGetClassDevs(
    IN LPGUID ClassGuid, OPTIONAL
    IN PCTSTR Enumerator, OPTIONAL
    IN HWND hwndParent, OPTIONAL
    IN DWORD Flags
    );
```

Первый параметр задаёт класс устройств для их перечисления. Если этот параметр равен нулю, то перечисляться будут все устройства. Второй и третий параметры могут быть равны нулю. Последний параметр может принимать следующие значения:

- DIGCF_ALLCLASSES — будет возвращён список всех устройств и всех классов, установленных в данный момент в операционной системе. Первый параметр будет проигнорирован.
- DIGCF_PRESENT — будет возвращён список устройств, которые в настоящее время присутствуют в операционной системе.

Требуется указать только класс устройств и указать последним параметром флаг DIGCF_PRESENT. Схема получения сначала списка класса устройств, а затем устройств внутри класса позволяет в будущем наложить фильтр на рассматриваемые классы устройств, таким образом, что ресурсов, требуемых для получения полного списка устройств, будет требоваться пропорционально множеству классов.

Имея список устройств, требуется перечислить все устройства, находящиеся в нём. Для этого используется функция SetupDiEnumDeviceInfo:

```
WINSETUPAPI BOOL WINAPI
  SetupDiEnumDeviceInfo(
    IN HDEVINFO DeviceInfoSet,
    IN DWORD MemberIndex,
    OUT PSP_DEVINFO_DATA DeviceInfoData
    );
```

Второй параметр задаёт индекс в списке. Третий параметр — это указатель на структуру SP_DEVINFO_DATA, в которой будет сохранена информация об устройстве. Если

функция вернула значение TRUE, то информация извлечена успешно, а если FALSE, то это означает что достигнут конец списка. Для перечисления всего списка требуется в цикле вызывать функцию SetupDiEnumDeviceInfo, каждый раз увеличивая значение индекса на единицу до тех пор, пока не получится отрицательный результат.

Структура, в которой хранится информация об устройстве, выглядит следующим образом:

```
typedef struct _SP_DEVINFO_DATA {
  DWORD cbSize;
  GUID ClassGuid;
  DWORD DevInst;
  ULONG_PTR Reserved;
} SP_DEVINFO_DATA, *PSP_DEVINFO_DATA;
```

Основным является поле DevInst, которое хранит «указатель» устройства. Для того чтобы получить имя устройства (или его описание), следует использовать функцию SetupDiGetDeviceRegistryProperty:

```
WINSETUPAPI BOOL WINAPI
  SetupDiGetDeviceRegistryProperty(
    IN HDEVINFO DeviceInfoSet,
    IN PSP_DEVINFO_DATA DeviceInfoData,
    IN DWORD Property,
    OUT PDWORD PropertyRegDataType, OPTIONAL
    OUT PBYTE PropertyBuffer,
    IN DWORD PropertyBufferSize,
    OUT PDWORD RequiredSize OPTIONAL
    );
```

Второй параметр — указатель на структуру SP_DEVINFO_DATA. Третий параметр задаёт тип информации, которую требуется получить, важны два флага: SPDRP_FRIENDLYNAME и SPDRP_DEVICEDESC. Далее идёт опциональный параметр, который задаёт указатель на переменную. В указанной переменной будет сохранён тип данных ключа реестра, из которого была извлечена информация. Далее идёт ещё три параметра, которые задают, соответственно, указатель на буфер для сохранения информации, размер буфера и размер реально скопированных данных в него. Если использовать флаг SPDRP_FRIENDLYNAME, то вернется вместо модели жёсткого диска «дисковый накопитель», а при использовании флага SPDRP_DEVICEDESC вернется модель

жёсткого диска. Не всегда информация для обоих параметров представлена, иногда есть только для SPDRP_FRIENDLYNAME, а иногда есть только для SPDRP_DEVICEDESC. Если при использовании первого флага возвращается пустая строка, то требуется получить информацию с использованием второго флага.

2.1.4.2 Мониторинг изменения конфигурации АРМ

Каждый раз, когда происходят какие-либо изменения в аппаратном профиле, главному окну приложения посылается сообщение WM_DEVICECHANGE. Вместе с сообщением в операционной системе передаются два параметра: WParam и LParam, уточняющие параметры сообщения. При получении этого сообщения WParam содержит код события. Требуется рассмотреть только три кода: DBT_DEVICEARRIVAL, DBT_DEVICEREMOVECOMPLETE и DBT_DEVNODES_CHANGED.

Событие DBT_DEVNODES_CHANGED обозначает, что произошли изменения в аппаратном профиле. LParam в данном случае равен нулю. События DBT_DEVICEARRIVAL и DBT_DEVICEREMOVECOMPLETE идентичны и различаются тем, что первое событие обозначает присоединение устройства, а второе — отсоединение устройства. LParam отличен от нуля при этих событиях и указывает на структуру DEV_BROADCAST_HDR. В зависимости от поля dbch_devicetype в этой структуре дальнейшие поля могут варьироваться. Например, если dbch_devicetype равен DBT_DEVTYP_VOLUME, то LParam в этом случае указывает на структуру DEV_BROADCAST_VOLUME и поле dbcv_unitmask в этой структуре содержит битовую маску новых дисков (нулевой бит обозначает букву A, второй — букву B, третий — букву C, и так далее).

Для отслеживания изменения конфигурации требуется корректно обработать пришедшее событие. Для этого требуется получить «указатель» на устройство и его имя, для этого используется метод и функции, описанные выше.

Для того чтобы «подписаться» на сообщения операционной системы, надо вызвать функцию RegisterDeviceNotification:

```
HDEVNOTIFY WINAPI RegisterDeviceNotification(
  __in HANDLE hRecipient,
  __in LPVOID NotificationFilter,
  __in DWORD Flags
);
```

Первый параметр — это «указатель» статуса сервиса либо «указатель» формы. Параметр NotificationFilter является указателем на структуру DEV_BROADCAST_HDR и задаёт тип устройств для отслеживания. Для отслеживания всех устройств поле dbch_devicetype должно быть равно значению DBT_DEVTYP_DEVICEINTERFACE. Если третий параметр равен DEVICE_NOTIFY_WINDOW_HANDLE, то первый параметр должен быть «указателем» окна, если DEVICE_NOTIFY_SERVICE_HANDLE, то первый параметр — это «указатель» статуса сервиса. Также для получения сообщений об изменении всех классов устройств этот параметр должен включать флаг DEVICE_NOTIFY_ALL_INTERFACE_CLASSES.

2.1.5 Сбор записей журналов

Для сбора данных журналов ОС Microsoft Windows в агенте сбора предлагается реализовать отдельные нити для каждого обрабатываемого журнала — это позволит осуществлять независимое чтение и обработку вновь записанных в журнал событий. При этом чтение вновь появившейся записи будет происходить в режиме, приближенном к реальному времени. Основные шаги алгоритма цикла чтения следующие:

1. Резервное копирование справочников (речь о них пойдет ниже в подразделе 2.3).
2. Обновление конфигурационных настроек чтения журнала.
3. Считывание из журнала новой информации.
4. Фильтрация вновь прочитанной информации.
5. Дополнительная обработка информации.
6. Запись полученных событий в локальный буфер агента для отправки.
7. Обновление идентификатора последнего прочитанного события.
8. Проверка необходимости отправки данных на сервер консолидации, и в случае необходимости — осуществление отправки.

Для синхронизации последнего прочитанного события предлагается использовать уникальный в рамках журнала идентификатор события, который есть у любого события в журнале регистрации ОС Microsoft Windows. Данный идентификатор должен сохраняться после каждого цикла опроса журнала агентом и использоваться при его старте для синхронизации последнего прочитанного события.

Чтение событий из журнала регистрации предлагается осуществлять с помощью стандартного API операционной системы, в частности, с помощью функции

ReadEventLog, описанной в заголовочном файле Events.h. Прототип функции ReadEventLog выглядит следующим образом:

```
BOOL ReadEventLog(
  HANDLE hEventLog, // дескриптор журнала
  DWORD dwReadFlags, // флаги
  DWORD dwRecordOffset, // номер события, с которого
требуется начать чтение
  LPVOID lpBuffer, // указатель на буфер для прочитанных
событий
  DWORD nNumberOfBytesToRead, // максимальный размер данных
в байтах, которые требуется прочитать
  DWORD* pnBytesRead, // указатель на переменную, куда
запишется количество прочитанных байт
  DWORD* pnMinNumberOfBytesNeeded // указатель на переменную,
куда будет помещен размер следующего события в логе
);
```

При использовании данного метода следует обратить внимание на время его работы. Метод может использоваться как для чтения одного события, так и для чтения группы событий. В случае вызова чтения в момент, когда в журнале находится много непрочитанных событий, что может быть, например, при первом запуске агента, данный метод вызывается один раз для чтения группы событий. Вызов данного метода для чтения каждого отдельного события может быть расточительным и вносить дополнительную нагрузку на АРМ. Количество событий, читаемых за один вызов метода ReadEventLog определяется параметрами nNumberOfBytesToRead и lpBuffer. Принимая во внимание, что агент может работать в режиме отложенной передачи данных на сервер (например, раз в час — см. подраздел 2.4 ниже), предлагается, чтобы события всегда считывались из журнала группами, раз в единицу времени. Промежуток между циклами чтения данных из журнала предлагается сделать параметром работы агента и установить в диапазоне нескольких секунд (например, 1–15 секунд).

2.1.6 Контроль аудита на АРМ пользователей

По умолчанию в ОС Microsoft Windows включен не полный список аудита событий, поэтому необходима настройка политики аудита, которая описывает, в частности, какие из

требуемых событий будут фиксироваться в журналах. Результирующая политика аудита на каждом конкретном АРМ складывается из *глобальной политики*, действующей в рамках группы АРМ или всего домена и определенной на контроллере домена, и *локальной политики*, установленной на конкретном АРМ. Причем глобальная и локальная политики могут либо дополнять друг друга, либо доменная политика может перекрывать локальную политику на конкретном АРМ. Для настройки результирующего аудита на каждом АРМ настраиваются требуемые параметры локальной политики и отключаются их перекрытие на контроллере домена (значение по умолчанию), так как включать аудит для всего домена может быть некорректно, особенно в случае установки агентов лишь на небольшое подмножество АРМ домена.

Для настройки локальной политики аудита (например, в ходе установки агента на АРМ) требуется вызвать стандартную утилиту ОС Windows secedit.exe, на вход которой подается специальный конфигурационный файл, описывающий, аудит каких событий должен быть включен или выключен. Данный конфигурационный файл можно включить в дистрибутив агента, чтобы настройки аудита применялись в момент установки агента.

Несанкционированное отключение аудита и, как следствие, изменение результирующей политики можно выявлять с помощью диагностического инструментария, который предлагается реализовать в рамках функционала агента сбора по проверке его статуса работы.

2.2 Предварительная обработка событий

Как было заявлено выше, под предварительной обработкой событий понимается:

- фильтрация прочитанных из журналов событий данных с целью отсечения ненужных для анализа записей журналов;
- расширение набора атрибутов прочитанных событий с целью уточнения характеристик событий значениями, которые могут быть получены только на АРМ.

2.2.1 Фильтрация данных

Задача фильтрации состоит в отсечении ненужных событий, которая, согласно предлагаемому подходу, будет осуществляться на агенте сбора, что позволит существенно снизить нагрузку на механизмы консолидации. Прочитанные значения всех атрибутов всех событий всех журналов и модулей сбора параметров работы пользователей должны проверяться на выполнение условий фильтрации, и только в случае соответствия значений

фильтрам, элементарное событие будет обработано. Фильтрацию предлагается организовать в два этапа: во-первых, из журнала будут считываться события только требуемых типов; во-вторых, для каждого атрибута требуемых типов событий будут применяться два списка фильтров: для пропуска и для блокирования событий. Фильтрация значений атрибутов предлагается организовать как проверку на соответствие или несоответствие регулярным выражениям. Если хотя бы один атрибут события не соответствует ни одному из требуемых регулярных выражений из списка пропуска, то всё событие целиком игнорируется. То же происходит, когда хотя бы один атрибут соответствует хотя бы одному регулярному выражению из списка блокирования. Предложенный подход позволяет описать как события, которые требуется собирать, так и события, которые требуется пропустить, при этом предлагаемые механизмы фильтрации обладают достаточной гибкостью, так как проводятся на уровне самой малой составляющей событий — атрибутов.

Для реализации гибкой фильтрации на основе регулярных выражений для каждого атрибута каждого события необходимо:

1. Поддерживать список источников данных (журналов, источников данных в рамках журнала и типов событий в рамках каждого источника данных), которые требуется собирать.
2. Поддерживать список регулярных выражений для каждого атрибута каждого события, каждого журнала, с учетом флага пропуска/блокировки — т.е. либо пропускаются события, у которых атрибут удовлетворяет регулярному выражению, либо пропускать события, у которых атрибут, наоборот, не удовлетворяет выражению.
3. Проверять выполнение всех необходимых регулярных выражений после чтения каждого нового события. Обрабатывать событие дальше только в случае выполнения всех условий. Отсутствие условия на значение атрибута эквивалентно выполнению всех условий для атрибута.

Поддержку и хранение набора журналов, типов событий и регулярных выражений для фильтрации предлагается реализовать в виде конфигурационного файла (например, на основе XML), который можно распространять вместе с агентом. В случае изменения настроек и фильтров событий для АРМ на агент достаточно передать обновленный файл настроек.

Последовательность проверок условий фильтрации может влиять на загрузку конечного узла. Например, если на атрибуты какого-то события наложено значительное количество условий, то проверку их выполнения имеет смысл производить до первого невыполняющегося условия. А условие фильтрации, которое чаще всего срабатывает,

проверять первым. Тогда возникает требование упорядоченности проверки условий, которая может быть задана в конфигурационном файле, при этом указанный порядок следования условий задает эксперт при создании файла.

2.2.2 Расширение набора атрибутов событий

2.2.2.1 Определение имен родительских процессов

Стандартные средства аудита запуска процессов в ОС Microsoft Windows обладают несколькими особенностями:

- событие запуска процесса среди своих атрибутов содержит идентификатор родительского процесса, но не содержит его имени;
- не для всех запущенный процессов присутствует соответствующая запись в журнале (например, запуск процесса «SYSTEM» не фиксируется).

Таким образом, для получения имени родительского процесса требуется просматривать журнал в обратном направлении с целью поиска предшествующего события запуска процесса с идентификатором процесса, равным требуемому идентификатору родительского процесса, а затем получить имя процесса из найденной записи журнала. Такая схема не будет работать в случае, если запуск родительского процесса не был зафиксирован в журнале, или если журнал был очищен, либо события о запуске процесса уже затерлись новыми событиями (например, в случае, когда журнал имеет фиксированный объем и перезаписывается циклически). Также следует отметить, что процесс просмотра журнала в обратном направлении и поиска требуемого события может быть слишком ресурсоемким, что недопустимо.

Решить проблему получения имени родительского процесса с помощью вызова соответствующих функций операционной системы также не всегда возможно, так как в момент чтения из журнала события о запуске процесса, данного процесса уже может не существовать.

Для решения задачи определения имен родительских процессов предлагается использовать модификацию первого подхода. Во-первых, дополнить стандартные средства аудита запуска процессов ОС Windows путем фиксации в журнале событий запуска всех процессов, включая системные. Во-вторых, заменить поиск соответствующего события запуска родительского процесса в журнале поиском в буфере процессов, активных на момент наступления прочитанного события. Буфер активных процессов, в свою очередь, предлагается поддерживать на агенте сбора.

Мониторинг фактов запуска процессов

Для определения родительских процессов требуется фиксировать идентификатор процесса, имя процесса, а также время запуска, которое требуется для упорядочивания процессов. При этом если фиксировать текущие процессы в памяти агента, то может возникнуть следующая ситуация: после перезапуска АРМ в журнале остались еще необработанные с предыдущей сессии записи, в частности, это могут быть записи о запуске процессов, а буфер активных процессов уже утерян. Такая ситуация приведет к потере информации о родительских процессах. Таким образом, активные процессы требуется фиксировать и вне оперативной памяти. Активные процессы фиксируются также в журнале, дополнительно это позволяет унифицировать обработку сообщений, как стандартных, так и реализованных дополнительно. Для определения имен всех запущенных процессов, идентификаторов и времен используется стандартный набор функций API ОС Windows.

Буфер активных процессов

Основная задача поддержания буфера процессов заключается в предоставлении набора процессов, активных на момент возникновения очередного события, прочитанного из журнала. Важно заметить, что очередное прочитанное из журнала событие могло возникнуть сколь угодно давно, т.е. возможно после него операционная система даже была перезагружена.

Для реализации и использования буфера активных процессов используется схема, основанная на выполнении следующих задач при запуске агента:

1. Вызывается функция или процесс, записывающий информацию обо всех активных на момент вызова процессов в журнал Activity. События должны быть записаны в порядке, отсортированном по времени и по очередности появления в операционной системе.
2. Формируется список процессов, активных на момент первого необработанного агентом события. Для этого определяется дата первого необработанного агентом события. Находится предшествующее ей событие запуска журнала аудита (EventLog Start up) в журнале System, что соответствует запуску операционной системы. Просматриваются журналы Security и Activity, начиная с найденной даты до даты первого необработанного агентом события, с целью восстановления множества активных процессов. Восстановление происходит следующим образом: при обнаружении факта запуска процесса, если процесса с таким id еще не было зарегистрировано, то он добавляется в таблицу, если уже был зарегистрирован, то запись о нем обновляется. При обнаружении

события завершения процесса запись с требуемым идентификатором удаляется из таблицы.

После выполнения указанных задач, в обычную работу агента по считыванию следующего события из журнала Security вносятся следующие коррективы:

1. При запуске агента из журнала Activity считываются все события, описывающие запуск процессов с датой, большей даты текущего необработанного агентом события.
2. Агент в цикле находит и считывает ближайшее по дате событие в журнале Security и списке считанных из журнала Activity событий. Если даты следующего события в журнале Security и списке событий совпадают, предпочтение отдается списку, так как может быть порождено несколько процессов в один момент времени, один из которых является родительским другому, а в списке информация изначально более полная и отсортирована по очередности появления процессов.
3. В случае считывания из списка, прочитанное событие удаляется из списка.
4. Выполняются стандартные действия по обработке события — в зависимости от типа прочитанного события вызывается соответствующий типу события сценарий обработки.

Стандартные действия по обработке прочитанных событий запуска и завершения процессов дополняются следующими действиями. При считывании события запуска процесса среди активных процессов по идентификатору родительского процесса находится имя родительского процесса. К считанному событию добавляется новый атрибут с именем родительского процесса. Информация о факте запуска процесса добавляется во множество активных процессов: если запись с таким идентификатором уже существовала, то информация обновляется. При считывании события завершения процесса из множества активных процессов удаляется информация о завершенном процессе.

2.2.2.2 Определение продолжительности сеансов работы и процессов

Продолжительность сеансов работы пользователей на АРМ и работы программ не фиксируются стандартными средствами аудита ОС Microsoft Windows. Для получения этих данных требуется определить два события (начала и окончания) и вычислять разницу времен их наступления. Определение продолжительности сеансов работы и процессов предлагается реализовать с помощью модификации алгоритма определения родительских процессов, путем добавления следующих шагов:

1. Аналогично буферу активных процессов поддерживать буфер активных сеансов пользователей.

2. При считывании события завершения сеанса работы или завершения процесса, в момент нахождения в соответствующем буфере события начала дополнительно вычислять *продолжительность*, дописывать ее как новый атрибут к событию начала, и передавать на сервер консолидации расширенное событие начала.

Следует заметить, что данный подход приведет к появлению двух событий для сеанса работы пользователя и двух событий для факта запуска процесса. Передавать только второе событие (с известной продолжительностью) неправильно, так как оно может быть сформировано только после завершения сеанса или процесса. Для некоторых видов анализа требуется получать события как можно быстрее, пусть и не с полным набором атрибутов. В результате на сервере консолидации первыми будут появляться события начала сеанса или процесса с продолжительностью равной нулю, затем через некоторое время они будут обновляться до событий, у которых продолжительность уже определена.

2.2.2.3 Определение имен удаленных хостов и портов для фактов сетевого взаимодействия

Методы определения имени хоста по IP-адресу и удаленного порта по идентификатору порта предлагается реализовать на основе использования стандартной функции `getnameinfo` из API операционной системы, описанной в заголовочном файле wspiapi.h.

Рассматриваемые преобразования с большой вероятностью будут повторяются: например, один и тот же IP-адрес будет часто заменяется на один и тот же хост. Принимая во внимание тот факт, что данные преобразования могут быть ресурсоемкими, предлагается использовать буферизацию уже выполненных преобразований. Буфер преобразованных значений следует делать общим для многих процессов, поэтому преобразование значений и поддержание буфера предлагается реализовывать унифицированным образом на агенте сбора. После считывания и фильтрации очередного события из журнала выполняются все необходимые преобразования события.

2.2.3 Итоговый набор предобработанных событий

На основе описанных механизмов сбора и предобработки элементарных событий с учетом требований пп. 4.1.2.3.1 и 4.1.2.3.2 ТЗ предлагается собирать и обрабатывать следующий набор фактов и атрибутов работы пользователей с информационными и вычислительными ресурсами защищаемой компьютерной системы:

– Факты работы пользователей с системой.
    - Доменное имя пользователя.
    - Сетевое имя компьютера.
    - Время начала сеанса работы пользователя.
    - Тип сеанса.
    - Запущенный процесс.
    - Продолжительность сеанса.
– Факты запуска процессов и приложений.
    - Доменное имя пользователя.
    - Сетевое имя компьютера.
    - Время запуска процесса.
    - Имя процесса.
    - Имя родительского процесса.
    - Продолжительность работы.
– Факты сетевых TCP-соединений.
    - Доменное имя пользователя.
    - Сетевое имя компьютера.
    - Дата/время передачи.
    - Имя процесса.
    - Локальный порт.
    - Удаленный порт.
    - Удаленный адрес.
    - Удаленный хост.
    - Тип протокола.
    - Переданный объем.
    - Полученный объем.
– Факты работы с файловой системой, включая внешние запоминающие устройства на основе ФС, а также файлы и папки на удаленных компьютерах.
    - Доменное имя пользователя.
    - Сетевое имя компьютера.
    - Дата/время обращения.
    - Имя процесса.
    - Тип устройства.

- Тип операции.
- Имя файла.
- Прочитанный объем.
- Записанный объем.

– Факты обращения к программе Windows Installer (стандартный компонент ОС Windows, отвечающий за изменение набора установленных программ).
  - Доменное имя пользователя.
  - Сетевое имя компьютера.
  - Дата/время события.
  - Тип операции.
  - Название программы.
– Факты изменения аппаратной конфигурации, свидетельствующего об установке, удалении или модификации состава технических средств.
  - Доменное имя пользователя.
  - Сетевое имя компьютера.
  - Дата/время события.
  - Тип операции.
  - Класс устройства.
  - Название устройства.
  - Описание устройства.

2.3 Сохранение предобработанных событий

Как было сказано выше, компоненты мультиагентной системы должны отвечать необходимым требованиям производительности. В частности, накладываются ограничения на объем собранных данных с целью минимизации сетевого трафика и объема хранилища данных. Иными словами, сбор и хранение данных в том виде, как они представлены в журналах регистрации, равно как и в любом текстовом формате (например, XML) невозможен. Решения, основанные на использовании СУБД для хранения собранных данных, не удовлетворяют поставленным требованиям производительности.

Для представления записей журналов регистрации, как на агенте сбора, так и на сервере консолидации предлагается использовать хранилище на основе файловой системы. Файлы такого хранилища делятся на три информационные части:

– *Основные параметры*, присутствующие во всех записях всех журналов: тип события, время генерации, имя пользователя и т.п.
– *Вспомогательные параметры*, описывающие дополнительную информацию о событии.
– *Справочники* для хранения реальных строковых значений как основных, так и вспомогательных параметров.

Запись о событии из журнала разделяется между двумя базовыми файлами и файлами справочников (см. рисунок 11): файл для хранения идентификаторов основных параметров, файл для хранения идентификаторов вспомогательных параметров и файлы справочников. Файлы справочников содержат только одну копию каждого значения и позволяют по идентификаторам (значениям хэш-функции) параметров получать текстовые значения. Три справочника необходимы для хранения имен пользователя, имен параметров и всех остальных значений, соответственно. Разделение на три справочника вместо одного общего оправдывается выигрышем производительности: время поиска значения в справочнике возрастает с размером справочника. Имя пользователя и имена параметров требуются для анализа каждой записи журнала регистрации, поэтому количество обращений к справочникам именно за этими значениями велико.

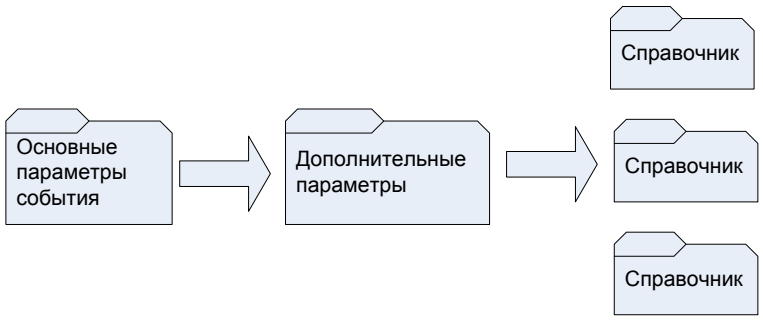


Рисунок 11 — Основные файлы.

Для сопоставления идентификатора параметра его реальному значению в справочнике предлагается использовать механизм хэширования. При этом, в силу возникновений потенциальных коллизий по хэшу, требуется для каждого значения хранить некий параметр, с помощью которого вычислялся хэш. Таким образом, получаем структуру записи справочника, представленной на рисунке 12.

Параметр вычисления hash функции	Размер значения в байтах	Значение

Рисунок 12 — Формат записи справочника.

Для справочника предлагается определить следующие логические операции: загрузка справочника из файла; добавление значения в справочник; получение по значению хэш-функции исходной строки.

Загрузка справочника заключается в последовательном прочтении всех записей справочника, вычислении для каждого прочитанного значения записи хэш-функции с использованием параметра для вычисления хэш-функции и заполнении отображения вычисленных значений хэш-функции в смещение в файле, указывающее на начало прочитанной записи. Таким образом, в памяти будут храниться только значения хэш-функции и смещения в файле справочника, реальное значение будет подгружаться из файла только в случае необходимости.

Добавление значения в справочник описывается следующим алгоритмом:

1. Основной сценарий.
   1.1. Для значения, используя стандартный параметр, вычисляется хэш-функция.
   1.2. Если вычисленное значение хэш-функции не найдено в уже загруженном справочнике, то оно добавляется, а в файл справочника дописывает тройка: «параметр для вычисления хэш-функции, длина добавляемой строки, значение строки».
2. Альтернативный сценарий.
   2.1. Если полученное значение хэш-функции уже присутствует в загруженном справочнике, то из файла справочника считывается реальное значение, которое сравнивается посимвольно с добавляемым значением. В случае совпадения, считается, что значение добавлено.
   2.2. Если считанное из файла значение не совпало с добавляемым, то псевдослучайным образом обновляется параметр для вычисления хэш-функции и итерация добавления в справочник повторяется.

Получение по значению хэш-функции исходной строки реализуется следующим алгоритмом:

1. Основной сценарий.
   1.1. В уже загруженном справочнике проверяется наличие переданного значения хэш-функции.
   1.2. Если значение найдено, то по нему из отображения восстанавливается смешение записи в файле справочника, происходит непосредственное считывание значения из файла.
2. Альтернативный сценарий.

2.1. Если значение хеш-функции не обнаружено в уже загруженном справочнике, считается, что такого значения в справочнике нет.

Предлагается реализовать следующие бинарные форматы базовых файлов для хранения данных, представленные на рисунках 13 и 14.

Идентификатор записи	Хэш значения источника события	Идентификатор события windows	Время генерации события	Время записи события в журнал регистрации	Хэш значения имени журнала регистрации	Хэш значения имени компьютера	Хэш значения имени пользователя	Количество дополнительных параметров

Рисунок 13 — Формат записи файла основных параметров.

Идентификатор записи	Хэш значения имени типа данных	Хэш значения имени параметра	Хэш текстового значения параметра	Целочисленное значение параметра

Рисунок 14 — Формат записи файла вспомогательных параметров.

В предлагаемом подходе каждую запись журнала регистрации будет описывать одна запись файла для хранения основных параметров и несколько записей файла для хранения дополнительных параметров. Добавление записи журнала регистрации будет происходить по следующему алгоритму:

1. Для каждого параметра события, не входящего в набор основных параметров, добавляется запись в файл для хранения дополнительных параметров. Счетчик дополнительных параметров увеличивается на 1.
2. Добавляется запись в файл для хранения основных параметров.

При этом добавление каждой записи будет сопровождаться добавлением необходимых значений в справочники. В качестве значений хэш-функций будут добавляться значения, вычисляемые при добавлении в справочники. Для восстановления события тогда требуется произвести действия в обратном порядке.

Для хранения данных на сервере консолидации предлагается использовать тот же подход с расширением организации хранения файлов событий и справочников в файловой системе. Для этого предлагается строить хранилище по схеме, приведенной на рисунке 15. Например, хранилище может быть организованно в виде дерева, в корне которого находятся каталоги с именем домена, внутри каждого каталога с именем домена находятся каталоги с именем АРМ этого домена. Внутри каталогов с именем АРМ находятся каталоги для

журналов регистрации каждого АРМ, внутри каталога журнала регистрации находятся файлы с собранными записями, разделенные по датам, например, по дням. Файлы-справочники могут храниться в произвольном месте на диске.

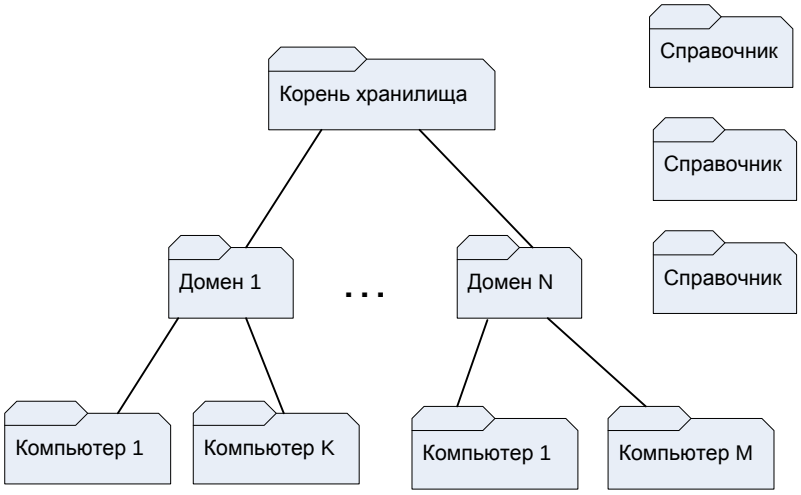


Рисунок 15 — Пример организации хранилища на сервере консолидации.

Приведенная иерархия каталогов (домен – имя АРМ – журнал – дата) не очень принципиальна, однако является достаточно наглядной с точки зрения организации хранени данных, позволяет простым способом получать, например, все события из домена или с определенных АРМ, поскольку обычно для дальнейшего анализа данных требуется получить из хранилища необходимый временной срез и дополнительно применить к нему фильтр (обычно по именам АРМ и типам журналов).

Хранилище данных на агенте сбора и на сервере консолидации предлагается реализовать на основе файловой системы NTFS — это позволит обеспечить защиту доступа к данным с помощью расстановки прав доступа к соответствующим файлам и каталогам.

2.4 Консолидация данных

В рамках агента сбора и сервера консолидации должен быть реализован функционал по организации передачи собранных на агенте данных серверу консолидации. Этот механизм подразумевает передачу больших объемов собранной информации по сети. Так как существует ряд сетей, где скорость передачи невелика, а объем трафика имеет значение (например, при передаче данных из регионального филиала в центральный офис), актуальным является наличие методов сжатия передаваемых данных и планирования передачи данных. Также требуются механизмы защиты данных, передаваемых по сети, с

целью гарантирования достоверности собранных и обрабатываемых данных, а также минимизации рисков утечки информации.

Для обеспечения распределения нагрузки на сеть и балансировки нагрузки на наблюдаемые компьютеры предложено организовывать передачу собранных данных по одной или нескольким из следующих стратегий:

- *Фиксированными объемами данных*. Агент накапливает определенный объем информации или фиксированное количество записей журналов, а затем передает их на сервер консолидации.
- *Через равные промежутки времени*. Агент через равные промежутки времени передает все имеющиеся у него в локальном хранилище данные независимо от их объема.
- *Немедленная передача*. Агент немедленно передает данные о каждой вновь прочитанной записи в журнале регистрации. Данная стратегия наиболее требовательная к ресурсам компьютера.

Указанный механизм регулируется заданием для каждого журнала регистрации следующих параметров:

- максимально возможный объем непереданных данных,
- максимально возможное количество непереданных записей журнала регистрации,
- максимально возможный интервал времени, в течение которого агент может не передавать данные.

Как только одно из максимально возможных значений достигнуто, все собранные данные помещаются в очередь на отправку.

Для обеспечения безопасности все передаваемые по сети данные должны шифроваться с помощью криптографического протокола SSL (Secure Sockets Layer). Предлагается применяться двустороннюю авторизацию для невозможности подмены принимающей или передающей стороны. Протокол обеспечивает конфиденциальность обмена данными между клиентом и сервером, использующими TCP/IP, для шифрования предлагается использовать асимметричный алгоритм с открытым ключом [36].

2.5 Выводы

В данном разделе были проведены теоретические исследования по разработке структур данных, методов сбора, предобработки, хранения и управления для поведенческой информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы.

С учетом специфики рассматриваемых проблем и требований ТЗ была проведена декомпозиция задачи на следующие: мониторинг работы пользователей с информационными и вычислительными ресурсами, предобработка и сохранение собранных данных, а также консолидация данных в центральном хранилище. В ходе решения указанных задач рассматривались основные типы журналируемой информации и механизмы ее сбора, предложены решения вопросов передачи, сбора, хранения и управления собранной информацией.

В основу предлагаемого решения был положен мультиагентный подход: агенты сбора должны на машинах пользователей должны осуществлять сбор поведенческой информации об работе с информационными и вычислительными ресурсами и передачу ее сервер консолидации, который, в свою очередь, должен обеспечивать принятие данных от множества агентов и помещать полученные данные в единое хранилище.

Результаты, полученные в рамках настоящего раздела, подтверждают соответствие требованиям пп. 2.1.3, 3.6, 4.1.1.1 ТЗ.

3 Разработка методов машинного обучения и математической статистики для построения и применения поведенческих моделей на основе биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы

Согласно требованиям п.3.7 ТЗ в ходе выполнения ПНИ должны быть разработаны методы машинного обучения и математической статистики для построения и применения поведенческих моделей с целью решения задач постоянной фоновой идентификации пользователей и раннего обнаружения внутренних вторжений на основе поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы.

Настоящий раздел посвящён теоретическим исследованиям решения указанных задач на основе проведённого аналитического обзора, осуществлённого выбора направления исследований и предложенного подхода, представленных в отчёте за предыдущий этап настоящих ПНИ (см. подразделы 1.2 и 2.2 в [1], соответственно).

По результатам выбора направления исследований для решения поставленных в рамках настоящих ПНИ задач постоянной фоновой идентификации пользователей и раннего обнаружения вторжений на основе поведенческой биометрической информации об особенностях работы пользователя с информационным и вычислительными ресурсами защищаемой компьютерной системы был предложен подход, основанный на формировании шаблона поведения пользователя и применении методов машинного обучения и математической статистики для выявления случаев аномального поведения пользователя. В качестве аналитического инструментария было предложено использовать метод ассоциативных правил к выявлению закономерностей в работе пользователей. Также было предложено провести теоретические исследования по разработке методов машинного обучения и математической статистики в рамках подхода анализа транзакций действий пользователя для построения и применения поведенческих моделей на основе биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы.

Как было показано в аналитическом обзоре, представленном в отчёте за предыдущий этап настоящих ПНИ (см. подраздел 1.2 в [1]), в качестве исходных данных для решаемой задачи согласно традиционно используемой технологии SIEM выступают элементарные события, фиксируемые в ходе деятельности пользователей/процессов в журналах различных приложений и операционных систем, работающих в рамках компьютерной системы. В общем случае, события журналов регистрации можно отнести к слабоструктурированным данным, поскольку помимо набора обязательных атрибутов могут содержать дополнительные данные. Традиционно в SIEM-системах используются подходы анализа фиксируемых событий на основе выделения из них числовых атрибутов, тем самым множество анализируемых событий представляются в структурированном виде. В настоящих исследованиях проводится изучение применимости традиционных подходов на примере подхода анализа транзакций действий пользователя.

С другой стороны, каждое фиксируемое событие можно представить в текстовой форме (в виде текстовых сообщений), описывающей его характеристики. В связи с этим к данному типу собираемой информации можно применить подход по анализу текстового содержимого, предложенный участниками коллектива в рамках выбора направления исследований возможности построения и применения поведенческих моделей на основе поведенческой биометрии работы пользователя с текстовыми данными. Главным достоинством выбранного подхода является то, что на природу происхождения текстовых данных не накладываются никакие ограничения. Таким образом, в настоящей работе будут проведены теоретические исследования предложенного подхода к анализу работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы на основе тематического моделирования сложившихся в прошлом тенденций работы (поведения) пользователя, определяемых по текстовым сообщениям событий. Отметим, что предлагаемый новаторский подход ранее никем не использовался.

Как указывалось в разделе 2 настоящего отчета, принимая во внимание требование п.4.1.2.9 ТЗ, дальнейшие теоретические исследования будут вестись применительно к данным, полученным из журналов регистрации ОС семейства Microsoft Windows.

3.1 Задача раннего обнаружения внутренних вторжений

Решением *задачи раннего обнаружения внутренних вторжений* является выявление фактов аномального или подозрительного поведения инсайдеров (авторизованных легальных пользователей или нарушителей, авторизовавшихся под чужим именем), которые могут

предшествовать или непосредственно являться частью организации попытки внутреннего вторжения. Это утверждение обусловлено тем, что в большинстве случаев непосредственно внутреннему вторжению предшествует аномальное (хотя возможно и разрешенное) поведение пользователя, т.е. пользователь еще до атаки или кражи информации начинает совершать действия, не характерные для его предыдущей активности.

На основе проведенного на предыдущем этапе настоящих ПНИ выбора направления исследований к решению задачи раннего обнаружения внутренних вторжений был предложен подход, основанный на формировании шаблона поведения пользователя и применении методов машинного обучения и математической статистики для выявления случаев аномального поведения пользователя. В основе данной группы методов лежит модель, описывающая нормальное поведение пользователя. В дальнейшем, текущее поведение сравнивается с построенной моделью с использованием некоторой функции сравнения, и в случае сильного «отличия» (превосходящего некоторое пороговое значение) фиксируется аномалия, которая может свидетельствовать о потенциальном внутреннем вторжении.

В рамках рассматриваемого подхода были выбраны методы машинного обучения и интеллектуального анализа данных (Data Mining). Основная идея применения этих методов состоит в предположении о том, что активность пользователя может быть отслежена, и построена ее математическая модель, которая позволит обнаруживать аномалии в поведении пользователей. По свидетельству многих специалистов по защите информации, такой подход является особенно перспективным в задачах обнаружения именно внутренних вторжений, поскольку он позволяет создавать так называемые системы «раннего обнаружения» (early warning). Обнаружение таких аномальных действий позволяет определить подозрительных пользователей, проверить, ужесточить и скорректировать настройки политик безопасности для них и установить более детальное наблюдение за их активностью еще до возможного злонамеренного действия с их стороны, что позволяет предотвратить многие внутренние вторжения.

Расследование и обнаружение подготовки внутренних вторжений может осуществляется как путем выявления аномальных действий, так и с помощью статистического анализа, т.е. путем выявления нецелевого использования, которое в свою очередь может быть основой для внимания служб безопасности. Также для противодействия вторжениям может использоваться анализ информации о предыдущих атаках на основе собранной информации. И наоборот выявленные аномалии в работе могут служить поводом

для проведения более детального статистического анализа за некоторый период работы, с целью выявления нецелевого использования компьютерной системы.

Таким образом, был сделан вывод, что указанные задачи мониторинга лучше всего решаются с помощью статистического и интеллектуального анализа данных, которые прекрасно дополняют друг друга.

3.1.1 Предлагаемый подход

С точки зрения разработанных в разделе 2 структур данных, методов сбора, предобработки, хранения и управления для поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы целью использования интеллектуального анализа данных является классификация фактов активности по степени аномальности относительно обучающего набора. Для этого из хранилища выбирается тренировочный набор фактов активности (обычно факты за временной период, соответствующей обычной работе). На выбранном тренировочном наборе «обучается» алгоритм выявления аномалий. Затем алгоритм применяется к вновь поступающим фактам, классифицируя их как нормальные или как аномальные, что позволяет оценить, насколько новая активность отличается от предыдущей («нормальной») активности, т.е. оценить аномальность отдельных событий и их атрибутов. Таким образом, для успешного решения задачи необходимо решить две подзадачи — задачу обучения модели поведения пользователя и задачу ее применения.

3.1.1.1 Задача обучения модели поведения пользователя

Задача обучения модели поведения пользователя (далее — модель) независимо от используемого алгоритма содержит следующие обязательные шаги:

1. *Определение обучающего набора* для построения модели. Определение набора атрибутов фактов, значения которых будут рассматриваться при построении модели. Имеет смысл включать в модель только действительно значимые атрибуты, потому как, например, включение атрибута «время» всегда будет приводить к некоторой аномальности атрибута при классификации, так как время практически у всех фактов разное. Определение фактов обучающего набора. Обычно это выделение событий, описывающих, по мнению эксперта, нормальную работу.
2. *Кластеризация числовых атрибутов*. Если рассмотреть значения числовых атрибутов какого-либо факта, например, объем переданной информации для фактов сетевой

активности, то почти все они будут различные. Это опять же приведет к обнаружению «ненужных» аномалий для данных атрибутов. Для числовых атрибутов производится кластеризация, т.е. выделение основных групп значений. Считается, что значения различны, только, если они принадлежат разным кластерам.

3. *Построение модели*, описывающей корреляцию значений атрибутов. К построенной модели и методам ее обучения предъявляется лишь одно требование. Модель должна учитывать корреляцию атрибутов фактов с целью реализации функции, позволяющей определить распределение условной вероятности значений каждого атрибута в зависимости от значений остальных атрибутов.

Учет корреляции атрибутов в рамках рассматриваемого подхода описывается с помощью ассоциативных правил. Для построения ассоциативных правил из собранных данных выделяются факты, например, за требуемый временной период. Основная идея подхода заключается в использовании алгоритмов поиска ассоциативных правил для выявления корреляций между элементами в транзакции, что в нашем случае соответствует корреляциям между значениями атрибутов факта.

В рамках решаемой задачи применение метода ассоциативных правил выглядит следующим образом. Пусть любой факт x описывается набором из n атрибутов (характеристик), где область определения атрибутов задана следующим образом: $x = (x_1, ..., x_n) \in X = dom(x_1) \times ... \times dom(x_n)$. Результатом работы алгоритма является система из m ассоциативных правил $\{R_s(x)\}_{s=1}^{m}$ вида (4):

$$R_s(x) = "A_{i1}(x), ..., A_{il}(x) \Rightarrow B_{j1}(x), ..., B_{jk}(x)" \qquad [c, s] \tag{4}$$

где: $A_l(x), B_l(x)$ — предикаты, задающие условия на значения *l-го* атрибута $x_l \in X_l$, в частности, это может быть диапазон значений для числовых атрибутов (например «Duration=77 – 1384»), и конкретные значения для дискретных атрибутов (например, «Process=cmd.exe»); s — поддержка (частота встречаемости) правила, определенная по следующей формуле (5) (число фактов, для которых выполнены все предикаты правила, как в левой, так и в правой части):

$$s = \text{support}(R(x)) = \left| \left\{ x \in X \mid A_{i1}(x) \wedge ... \wedge A_{il}(x) \wedge B_{j1}(x) \wedge ... \wedge B_{jk}(x) \right\} \right| \tag{5}$$

c — достоверность правила, определенная по следующей формуле (6) (число фактов, в которых, из выполнения всех предикатов левой части правила, следует выполнение всех предикатов правой):

$$c = \text{confidence}(R(x)) = \frac{\left|\left\{x \in X \mid A_{i1}(x) \wedge ... \wedge A_{il}(x) \wedge B_{j1}(x) \wedge ... \wedge B_{jk}(x)\right\}\right|}{\left|\left\{x \in X \mid A_{i1}(x) \wedge ... \wedge A_{il}(x)\right\}\right|} \quad (6)$$

Ассоциативное правило *R(x)* может быть проинтерпретировано так: «если атрибуты факта *x* удовлетворяют предикатам $A_{i1},...,A_{il}$, то с вероятностью *c* данный факт будет удовлетворять предикатам $B_{j1},...,B_{jk}$». Например, правило вида «User=UserA, Process=cmd.exe => Duration=77 – 1384" (probability=1.0) означает, что «если пользователь UserA запустил процесс cmd.exe, то этот процесс всегда проработает не меньше 77 и не больше 1384 секунд». Помимо простой и эффективной интерпретации самих правил, сильной стороной ассоциативного подхода является простое и эффективное определение «частого эпизода», т.е. часто встречаемой устойчивой комбинации атрибутов. Любой «частый эпизод» также напрямую определяется ассоциативным правилом *R(x)*, где частота определяется как значение поддержки (support) правила. Важность правила определяется следующим образом (7):

$$\text{Importance}(A \Rightarrow B) = \log(P(a \mid b) / P(a \mid \neg\, b)) \quad (7)$$

где A и B — пары вида <атрибут = значение>. Нулевая важность указывает на отсутствие корреляции между A и B. Положительная важность означает, что вероятность B увеличивается, при выполнении A. Отрицательная важность говорит о том, что, вероятность выполнения B уменьшается при условии выполнения A.

Реализация алгоритма построения ассоциативных правил основывается на следующих шагах:

- На основе множества входных фактов выделяется набор групп атрибутов и их значений, которые встречаются вместе как минимум MINIMUM_SUPPORT (параметр алгоритма) раз. Примером такой группы из 2х атрибутов для фактов запуска процессов может быть «Process = cmd.exe; Parent = explorer.exe», с поддержкой, например, 147.
- Затем на основе выделенных групп строятся правила, предсказывающие значения одних атрибутов в зависимости от значений других атрибутов, которые алгоритм определил, как важные. Например, для группы из трех атрибутов правило может иметь вид «если Process = cmd.exe и Parent = explorer.exe, то User Name = mike» и иметь вероятность 0,781.

3.1.1.2 Задача применения модели

Идея применения модели базируется на том, модель можно использовать для прогнозирования значений одних атрибутов по другим. Для этого строится функция

$P(x_i \mid x_1,...,x_{i-1},x_{i+1},...,,x_n)$, которая вычисляет распределение условной вероятности значений *i-го* атрибута в зависимости от остальных атрибутов. В этом случае, уровень достоверности (нормальности) значения *i-го* атрибута определяется как отношение условной вероятности реально наблюдаемого значения a_s к вероятности наиболее ожидаемого значения (8):

$$Score(x_i \mid x_i = a_s) = \frac{P(x_i = a_s \mid x_1,...,x_{i-1},x_{i+1},...,,x_n)}{\max_l P(x_i = a_l \mid x_1,...,x_{i-1},x_{i+1},...,,x_n)} \quad (8)$$

Очевидно, что если значение *i-го* атрибута совпадает с наиболее ожидаемым, т.е. тем, которое прогнозируется на основе найденной ранее в модели, то уровень достоверности такого атрибута равен 1, т.е. абсолютно «ожидаемое» значение. Если же такое значение ранее вообще не встречалось, то его условная вероятность будет равна нулю и уровень достоверности атрибута также будет равен нулю, что соответствует абсолютно «аномальному» значению. В остальных случаях значение достоверности будет меняться от 0 до 1, чем меньше это значение, тем аномальнее значение атрибута. Достоверность всего события *x* в этом случае можно определить, например, как произведение достоверностей его атрибутов (9):

$$Score(x) = \prod_i Score(x_i) \quad (9)$$

Такой подход дает возможность не только обнаружить аномальные факты (события), но и найти причину аномальности, т.е. указать те атрибуты, которые являются не нормальными с точки зрения предыдущей активности пользователей.

Задача применения модели состоит из следующих обязательных шагов:

1. Определение подмножества фактов, которые требуется проклассифицировать.
2. Определение набора атрибутов фактов, которые желательно проклассифицировать по степени аномальности.
3. Пересечение набора атрибутов фактов, используемых при обучении модели и выбранных для применения модели. Так как модель содержит информацию лишь об атрибутах фактов, на которых она обучалась, а для применения модели выбрано новое подмножество атрибутов, то именно для множества атрибутов, образующих пересечение, может быть вычислена аномальность, и соответственно, только на основе значений этих атрибутов определяется аномальность всего факта.
4. Классификация нормальности фактов на основе условной вероятности появления значений атрибутов.

В следующем пункте приводятся экспериментальные исследования предложенного метода на основе ассоциативных правил при решении задачи раннего обнаружения внутренних вторжений на основе поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы.

3.1.2 Экспериментальные исследования

Первоочередной задачей при проведении экспериментальных исследований предлагаемого подхода раннего выявления внутренних вторжений являлся выбор тестового набора данных. Исходя из формулировки решаемой задачи были предъявлены следующие критерии к набору экспериментальных данных:

- данные о работе пользователей в рамках корпоративной среды;
- возможность сопоставления записей журналов с пользователями;
- возможность определения времени каждой записи в журналах;
- возможность выделения записей журналов, свидетельствующих о вероятном вторжении.

На основе сформулированных выше требований для экспериментального исследования предлагаемого подхода был выбран набор НЭД1 (см. подраздел 1.2 ОВБС), сформированном на основе набора DARPA 1999 [37], находящегося в свободном доступе.

Для экспериментального исследования предлагаемого подхода данные из набора НЭД1 были выгружены в экспериментальную базу данных. Для ее организации использовалась СУБД Microsoft SQL Server 2014 — это обуславливалось тем, что в качестве реализации предлагаемого подхода было принято решение об использовании аналитического инструментария ПО Microsoft SQL Server 2014 Analysis Services, в состав которого также входит реализация алгоритма взаимосвязей Microsoft Association Rules [38].

События из набора НЭД1 были выгружены в таблицу БД, имеющей структуру, приведенную в таблице 8. Затем были построены представления (VIEW), позволяющие получать подмножества событий за каждую неделю сбора, которые приведены в таблице 9.

Таблица 8 — Структура таблицы выгруженных событий.

Наименование столбца	Описание
ID	Идентификатор записи в таблице, соответствующей

	выгруженному событию
TIMECREATED	Дата и время фиксации события
ISTRAIN	Признак, принадлежит ли событие обучающему набору
HASINTRUSION	Признак, что данное событие соответствует какой-либо атаке, выявленной с помощью программного средство маркировки вторжений
EVENTID	Значение атрибута «Идентификатора события»
LEVEL	Значение атрибута «Тип события»
LOGNAME	Указание журнала-источника события
RECORDID	Номер записи в журнале событий
PROVIDER	Источник события
USERNAME	Имя пользователя, от имени которого зафиксировано событие
DESCRIPTION	Текстовое описание события

Таблица 9 — Характеристики временных интервалов сбора событий набора DARPA 1999.

	Неделя	Начало	Окончание
Обучающий набор	1	01.03.1999 00:00:00	07.03.1999 23:59:59
	2	08.03.1999 00:00:00	14.03.1999 23:59:59
	3	15.03.1999 00:00:00	28.03.1999 23:59:59
Тестовый набор	4	29.03.1999 00:00:00	04.04.1999 23:59:59
	5	05.04.1999 00:00:00	11.04.1999 23:59:59

При построении указанных представлений учитывались результаты обработки исходного набора DARPA 1999, проведенной в ходе формирования набора НЭД1 (см. подпункт 1.2.1.1 ОВБС). Поэтому было принято решение для формирования обучающего набора использовать события, собранные в рамках 1ой и 3ей недель, а события, собранные в рамках 4ой и 5ой недель, — для формирования тестового набора.

Сценарий проведения экспериментальных исследований состоит из следующих шагов:

1. На основе обучающего набора производилось построение и обучение модели на основе указанного алгоритма взаимосвязей Microsoft Association Rules с подбором параметров алгоритма и множества атрибутов событий, используемых в модели.
2. К тестовому набору (к каждому событию) применялась построенная модель, рассчитывались оценки аномальности каждого из исследуемых атрибутов событий и

вычислялась общая оценка аномальности текущего события. Результат расчета также заносился в экспериментальную базу.

3. К тестовому набору применялось программное средство маркировки вторжений, подготовленное в рамках формирования набора НЭД1. Данное средство отмечало в тестовом наборе те события, которые соответствуют признаку какой-то конкретной атаки (исследуемой в текущей серии экспериментов) либо произвольной атаки (для оценки применимости метода выявления произвольной атаки).
4. На основе сопоставления полученных в пп. 2 и 3 характеристик (откликов) оценивалось качество полученного классификатора.

Были проведены несколько серий экспериментов, в рамках которых подбирались параметры алгоритма взаимосвязей, описанные в [39], а также набор атрибутов событий, участвующих при построении модели. Наилучшие результаты были получены в серии экспериментов, характеристики с нестандартными значениями которой приведены в таблице 10.

Таблица 10 — Характеристики серии экспериментов.

Параметр	Значение
MAXIMUM_ITEMSET_SIZE	3
MINIMUM_ITEMSET_SIZE	2
MAXIMUM_SUPPORT	1.0
MINIMUM_SUPPORT	0.0
MINIMUM_PROBABILITY	0.4
Список атрибутов событий	EVENTID, LEVEL, PROVIDER, USERNAME

Программное средство маркировки вторжений позволяет размечать события, советующие сигнатурам следующих видов атак (в формулировках [40]):

- netbus — атака заключается в установке на машине жертвы NetBus-сервера. В последствии атакующий может подключаться к серверу удаленно и выполнять практически любые действия от имени работающего в данный момент пользователя.
- yaga — атака, направленная на создание нового пользователя в группе администраторов путем взлома реестра.
- NTinfoscan (или ntis) — атака, заключающаяся в NetBIOS-сканировании с целью сбора сведений информационной безопасности. На целевой NT-машине собирается вся доступная информация, включая имена всех пользователей, работающих сервисов и пр.,

и сохраняется, например, в html-файл, который впоследствии скачивается злоумышленником.

– CaseSen — атака, направленная на получение администраторских прав пользователем. Атака использует чувствительность к регистру каталога объектов NT.

В ходе экспериментальных исследований были проведены серии экспериментов по выявлению каждого типа атаки по отдельности, так и в комплексе.

Для оценки качества классификации обычно используют ROC-кривые, описывающие в графической форме качество бинарного классификатора, зависимость доли верных положительных классификаций от доли ложных положительных классификаций при варьировании порога решающего правила (отклонения) [23]. Для сравнения нескольких моделей классификации будем использовать значение AUC (англ. Area Under Curve), которое вычисляется как площадь под ROC-кривой и является агрегированной характеристикой качества классификации, не зависящей от соотношения цен ошибок [23]. Чем больше значение AUC, тем «лучше» модель классификации. Для серии экспериментов, описанной в таблице 10, получены значения AUC, приведенные в таблице 11.

Таблица 11 — Значения AUC для метода выявления вторжений на основе алгоритма взаимосвязей.

Атака	AUC
netbus	0,609995316
yaga	0,901150192
ntis	0,990056742
CaseSen	0,991929573
Комбинация атак	0,82659979

Проведенные экспериментальные исследования показали высокое качество решения задачи раннего обнаружения внутренних вторжений на основе поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы, тем самым подтвердив применимость предложенного подхода на основе использования методов машинного обучения и интеллектуального анализа данных.

3.2 Задача постоянной фоновой идентификации пользователей

Решение *задачи постоянной фоновой идентификации пользователей* заключается в оценке достоверности того, что пользователь, работающий с защищаемой компьютерной системой, является действительно тем, от имени кого он авторизовался. В отличие от аутентификации в настоящей задаче не подразумевается применение явных процедур проверки, требующих интерактивных действий от пользователя. В настоящем разделе рассматриваются методы машинного обучения и математической статистики для построения и применения поведенческих моделей с целью решения данной задачи на основе информации о работе пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы.

В основе подхода к решению задачи идентификации, предложенного коллективом авторов на этапе выбора направления исследований, лежит моделирование работы пользователя с информационными и вычислительными ресурсами на основе данных, собранных за заданное модельное время, где под модельным временем понимается временной интервал, в течение которого собиралась информация о характерной работе данного пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы.

Суть подхода заключается в проведении анализа модельного времени с целью расчета характеристик, описывающих сложившиеся в прошлом (в рамках модельного времени) тенденции работы пользователя с информационными и вычислительными ресурсами. Для этого модельное время разбивается на последовательно измеренные через некоторые промежутки времени интервалы. Например, в качестве промежутка времени (шага) может быть выбран час, день, а также время, за которое происходит заданное число событий [2, 3]. Таким образом, выбираемые промежутки времени не обязательно должны быть равны между собой. В результате проведенного анализа вычисляется некоторый вектор числовых характеристик каждого временного интервала модельного времени. В своей совокупности указанные векторы образуют многомерный временной ряд, описывающий работу пользователя с информационными и вычислительными ресурсами во времени. Далее по сформированным временным рядам строятся прогнозы. На основе значений отклонений характеристик текущей работы пользователя от спрогнозированных данных определяются временные интервалы работы пользователя, не свойственные ему. Таким образом, вычисляемые отклонения являются оценками достоверности того, что пользователь, работающий с защищаемой компьютерной системой, является действительно тем, от имени

кого он авторизовался, тем самым решается задача идентификации пользователей в постановке, приведённой в начале настоящего подраздела.

Как было показано в аналитическом обзоре, представленном в отчёте за предыдущий этап настоящих ПНИ (см. подраздел 1.2 в [1]), в качестве исходных данных для решаемой задачи согласно традиционно используемой технологии SIEM выступают элементарные события, фиксируемые в ходе деятельности пользователей/процессов в журналах различных приложений и операционных систем, работающих в рамках компьютерной системы. В общем случае, события журналов регистрации можно отнести к слабоструктурированным данным, поскольку помимо набора обязательных атрибутов могут содержать дополнительные данные, которые, например, в журналах аудита ОС Microsoft Windows представляются пользователю в текстовом поле Description («Описание»). Подходы, предлагаемые в рамках настоящих теоретических исследований, основываются на различных представлениях элементарных событий. В одном случае, из каждого события извлекается фиксированный набор числовых атрибутов (возможно, с пропущенными значениями), тем самым представление события относится к структурированному. В другом подходе каждое событие представляется в виде текста его описания (или сообщения), тем самым «забывая» его структуру, и в этом случае применяются методы обработки неструктурированных данных (таких, как текстовые данные).

В качестве методов поведенческого моделирования, используемого в описанном подходе, на этапе выбора направления исследований были предложены методы расчета статистических характеристик на основе транзакционного подхода (использующего структурное представление каждого события), а также методы, основанные на применении тематического моделирования с использованием техники неотрицательной матричной факторизации (рассматривающих события как поток сообщений). Ниже в настоящем разделе проводятся теоретические исследования вопросов применимости каждого из методов для решения поставленной задачи.

3.2.1 Метод на основе тематического моделирования

Участники коллектива имеют значительный задел в разработке методов тематического моделирования с использованием методов неотрицательной матричной факторизации [3, 5–9]. Кроме того, был разработан метод на основе неотрицательной матричной факторизации, который хорошо себя показал в задаче поддержки принятия решения в части предоставления доступа к ресурсам [1, 18]. Поэтому был поставлен вопрос о применимости методов тематического моделирования в задаче анализа событий из журналов

регистрации, являющихся исходными данными (в рамках рассматриваемой технологии SIEM), содержащими поведенческую биометрическую информацию об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы (см. подраздел 1.2 в [1]).

Предлагаемый метод на основе тематического моделирования применительно к задаче анализа текстовых данных детально описывается в пункте 1.1.1 настоящего отчета. Указанный метод ожидает на входе набор текстовых данных и в явной форме к множеству событий не применим. С другой стороны, как отмечалось выше, события журналов регистрации можно обрабатывать в неструктурированном представлении — например, в виде текстовых сообщений, содержащих информацию о произошедшем событии. Таким образом, был участниками коллектива было предложено опробовать метод тематического моделирования на основе неотрицательной матричной факторизации для анализа текстовых сообщений событий, зафиксированных в журналах регистрации ОС Microsoft Windows.

3.2.1.1 Набор экспериментальных данных

Первоочередной задачей при проведении экспериментальных исследований предлагаемого подхода раннего выявления внутренних вторжений являлся выбор тестового набора данных. Исходя из формулировки решаемой задачи были предъявлены следующие критерии к набору экспериментальных данных:

- данные о работе пользователей в рамках корпоративной среды;
- возможность сопоставления записей журналов с пользователями;
- возможность определения времени каждой записи в журналах.

На основе сформулированных выше требований для экспериментального исследования предлагаемого подхода был выбран набор НЭД2.1, являющийся частью набора НЭД2 (см. подраздел 1.2 ОВБС) и содержащий обезличенные данные, собранные в рамках реальной организации. Отличительной особенностью набора является то, что на каждом отдельном компьютере работал один пользователь. С помощью разработанного программного средства, входящего в состав сформированного набора НЭД2, собранные данные были выгружены в виде текстовых файлов в файловую систему с разбиением по компьютерам (т.е. для каждого компьютера была создана папка, в которую помещались файлы с информацией о событии: для каждого события — отдельный файл). Заголовок каждого сгенерированного файла формировался на основе подстановки значений атрибутов текущего события в заранее заданный пользователем шаблон заголовка, а именно:

информация о времени фиксации события, названии журнала, глобального идентификатора события. Содержимое текстового файла является текстовым описанием события, зафиксированного в журнале ОС.

3.2.1.2 Предварительные экспериментальные исследования

Целью предварительных исследований было изучение вопроса о применимости предложенного подхода к анализу событий в текстовом представлении. Для этого из набора НЭД2.1 были отобраны данные, собранные на 6 компьютерах пользователей. В наборе НЭД2.1 данные собирались в течение 9 последовательных дней, причем большинство компьютеров включались только в рабочие часы пользователей, поэтому данные в большинстве случаев представлены за первые 5 рабочих дней (с понедельника по пятницу), затем следуют 2 дня выходных, когда компьютеры были выключены и данные не собирались, после этого в наборе представлены собранные данные для понедельника и вторника следующей недели. Отметим, что в наборе присутствуют несколько компьютеров, которые на выходные не выключались.

Для проведения предварительных экспериментов с отобранными данными была проведена следующая предобработка. Для каждого компьютера события были сгруппированы по дням, т.е. для каждой даты был получен текстовый файл, содержащий текстовые описания всех зафиксированных событий (в качестве *временного интервала* выступают сутки работы). После этого была произведена замена имен пользователей и компьютеров на унифицированные. Сделано это было с целью моделирования ситуации, когда один пользователь смог завладеть доступом к работе на компьютере другого пользователя под его учетной записью. Полученные таким образом данные применимы для проведения экспериментальных исследований решения задачи идентификации пользователей в формулировке, приведенной в начале настоящего подраздела.

В дальнейшем, данные, собранные в течение первой недели (за первые 5 или 7 дней) составляли обучающий набор, а данные, собранные в течение второй недели (8 и 9 дни), — тестовый. К обучающему набору для выбранного компьютера (анализируемого пользователя) был применен метод тематического моделирования (с параметром числа тематик $k=3$), описанный в пункте 1.1.1 настоящего отчета, а затем полученную модель применили к тестовому набору, содержащему предобработанные данные для всех 6 компьютеров. Таким образом, для каждого «склеенного» файла событий было получено его векторное представление в пространстве тематик, графически изображенных на рисунках 16 и 17.

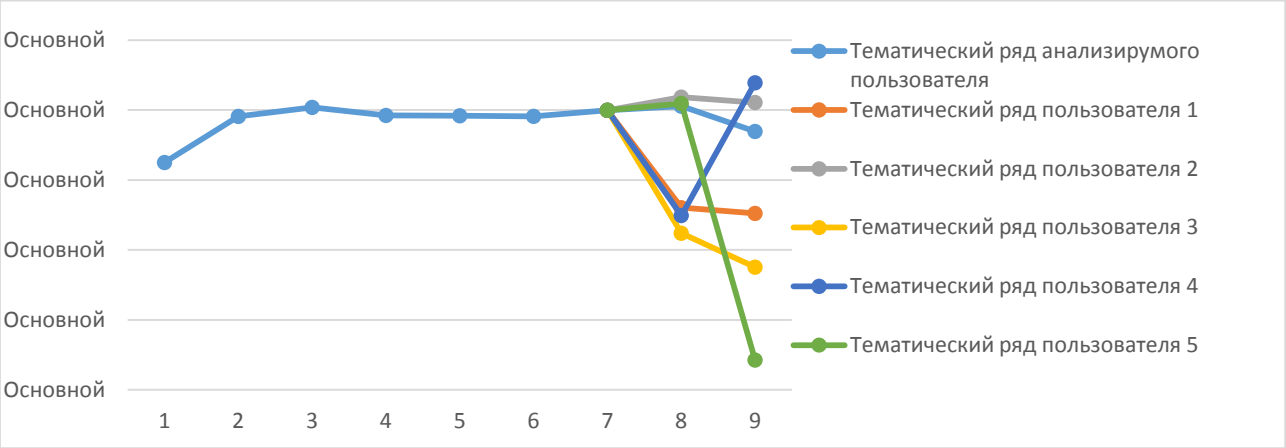


Рисунок 16 — Временной ряд для одной из тематик для пользователя, работавшего 7 дней. Точки 8 и 9 — точки тестового периода.

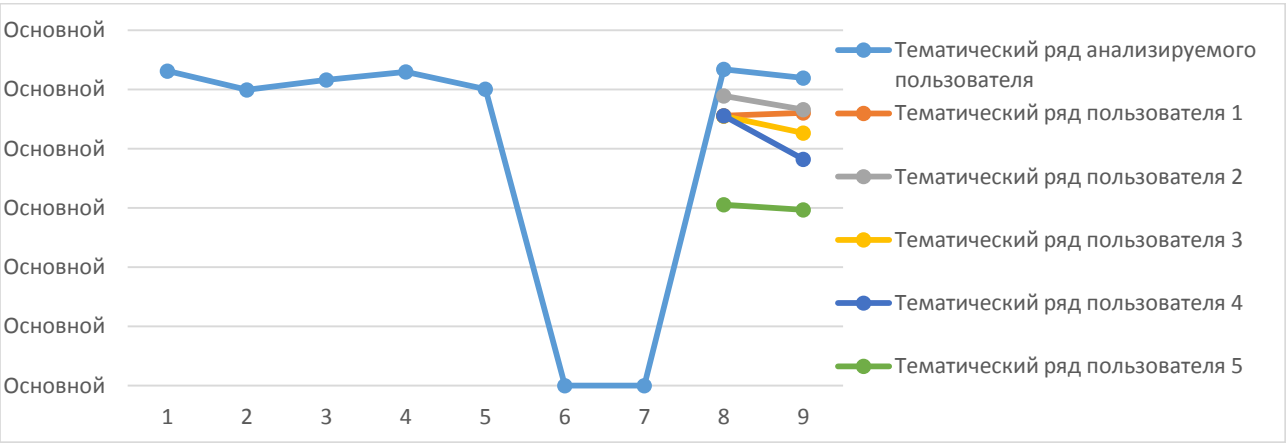


Рисунок 17 — Временной ряд для одной из тематик для пользователя, работавшего 5 дней. Точки 8 и 9 — точки тестового периода.

Наряду с расчетом тематических характеристик участниками коллектива было предложено дополнительно рассчитывать характеристику оценки качества аппроксимации тематической модели. Суть тематического моделирования заключается в аппроксимации исходной числовой матрицы A временных интервалов матрицей A_k: $A \approx A_k = W_k \cdot H_k$ (см. пункт 1.1.1 настоящего отчета). В качестве указанной характеристики предлагается использовать ошибку аппроксимации, рассчитываемой для каждой временного интервала как евклидова норма для каждого вектор-столбца матрицы $Approx = A - A_k$. Примеры временных рядов указанной характеристики представлены на рисунках 18 и 19.

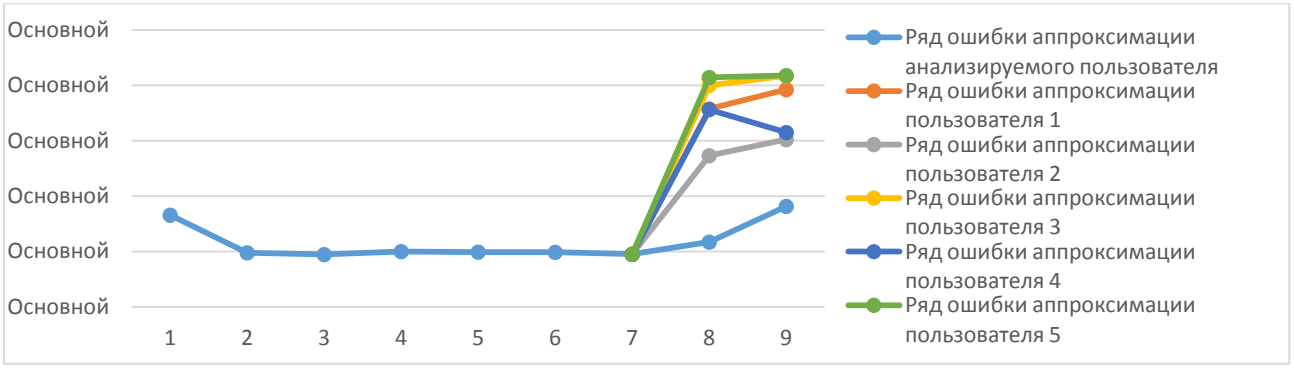


Рисунок 18 — Временной ряд ошибки аппроксимации для пользователя, работавшего 7 дней. Точки 8 и 9 — точки тестового периода.

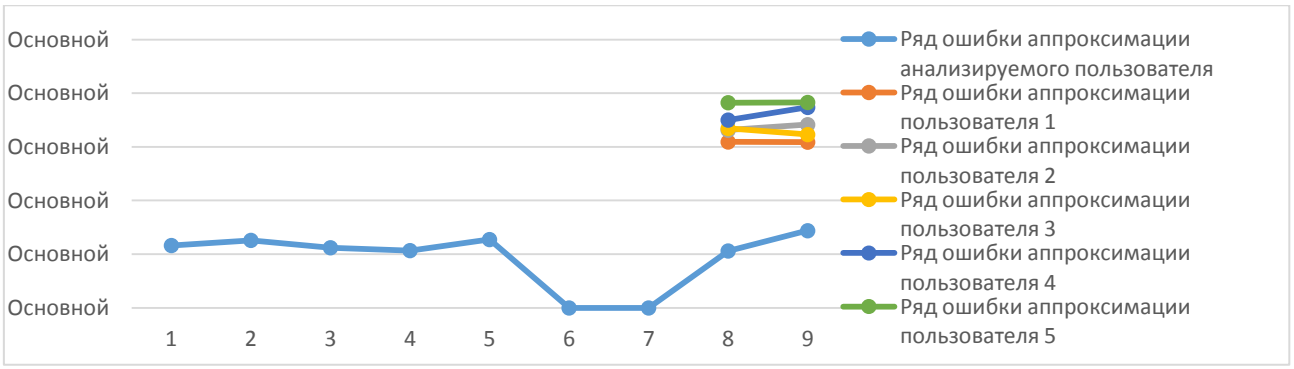


Рисунок 19 — Временной ряд ошибки аппроксимации для пользователя, работавшего 5 дней. Точки 8 и 9 — точки тестового периода.

В результате предварительных экспериментальных исследований был сделан вывод о применимости предложенного подхода на основе тематического моделирования к решению поставленной задачи идентификации пользователей и принято решение о проведении дальнейших теоретических исследований.

3.2.1.3 Экспериментальные исследования

В настоящем подпункте описаны экспериментальные исследования применения подхода на основе тематического моделирования текстовых сообщений событий, фиксируемых в журналах регистрации ОС о работе пользователей с информационными и вычислительными ресурсами защищаемой компьютерной системы, в рамках решения задачи идентификации пользователей на основе оценки отклонения поведения в работе пользователя от прогноза. Для построения прогноза временных рядов использовались модели, описанные в пункте 1.1.2 настоящего отчета и подпункте 2.3.1.2 в [1]: модель авторегрессионного интегрированного скользящего среднего (ARIMA — AutoRegressive

Integrated Moving Average) [17, 41, 42], авторегрессионная модель дерева (AutoRegressive Tree Model, ART) [16, 17] и предложенная оригинальная модель на основе ортонормированной неотрицательной матричной факторизации [3].

Для проведения экспериментальных исследований использовался набор НЭД2.1, описанном в подпункте 3.2.1.1 настоящего отчета. Использовались данные, собранные для всех представленных в наборе пользователей (в количестве 20 человек, или компьютеров). Предобработка данных проводилась аналогично указанной в подпункте 3.2.1.2 следующим образом. В качестве временного интервала использовались сутки. Для каждого временного интервала, представленного в наборе НЭД2.1 был получен текстовый файл, содержащий текстовые описания всех зафиксированных событий за этот день. После этого была произведена замена имен пользователей и компьютеров на унифицированные.

Для каждого пользователя в наборе данные, собранные в течение первой недели (за первые 5 или 7 дней) составляли *обучающий набор* (т.е. данные модельного времени), а данные, собранные в течение второй недели (8 и 9 дни), — *тестовый* (т.е. данные времени прогноза).

Для демонстрации предлагаемого подхода идентификации работы пользователя рассматривалась следующая задача бинарной классификации: требуется отделить временные интервалы работы анализируемого пользователя от временных интервалов работы остальных 19-ти пользователей из набора для каждого тестового набора.

Сценарий проведения экспериментальных исследований выглядит следующим образом.

1. Для каждого обрабатываемого текстового файла сообщений событий использовались средства текстовой предобработки. Поскольку все сообщения в наборе НЭД2.1 являются англоязычными, то для формирования словаря термов использовались такие методы предварительной обработки текста, как удаление стоп-слов и приведение слов к нормализованной форме на основе семантической сети WordNet [22]. Для вычисления весов термов использовался только локальный логарифмический вес. Векторы временных интервалов нормализовались по евклидовой норме.
2. К сформированной матрице модельных временных интервалов анализируемого в настоящий момент пользователя применялось тематическое моделирование на основе ортонормированной неотрицательной матричной факторизации для получения матрицы «портрета» пользователя (W_k) и матрицы представления временных интервалов в пространстве тематик (H_k). В итоге для анализируемого пользователя получаем k

тематических временных рядов для модельного времени (в проводимых экспериментах k=3).

3. Отображения векторов временных интервалов времени прогноза для всех пользователей (и анализируемого, и остальных) осуществлялось с использованием матрицы «портрета» анализируемого пользователя (W_k). Таким образом, получаем реальные тематические данные всех пользователей для временных интервалов прогноза.
4. С помощью каждого перечисленного выше метода прогнозирования строились прогнозы на основе тематических временных рядов анализируемого пользователя за модельное время.
5. Для каждого метода прогнозирования рассчитывалась оценка отклонения каждой временной точки времени прогноза всех пользователей от спрогнозированных значений. В качестве оценки отклонения временной точки от прогноза использовалась *абсолютная оценка* — сумма по всем k тематикам абсолютного отклонения реальных значений весов тематик от спрогнозированных.

В качестве реализации моделей ART и ARIMA использовался алгоритм временных рядов Microsoft Analysis Services [17], который использует оба метода и объединяет результаты для повышения точности прогнозирования. Через параметр *PREDICTION_SMOOTHING* алгоритма настраивалась степень влияния каждого из базовых методов на итоговый результат. В экспериментах использовались модели со значениями *PREDICTION_SMOOTHING* = 0; 0.5; 1.0. Будем обозначать эти модели, соответственно, MS_0.0, MS_0.5 и MS_1.0. Модель на основе ортогональной неотрицательной матричной факторизации будем обозначать ONMF.

После проведения вышеописанной процедуры для каждого метода прогнозирования получаем, что всем прогнозируемым временным интервалам всех пользователей сопоставлена их оценки отклонения. Фиксируя значение порога допустимого отклонения от прогноза анализируемого пользователя получаем бинарную классификацию для всех прогнозируемых временных интервалов. Данные серии экспериментов были проведены для всех 20 пользователей, представленных в наборе НЭД2.1. Для оценки качества классификации использовались ROC-кривые и оценка AUC, описанные выше в пункте 3.1.2 настоящего отчета. Таким образом, были получены 80 значений AUC. Для оценки полученного множества значений AUC использовались устойчивые (робастные) оценки центральной тенденции (медиана) и разброса (интерквартильный размах, ИКР) [25, 26]. Интерквартильным размахом (англ. Interquartile range) называется разность между третьим и

первым квартилями множества значений AUC. Полученные значения медиан и интерквартильных размахов для норм приведены в таблице 12.

Таблица 12 — Значения медиан и интерквартильных размахов.

	Медиана	Интерквартильный размах
ONMF	0,87162162	0,32432432
MS_0.0	0,85810811	0,31081081
MS_0.5	0,81081081	0,32094595
MS_1.0	0,79054054	0,38193457

На основе экспериментальных исследований можно сделать следующие выводы:

- Предложенный подход идентификации работы пользователя на основе отклонений тематической направленности текстовых сообщений событий о работе пользователя с информационными и вычислительными ресурсами, фиксируемых в журналах регистрации ОС, от спрогнозированных данных показывает высокое качество идентификации даже при использовании стандартных методов прогнозирования.
- Предложенный авторами метода прогнозирования временных рядов, основанный на ортонормированной неотрицательной матричной факторизации, показал высокое качество прогнозирования и свою применимость в рассмотренном подходе идентификации работы пользователя.

3.2.2 Метод на основе транзакционного подхода

В рамках настоящего пункта исследуется подход к решению задачи идентификации пользователей на основе транзакционного подхода. Для обработки событий, попадающих во временной интервал, исследуется подход, рассматривающий каждое событие, как набор числовых характеристик (возможно, с пропущенными значениями) его параметров (например, длительность некоторого действия, число прочитанных байт и т.п.). Далее для каждого временного интервала на основе данных из событий, попавших в него, рассчитывается предопределенный набор статистических показателей: например, оценки статистического распределения длительности работы пользователя с выбранными приложениями (на основе информации о длительностях запусков соответствующих процессов). При этом сам временной интервал рассматривается как *транзакция*, в рамках которой осуществлялся некоторая совокупность действий пользователя. В дальнейшем на выделенном множестве временных интервалов, образующем обучающий набор, строится

модель одноклассового классификатора, которая затем применяется к каждому временному интервалу тестового набора для получения оценки принадлежности данного тестового интервала к модельному классу. Данная оценка будет рассматриваться как оценка достоверности того, что пользователь, работающий с защищаемой компьютерной системой, является действительно тем, от имени кого он авторизовался, тем самым решается задача идентификации пользователей в постановке, приведённой в начале настоящего подраздела.

3.2.2.1 Набор экспериментальных данных

Для проведения экспериментальных исследований рассматриваемого подхода к набору данных выдвигались те же требования, что и в подпункте 3.2.1.1 настоящего отчета. Поэтому для проведения экспериментальных исследований был взят тот же набор НЭД2.1, являющийся частью набора НЭД2 (см. подраздел 1.2 ОВБС).

С помощью разработанного программного средства, входящего в состав сформированного набора НЭД2, собранные данные были выгружены в виде текстовых файлов в файловую систему с разбиением по компьютерам (т.е. для каждого компьютера была создана папка, в которую помещались файлы с информацией о событии: для каждого события — отдельный файл). Заголовок каждого сгенерированного файла формировался на основе подстановки значений атрибутов текущего события в заранее заданный пользователем шаблон заголовка, а именно: информация о времени фиксации события, названии журнала, глобального идентификатора события. Содержимое текстового файла является набором значений числовых параметров, разделенных запятой (т.н. CSV-формат — Comma Separated Values, дословно «значения, разделенные запятой»). Фиксировались следующие числовые параметры:

- длительность;
- число переданных байт;
- число принятых байт;
- число байт, участвующих в обмене;
- количество нажатий клавиш клавиатуры;
- количество действий с мышью.

В случае отсутствия информации в событии о каком-либо параметре, ему присваивалось нулевое значение.

3.2.2.2 Экспериментальные исследования

Для проведения экспериментальных исследований проводилась предобработка исходных данных. В качестве исходных данных брались события, описывающих работу пользователя со следующими приложениями:

- текстовый редактор Microsoft Word (процесс WINWORD.EXE);
- программа для работы с электронными таблицами Microsoft Excel (процесс EXCEL.EXE);
- программа просмотра электронных публикаций в формате PDF Adobe Reader (процесс ACRORD32.EXE);
- интернет-браузер Microsoft Internet Explorer (процесс IEXPLORE.EXE);
- персональный информационный менеджер с функциями почтового клиента Microsoft Outlook (процесс OUTLOOK.EXE).

Из набора были удалены данные пользователей, для которых в событиях не была представлена работа ни с одним из представленных приложений. В итоговый набор вошли данные о работе 15-ти пользователей.

В качестве транзакции брались следующие временные интервалы: 1 сутки, 8 часов, 2 часа, 1 час, 30 минут.

В рамках каждой транзакции для каждого указанного процесса рассчитывалось статистическое распределение для каждого указанного числового параметра события. Таким образом, каждая транзакция описывается вектором числовых характеристик.

Для каждого типа транзакции было проведено разбиение исходного набора НЭД2.1 на пересекающиеся временные интервалы (экспериментальные диапазоны, ЭД), каждый из которых делился на тренировочный и тестовый наборы. Детали разбиения приводятся в таблице 13. При разбиении удалялись ЭД, в которых для анализируемого пользователя отсутствовали тренировочные или тестовые данные.

Таблица 13 — Экспериментальные диапазоны.

№ серии	Длина транзакции	Размер тренировочного набора (транзакций, шт.)	Размер тестового набора (транзакций, шт.)	Количество ЭД
1	30 минут	96	48	16
2	1 час	96	24	25
3	1 час	72	24	36
4	1 час	48	24	42
5	2 часа	24	12	42
6	8 часов	6	3	42
7	1 сутки	7	2	10

В рамках каждой серии экспериментов на основе каждого обучающего набора строилась модель одноклассового классификатора. В качестве методов классификации использовались представители различных групп методов классификации, традиционно используемых в литературе, демонстрирующих высокую точность на широком спектре прикладных задач: метод из группы вероятностных (на примере метода SVM, описанного в подпункте 1.1.4.1 в [1]) и метрических (на примере метода kNN, описанного в подпункте 1.1.4.2 в [1]). Метод нейронных сетей был исключен из рассмотрения в связи с недостаточными размерами обучающих наборов. Для каждого выбранного метода проводились серии экспериментов по подбору параметров.

Для каждого ЭД для каждого метода классификации оценивалась характеристика AUC. Серии экспериментов с идентичными моделями для одинаковых транзакций оценивались с помощью устойчивыми оценками центральной тенденции (медианами) и разбросов (интерквартильными размахами, ИКР) [25, 26]. Наилучшие значения указанных оценок для серий экспериментов с длиной транзакций не менее 2 часов для методов группы SVM, приведенные в таблице 14.

Таблица 14 — Результаты экспериментального исследования.

№ серии	Параметры SVM	Медиана	ИКР
5	$\gamma=10^{-9}$, $\eta=0.7$	0,560235507	0,315178572
6	$\gamma=10^{-9}$, $\eta=0.3$	0,614379085	0,50780584
7	$\gamma=10^{-9}$, $\eta=0.5$	0,65	0,4

Проведенные эксперименты показали удовлетворительное качество решения задачи идентификации пользователей. Исследованный подход по качеству сильно уступает предложенному новаторскому подходу, описанному в предыдущем пункте настоящего отчета.

3.3 Выводы

В данном разделе производилась разработка методов машинного обучения и математической статистики для построения и применения поведенческих моделей с целью решения задач постоянной фоновой идентификации пользователей и раннего обнаружения внутренних вторжений на основе поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы.

Для решения задачи постоянной фоновой идентификации пользователей были разработаны следующие методы:

- метод расчета оценки отклонений тематической направленности текстовых сообщений событий о работе пользователя с информационными и вычислительными ресурсами, фиксируемых в журналах регистрации ОС от сложившихся в прошлом тенденций его работы. В основе данного метода лежит тематическое моделирование, которое предполагает определение основных тематик текстового контента сообщений и расчёт соответствующих им весов в заданные интервалы времени. Рассчитанные веса сформированных тематик задают многомерный временной ряд поведения пользователя, на основе которого строится прогноз и осуществляется расчет указанной оценки отклонений. Данный метод показывает высокое качество идентификации даже при использовании стандартных методов прогнозирования. Стоит отметить, что разработанный метод ранее не применялся для решения рассматриваемой задачи идентификации пользователей на основе поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы.
- метод прогнозирования временных рядов, основанный на ортонормированной неотрицательной матричной факторизации, который показал высокое качество прогнозирования и свою применимость в рассмотренном подходе идентификации работы пользователя;

– метод решения задачи идентификации пользователей на основе транзакционного подхода анализа событий о работе пользователя с информационными и вычислительными ресурсами, фиксируемых в журналах регистрации ОС, который показал удовлетворительное качество решения задачи.

Для решения задачи раннего обнаружения внутренних вторжений был разработан метод на основе ассоциативных правил, который показал высокое качество выявления фактов работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы, несвойственных ему.

Проведённые экспериментальные исследования предложенных методов для решения задач постоянной фоновой идентификации пользователей и раннего обнаружения внутренних вторжений на сформированных наборах экспериментальных данных НЭД1 и НЭД2 подтвердили полученные выводы.

Результаты, полученные в рамках настоящего раздела, подтверждают соответствие требованиям пп. 2.1.4, 3.7, 4.1.1.2 подпункт 2) ТЗ.

4 Разработка структур данных, методов сбора, предобработки, хранения и управления для поведенческой биометрической информации об особенностях работы со стандартными устройствами ввода-вывода (клавиатура, мышь, монитор)

4.1 Решаемые задачи

В данном разделе представлено предложенное решение задач разработки структур данных, методов сбора и хранения данных, методов предобработки и управления данными поведенческой биометрической информации об особенностях работы со стандартными устройствами ввода-вывода.

Для решения задач статической аутентификации и фоновой идентификации сначала необходимо произвести сбор биометрических образцов пользователя. В качестве биометрического образца выступают какие-либо характеристики, описывающие динамику взаимодействия пользователя со стандартными устройствами ввода-вывода. Далее из биометрических образцов производится выделение необходимых уникальных характеристик и построение поведенческой модели пользователя. Если рассматривать задачу биометрической аутентификации как задачу машинного обучения, то на данном этапе из биометрических образцов производится предобработка данных и формирование модели представления. По полученной модели с помощью алгоритма машинного обучения может быть построена целевая решающая функция. Далее, на этапе применения модели новый биометрический образец сравнивается с моделью пользователя, в качестве которого производится попытка аутентификации или идентификация. В результате такого сравнения принимается решение, принадлежит ли новый образец пользователю.

4.2 Разработка структур данных поведенческой биометрической информации об особенностях работы со стандартными устройствами ввода-вывода

Согласно пункту ТЗ 4.1.2.1 процедура сбора данных должна обеспечивать:

– обработку события нажатия любых кнопок клавиатуры, включая служебные, с фиксацией времени нажатия и времени удержания кнопок;
– фиксацию траектории движения мыши в рамках выбранных приложений;
– функционирование процедуры сбора в фоновом режиме.

В связи с этим было принято решение о выборе следующих форматов сбора и хранения данных о динамике работы пользователя с клавиатурой и мышью:

Данные о фиксируемых событиях сохраняются в текстовые файлы, одна строка соответствует одному зафиксированному событию.

Для клавиатуры фиксируется следующая информация о событии:

– *key* — код клавиши,
– *action* — тип события:
    - 1 — клавиша нажата,
    - 2 — клавиша отпущена,
– *time* — время, когда произошло событие,
– *modifiers* — признак удержания клавиши-модификатора при наборе,
– *process* — имя процесса, в котором произошло событие.

Данные о событии формируются в строку в следующем порядке:

`key,action,time,modifiers,process`

Для мышки фиксируется следующая информация о событии:

– *button* — отображает, нажата ли кнопка мыши:
    - 0 — никакая кнопка не используется,
    - 1 — левая кнопка,
    - 2 — правая кнопка,
    - 3 — обе,
– *buttons* – отображает, какая кнопка мыши была нажата в предыдущий момент времени:
    - 0 — никакая кнопка не была нажата,
    - 1 — левая кнопка,
    - 2 — правая кнопка,
    - 3 — обе,
– *x* — горизонтальная координата мыши на экране,
– *y* — вертикальная координата мыши на экране,
– *wheel* — отображает, использовалось ли колесо прокрутки:
    - 0 — не использовалось,

- 1 — использовалось,

– *action* — тип действия:
  - 1 — зажата кнопка,
  - 2 — кнопку отпустили,
  - 4 — перемещение (кнопки не использованы),
– *time* — время, когда произошло событие,
– *process* — название используемого приложения (в виде пути к исполняемому файлу).

Данные о событии формируются в строку в следующем порядке:

```
button,buttons,x,y,wheel,action,time,process
```

4.3 Разработка методов сбора данных поведенческой биометрической информации об особенностях работы со стандартными устройствами ввода-вывода

Инструментарий для сбора данных разрабатывался для операционных систем семейства Windows, начиная с версии XP SP2 и выше.

Основной технической задачей при сборе данных является фиксация событий операционной системы, приходящих от клавиатуры и мышки. В соответствии с пунктом ТЗ 4.1.2.1 процедура сбора данных должна обеспечивать:

– обработку события нажатия любых кнопок клавиатуры, включая служебные, с фиксацией времени нажатия и времени удержания кнопок;
– фиксацию траектории движения мыши в рамках выбранных приложений;
– функционирование процедуры сбора в фоновом режиме.

Для того чтобы учесть поставленные требования было принято решение использовать технологию перехвата событий операционной системы, основанную на установке низкоуровневого хука с помощь функции WinAPI: SetWindowsHookEx, при этом устанавливаются перехватчики для следующих событий:

– EVENT_KEY_PRESSED,
– EVENT_KEY_RELEASED,
– EVENT_MOUSE_PRESSED,
– EVENT_MOUSE_RELEASED,
– EVENT_MOUSE_WHEEL,
– EVENT_MOUSE_MOVED.

Для событий клавиатуры фиксировался виртуальный код кнопки, тип события (нажатие или отпускание), время события, какие кнопки-модификаторы были нажаты, имя процесса в котором произошло событие.

Для мыши фиксировалась кнопка, тип события, время события и координаты курсора в момент события, имя процесса в котором произошло событие.

Таким образом, в рамках работ по разработке методов сбора данных, на языке C++ была реализована библиотека, устанавливающая хук и собирающая информацию о событиях, приходящих от клавиатуры и мыши. Собранная информация передается в пользовательское приложение, которое обрабатывает её и сохраняет на диск. Объем кода реализации составил около 3000 строк. Эта библиотека используется для обеих задач статической аутентификации и фоновой идентификации; приложения, обеспечивающие постобработку и сохранение информации, были реализованы для каждой задачи отдельно.

Сбор данных для решения задач статической аутентификации на основе биометрической информации о работе пользователя с мышью и клавиатурой осуществлялся по следующим сценариям:

1. ввод пароля на физической клавиатуре: фиксируется динамика работы пользователя клавиатурой;
2. ввод пароля на виртуальной клавиатуре: фиксируется динамика работы пользователя с мышкой;
3. ввод изображения (графической подписи) в область на экране: фиксируется динамика работы пользователя с мышкой;
4. преследование точки на экране: фиксируется динамика работы пользователя с мышкой.

Для сбора данных по этим сценариям были разработаны два приложения, одно для реализации первых трех сценариев, другое для реализации четвертого сценария.

Общий принцип работы первого приложения состоит в сборе активности мышки при попытке аутентификации пользователя, активности сохраняются в текстовые файлы: одна попытка, один файл с записью зафиксированных событий. Таким образом, набор файлов образует «модель» работы пользователя при выбранном способе аутентификации. На основе этих данных можно обучить классификатор и провести тестовую аутентификацию с отображением полученной меры сходства.

При добавлении новой модели можно задать имя модели, текст, который нужно будет вводить пользователю, минимальное количество обучающих вводов, после которого будет возможно тестирование модели, дообучать или нет модель в случае успешного ввода при тестировании модели, использовать или нет виртуальную клавиатуру (рисунок 20), включить

или нет режим графической подписи (рисунок 21). По умолчанию используется режим стандартного клавиатурного ввода последовательности символов, что соответствует первому сценарию сбора данных.

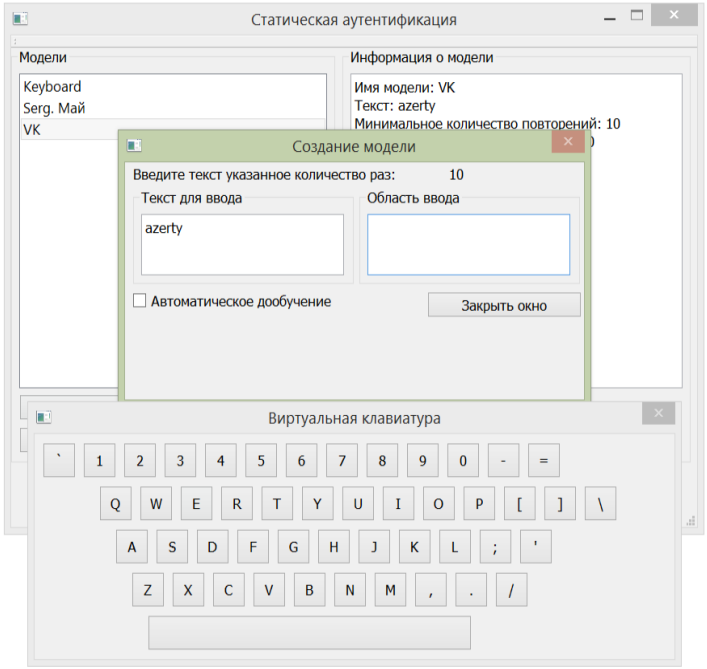


Рисунок 20 — Виртуальная клавиатура.

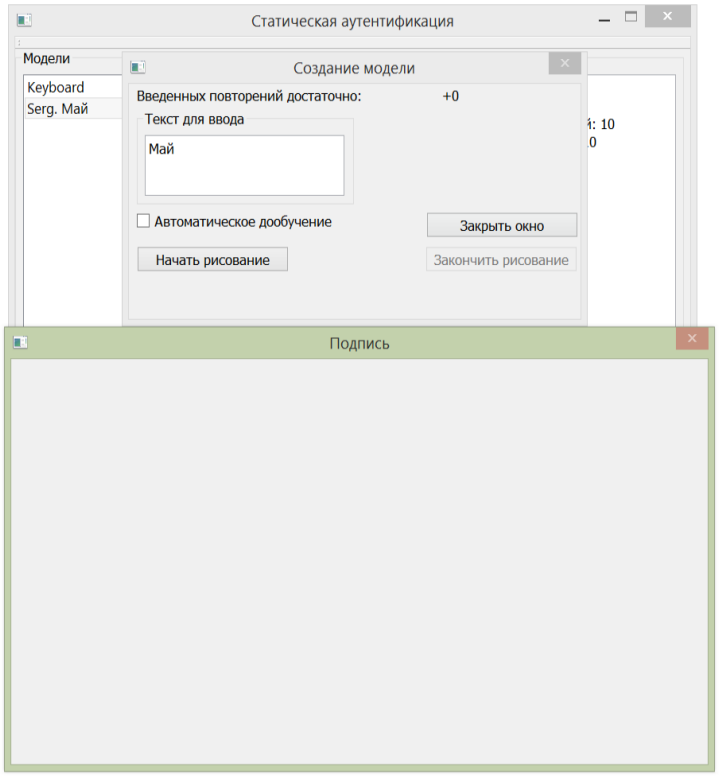


Рисунок 21 — Графическая подпись.

При включении режима виртуальной клавиатуры пользователю необходимо осуществить ввод пароля воспользоваться мышью, что соответствует второму сценарию сбора данных.

При включении режима использования подписи пользователь должен нарисовать предопределенное слово (графическую подпись) мышью в области приложения, что соответствует третьему сценарию сбора данных.

Для реализации четвертого сценария сбора данных для статической аутентификации было разработано приложение, представляющее собой графическую среду для слежения за точкой (см. рисунок 22). Пользователь должен осуществить мышью наведение на точку, которая появляется в области приложения; после успешного наведения появляется следующая точка. При прохождении траектории системные события динамики работы пользователя с мышью фиксируются с помощью приложения для сбора данных.

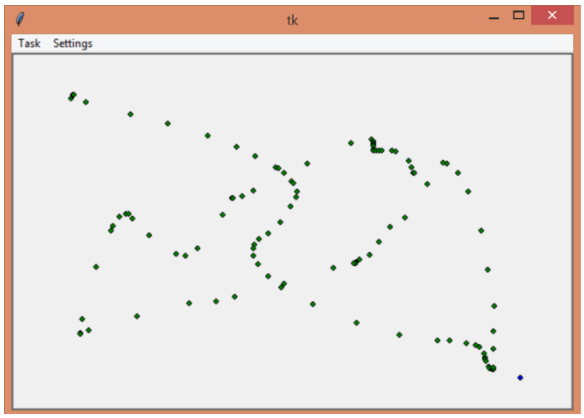


Рисунок 22 — Слежение за точкой.

Сбор данных для статической аутентификации на основе виртуальной клавиатуры, слежения за точкой и графической подписи осуществлялся на одной машине, в сборе участвовало 10 человек. На виртуальной клавиатуре пользователи вводили с помощью мыши слова длины от одного до двенадцати. Слова большей длины не пробовали, так как длинные пароли не удобны для пользователя и применяются редко. Траектория слежения за точкой, которую должен был пройти пользователь, включала в себя десять точек (в рамках одного образца ввода). Образцы для графической подписи включали в себя слова, длиной от одного до шести. При вводах пользователя приложением для сбора данных фиксируются системные события динамики действий пользователя с мышкой: нажатия и отпускания кнопок мышки, перемещения мышки. Каждое событие мыши характеризуется его типом, координатами курсора и временем, когда это событие произошло. Каждый пользователь делал 20 повторений ввода различных слов или прохождения траектории.

4.4 Разработка методов предобработки данных поведенческой биометрической информации об особенностях работы со стандартными устройствами ввода-вывода

В данном разделе представлены методы предобработки и представления данных динамики работы с клавиатурой и мышью для фоновой идентификации и статической аутентификации. В части методов предобработки рассматривается разбиение на временные окна, обработка пауз в действиях пользователя, обработка перехода пользователя между приложениями, обработка аномальных значений в распределениях производных признаков и нормализация признаков.

4.4.1 Разбиение на временные окна и фильтрация данных

Для задачи фоновой идентификации характеристики, описывающие динамику взаимодействия пользователя со стандартными устройствами ввода-вывода, рассчитываются по фрагментам собранной поведенческой информации, называемым временными окнами. Размер окна влияет на точность идентификации, поскольку окна большого размера более обобщенно представляют поведение пользователя и дают лучшее качество идентификации. В то же время для этого требуется большее чисто событий, что критично при анализе действий пользователя в режиме фоновой идентификации (злоумышленник может уже успеть завершить свои действия). Чтобы найти компромисс в этой ситуации, предлагается использовать перекрывающиеся на некоторое число действий окна. Таким образом, для принятия решения есть возможность проанализировать большее количество прогнозов и принять более точное решение (по сравнению с неперекрывающимися окнами), имея в распоряжении некоторое ограниченное количество событий.

Если в процессе выделения временного окна обнаруживается пауза в событиях, то часть событий до и после паузы предлагается относить к разным временным окнам. При этом, если в каком-либо из окон количество событий оказывается меньше некоторого порога, такое временное окно удаляется из анализа.

Если в процессе выделения временного окна пользователь перешел из одного приложения в другое, то часть событий до и после перехода может быть отнесена к разным временным окнам. При этом, если в каком-либо из окон количество событий оказывается меньше некоторого порога, такое временное окно удаляется из анализа.

Если при расчете значений признаков значение какого-то из признаков аномально, его можно заменить на усредненное значение по данному признаку. Для фильтрации использовались такие статистические меры как математическое ожидание и стандартное отклонение для оценки распределений значений признаков в соответствие с нормальными (Гауссовскими) распределениями.

4.4.2 Модели представления данных для задачи фоновой идентификации пользователей на основе динамики работы с клавиатурой

Как показал предварительный анализ, проведенный на прошлом этапе работ, основным методом представления динамики работы пользователя с клавиатурой в задачах фоновой аутентификации являются N-граммы, с N, не более 3. Для них считают различные статистические характеристики и формируют вектор признаков. В ряде работ клавиши разбивают на группы и считают характеристики для каждой группы отдельно. Методы, основанные на таких представлениях, показывают лучшую точность.

Стоит отметить, что методы представления данных, используемые при статической аутентификации, к данной задаче в большинстве своем не применимы, т.к. размерность их пространства признаков прямо пропорциональна длине вводимых данных.

В работе [43] авторы провели исследования и выяснили, что высокой описательной способностью обладают устойчивые N-граммы: были выделены основные N-граммы английского языка с N=2,3,4, но, чем выше N, тем больше требуется выборка для обучения, также необходимо знать язык, на котором пользователь осуществлял ввод, что не всегда возможно.

В связи с этим для случая фоновой идентификации был предложен модифицированный метод представления данных, который заключается в определении характерных (часто встречаемых) для пользователя N-грамм, вычислении для них среднего и стандартного отклонения и составлении из них вектора признаков. Этот метод, во-первых, позволяет более точно подстроиться под характер набора конкретного пользователя, а во-вторых, не зависит от используемого языка.

Подход на основе N-грамм учитывает не только характеристики нажатия отдельных клавиш, но и их порядок нажатия-отпускания. Тем не менее очевидно, что параметры, относящиеся только к одному нажатию, несут в себе меньше информации, нежели параметры, описывающие их серию. Поэтому также предлагается новый подход, основанный на потенциальных функциях [44], позволяющий сохранять недавнюю активность пользователя при работе с клавиатурой в рамках одного вектора признаков фиксированного

размера. Этот подход хорошо зарекомендовал себя в случае закрытого множества пользователей [44], когда заранее известна информация о потенциальных злоумышленниках, предлагается рассмотреть его применимость к случаю, когда такой информации нет.

4.4.2.1 Модель представления данных на основе N-грамм

N-грамма — последовательность из N символов. В задачах, связанных с анализом динамики работы пользователя с клавиатурой, под N-граммами подразумеваются характеристики, которые можно извлечь из пользовательского ввода, если рассматривать его как последовательность нажатий N клавиш.

Заметим, что подход на основе анализа характеристик одиночных нажатий можно рассматривать как частный случай N-граммы при N=1, однако, это будет не совсем корректно— понятие N-грамм (при N>=2) вводится, в первую очередь, для того, чтобы рассматривать нажатие не как последовательность независимых нажатий, а набор взаимозависимых в некоторой окрестности событий. N-грамму при N=2 обычно называют диграммой, см. рисунок 23, при N=3 — триграммой. Чаще всего в качестве признаков используют диграммы, реже триграммы.

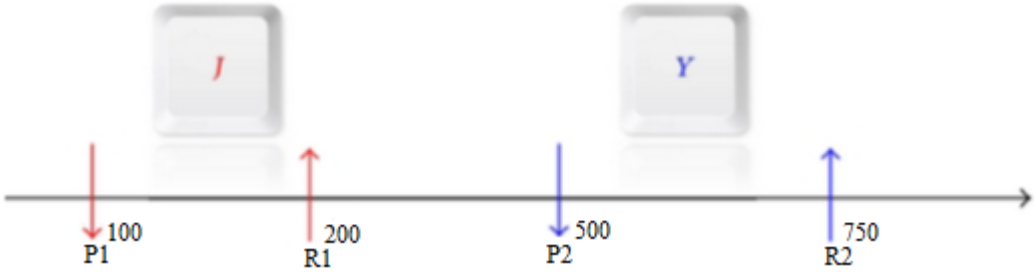


Рисунок 23 — Пример последовательного нажатия двух клавиш.

Диграмма обычно характеризуется следующими величинами:

– общим временем ввода диграммы (10):

$$(t^{total} = t_1^{down} - t_2^{up}); \quad (10)$$

– временем между нажатиями первой и второй клавиши (11):

$$(t^{dd} = t_1^{down} - t_2^{up}); \quad (11)$$

– временем между отпусканием первой и нажатием второй клавиши (12):

$$(t^{ud} = t_1^{up} - t_2^{down}); \quad (12)$$

– временем между отпусканием первой и отпусканием второй клавиши (13):

$$(t^{uu} = t_1^{down} - t_2^{up}), \quad (13)$$

где t_i^j — время i-го события (нажатие – down, отпускание – up).

Аналогично вводятся характеристики N-грамм больших размерностей. В вектор признаков обычно составляют из некоторых статистических величин, наиболее часто встречаются следующие:

– средняя длительность выбранной характеристики (14):

$$E_{ngr} = \frac{\sum_{i=1}^{n} t_{i_{ngr}}}{n} \quad (14)$$

– среднее отклонение выбранной характеристики от математического ожидания (15):

$$M_{ngr} = \frac{\sum_{i=1}^{n} \left| E_{ngr} - t_{i_{ngr}} \right|}{n} \quad (15)$$

– стандартное отклонение выбранной характеристики от математического ожидания (16):

$$D_{ngr} = \sqrt{\frac{\sum_{i=1}^{n} \left(E_{ngr} - t_{i_{ngr}} \right)^2}{n}} \quad (16)$$

где нижний индекс ngr обозначает соответствие величины выбранной N-граммы, а верхний предел суммирования n — количество N-грамм указанного типа в обрабатываемой последовательности.

4.4.2.2 Модель представления данных на основе потенциальных функций

Предлагаемое представление данных основывается на отображении (17):

$$\varphi = (A, t_d, t_h) \to H, A \in \Omega, \quad (17)$$

где Ω — некоторый конечный алфавит действий пользователя, t_d — время, в которое была нажата клавиша, t_h — время, в течение которого клавиша удерживалась нажатой, H — вектор признаков. Это отображение должно оказывать большее влияние на анализ динамики работы пользователя с клавиатурой в зависимости от следующих факторов:

– недавно нажатые клавиши;

– частота совершения нажатий;

– протяженность нажатий и пауз между ними.

Для построения такого отображения была использована теория потенциальных функций [45]. Пусть каждое возможное событие (нажатие или отпускание клавиш) $A_i \in \Omega$ имеет свой потенциал в момент t. Он убывает в зависимости от времени, прошедшего с момента совершения действия. Этот процесс характеризуется функцией $P_f: Time \times Time \to$

R. Если последовательность содержит два или более событий A_i, то их потенциалы суммируются.

Таким образом, можно определить отображение последовательности нажатий пользователя в L-мерный вещественный вектор, где $L = |\Omega|$, в виде (18):

$$\varphi(H(U), t) = \left(\sum_{(A,t_m)\epsilon H(u), t>t_m} Pf(t, t_m)\right)_{A\ \epsilon\Omega} \quad (18)$$

Согласно формуле (18), активность пользователя в каждый конкретный момент времени t может быть определена как множество из $L = |\Omega|$ потенциалов $\varphi_A(H(U), t)$. Такой подход учитывает как частоту предыдущих нажатий, так и время, в которое текущее нажатие было совершено.

Был выбран класс радиально-базисных функций, поскольку эти функции обладают необходимым в данном случае свойством: они зависят от интервалов между действиями, но не зависят от абсолютного времени их совершения. Кроме того, эти функции легко параметризовать для того, чтобы добиться эффективности в задачах аутентификации и идентификации.

В качестве потенциальной радиально-базисной функции была выбрана экспоненциальная функция $Pf(x, y) = e^{-\sigma\|x-y\|}$, где σ - коэффициент затухания, отвечающий за то, как быстро потенциал будет убывать. Кроме того, для конечной последовательности действий эта функция может быть рассчитана рекурсивно (19) (полагая $\varphi_A(0) = 0$):

$$\varphi_A(t_n) = \begin{cases} \varphi_A(t_{n-1}) \cdot e^{-\sigma}\|t_n - t_{n-1}\|, A\ \neq A_n \\ e^{-\alpha}, A\ = A_n \end{cases} \quad (19)$$

Для достижения максимальной эффективности коэффициенты σ и α могут быть подобраны, учитывая специфику конкретной задачи.

Коэффициент α отвечает за влияние времени нажатия на значение потенциала.

Для того чтобы избежать зашумленности данных низкими значениями, можно ввести константный порог ε следующим образом (20):

$$\widehat{\varphi_A(t_n)} = \begin{cases} \varphi_A(t_n), \varphi_A(t_n) \geq \varepsilon \\ 0,\ \varphi_A(t_n) < \varepsilon \end{cases} \quad (20)$$

Текущая активность пользователя описывается последовательностью n-мерных векторов (где $n = |\Omega|$), содержащих потенциалы $\widehat{\varphi_A}$ для каждого действия A.

В качестве примера рассмотрим набор слова «Hello» и полученные при этом векторы признаков. Для данного примера можно ограничить Ω буквами H,E,L,O. В таком случае векторы признаков будут четырехмерными и имеют вид, изображенный в таблице 15.

Таблица 15 — Получаемые векторы при наборе слова Hello

Событие	Состояние вектора			
	H	E	L	O
Начальное состояние	0	0	0	0
Нажата H	1	0	0	0
Нажата E	0.82	0	0	0
Нажата L	0.53	0.64	1	0
Нажата L	0.34	0.41	1	0
Нажата O	0.23	0.30	0.80	1

На графике на рисунке 24 видно, что элементы вектора представляют собой значения потенциальной функции соответствующего действия в моменты нажатий (по горизонтальной оси - время в сотых долях секунды, по вертикальной - значение потенциальной функции).

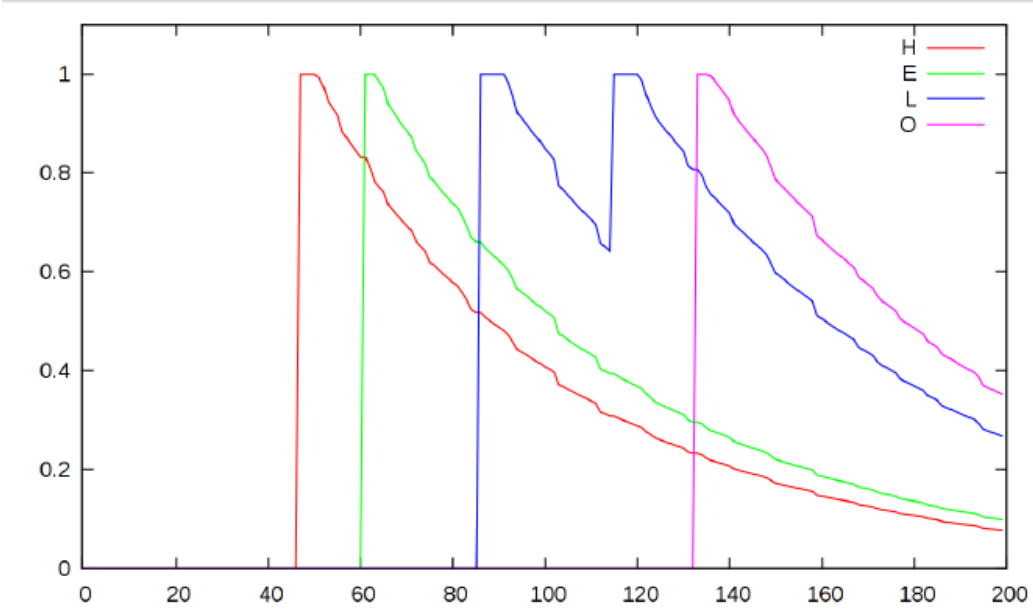


Рисунок 24 — Набор слова Hello

4.4.3 Модель представления данных для задачи фоновой идентификации пользователей на основе динамики работы с мышью

Для задачи фоновой идентификации была предложена модель представления, основанная на комбинации базовых статистических характеристик и на распределении шаблонов динамики действий пользователя при работе с мышью. Эти характеристики динамики работы пользователя рассчитываются по выделенным временным окнам.

Статистические характеристики динамики работы пользователя с мышью включают в себя следующие:

- среднее время клика,
- процент действий с мышью по направлениям (см. рисунок 25),
- процент пройденных расстояний по направлениям,
- процент времени перемещения мыши по направлениям,
- среднее пройденное расстояние при перемещениях мыши по направлениям,
- средняя скорость перемещений мыши по направлениям;
- средняя скорость в проекции на ось x по направлениям;
- средняя скорость в проекции на ось y по направлениям;
- средняя тангенциальная скорость по направлениям;
- среднее ускорение при перемещениях мыши по направлениям;
- распределение средней скорости в зависимости от пройденного расстояния;
- распределение среднего ускорения в зависимости от пройденного расстояния;
- распределение действий пользователя по областям экрана.

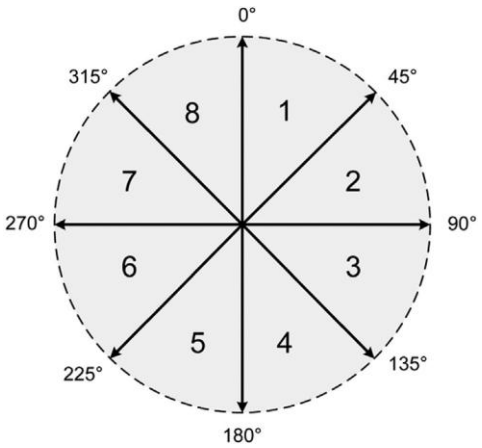


Рисунок 25 — Направления движения мыши [46].

Среднее время клика рассчитывается как среднее арифметическое времени всех кликов, выполненных в данном временном окне.

Процент действий с мышью по направлениям рассчитывается как отношение числа действий, выполненных в определенном направлении, к общему числу действий, выполненных в данном временном окне.

Процент пройденных расстояний по направлениям рассчитывается как отношение суммарного расстояния при перемещениях мыши в определенном направлении к

суммарному расстоянию при перемещениях мыши во всех направлениях (в данном временном окне).

Процент времени перемещения мыши по направлениям рассчитывается как отношение суммарного времени при перемещениях мыши в определенном направлении к суммарному времени при перемещениях мыши во всех направлениях (в данном временном окне).

Среднее пройденное расстояние при перемещениях мыши по направлениям рассчитывается как среднее арифметическое расстояний по всем направлениям при перемещениях мыши, выполненных в данном временном окне.

Средняя скорость перемещений мыши по направлениям рассчитывается как среднее арифметическое скоростей движения при перемещениях между двумя точками по всем направлениям (в данном временном окне). Скорость движения при перемещении между двумя точками рассчитывается по формуле (21):

$$V_i = \frac{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}{t_i - t_{i-1}}, \quad (21)$$

где x_i — координата точки i по оси x,

y_i — координата точки i по оси y,

t_i — момент времени прихода в точку i.

Средняя скорость в проекции на ось x по направлениям рассчитывается как среднее арифметическое скоростей движения в проекции на ось *x* при перемещениях между двумя точками по всем направлениям (в данном временном окне). Скорость в проекции на ось *x* рассчитывается по формуле (22):

$$V\mathrm{x}_i = \frac{\sqrt{(x_i - x_{i-1})^2}}{t_i - t_{i-1}}, \quad (22)$$

где x_i — координата точки i по оси x,

t_i — момент времени прихода в точку i.

Средняя скорость в проекции на ось y по направлениям рассчитывается как среднее арифметическое скоростей движения в проекции на ось *y* при перемещениях между двумя точками по всем направлениям (в данном временном окне). Скорость в проекции на ось *y* рассчитывается по формуле (23):

$$Vy_i = \frac{\sqrt{(y_i - y_{i-1})^2}}{t_i - t_{i-1}}, \quad (23)$$

где y_i — координата точки i по оси y,

t_i — момент времени прихода в точку i.

Средняя тангенциальная скорость по направлениям рассчитывается как среднее арифметическое тангенциальных скоростей движения при перемещениях между двумя точками по всем направлениям (в данном временном окне). Тангенциальная скорость рассчитывается по формуле (24):

$$Vt_i = \sqrt{Vx_i^2 + Vy_i^2} \quad (24)$$

Среднее ускорение при перемещениях мыши направлениям рассчитывается как среднее арифметическое ускорений при перемещениях между двумя точками по всем направлениям (в данном временном окне). Ускорение при перемещении между двумя точками рассчитывается по формуле (25):

$$a_i = \frac{V_i - V_{i-1}}{(t_i - t_{i-1})/2}. \quad (25)$$

При вычислении признаков распределения средней скорости и ускорения в зависимости от пройденного расстояния все дистанции подразделяются на три диапазон: короткие (меньше 300 пикселей), средние (300–600 пикселей) и длинные (больше 600 пикселей).

Распределение средней скорости в зависимости от пройденного расстояния рассчитывается как среднее арифметическое скоростей при перемещениях мыши на расстояния, попадающие в определенный диапазон (в данном временном окне).

Распределение среднего ускорения в зависимости от пройденного расстояния рассчитывается как среднее арифметическое ускорений при перемещениях мыши на расстояния, попадающие в определенный диапазон (в данном временном окне).

При вычислении признаков *распределения действий пользователя по областям экрана* экран условно подразделяется на девять областей равной площади (см. рисунок 26) и рассчитывается отношение числа действий в каждой из выделенных областей к общему числу действий, выполненных в данном временном окне.

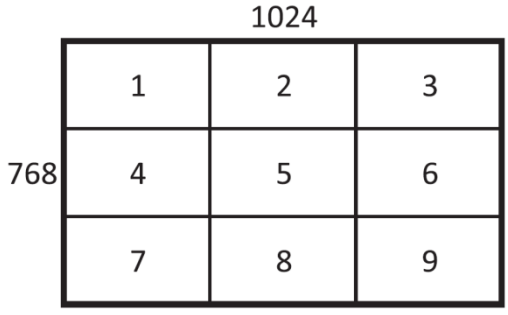


Рисунок 26 — Выделение областей нахождения курсора мыши на экране [47].

В качестве шаблонов поведения пользователя использовались диграммы и триграммы элементарных событий, то есть часто встречаемые последовательности событий длины два и три соответственно. В качестве характеристических признаков шаблонов рассчитывались частоты встречаемости этих последовательностей в рамках данного временного окна.

4.4.4 Модель представления данных для задачи статической аутентификации пользователей на основе динамики работы с клавиатурой

Для задачи статической аутентификации была предложена модель представления данных, основанная на комбинации двух признаков: времен удержания клавиш и временных промежутков между нажатиями и отпусканиями последовательных клавиш.

Так как в задаче статической аутентификации используется текст фиксированной длины, всегда можно построить вектор признаков конечного размера, описывающих каждый элемент текста. В данном случае он будет состоять из следующих элементов, рассчитывающихся для каждого вводимого символа и записанных последовательно (см. рисунок 27):

1. код нажатой клавиши;
2. время удержания нажатой клавиши;
3. временной промежуток между отпусканием предыдущей клавиши и нажатием текущей;
4. временной промежуток между нажатием предыдущей клавиши и нажатием текущей.

   Для первого нажатия признаки 3 и 4 не рассчитываются.

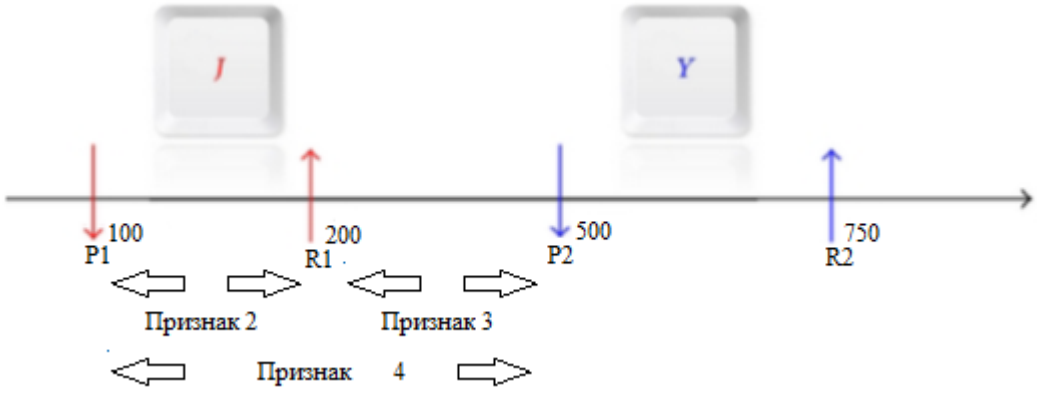


Рисунок 27 — Признаки метода представления данных для статической аутентификации по клавиатуре.

4.4.5 Модель представления данных для задачи статической аутентификации пользователей на основе динамики работы с мышью

Для задачи статической аутентификации было предложено применять модель представления на основе статистических признаков, описанную в пункте 4.4.3, для данных, собранных в среде слежения за точкой, и модель представления на основе сопоставления траекторий перемещения пользователем мыши методом динамической трансформации временной шкалы (DTW, dynamic time warping) [48, 49] для двух других сред сбора данных для статической аутентификации (виртуальная клавиатура и графическая подпись), описанных в подразделе 4.3.

Алгоритм DTW позволяет найти оптимальное соответствие между временными последовательностями. Для задачи статической аутентификации на основе событий работы пользователя с мышью в качестве временной последовательности используется вся траектория одного теста, выполняемого пользователем, а в качестве элементов сравниваемых временных рядов используется перемещение мыши.

Для вычисления отклонения между двумя последовательностями иногда бывает достаточно простого покомпонентного измерения расстояния (например, евклидово расстояние). Однако в реальных задачах часто две сравниваемые последовательности имеют примерно одинаковые общие формы, но эти формы не выровнены по оси времени. Метод DTW имеет существенное преимущество перед применением евклидова расстояния: если два временных ряда одинаковы, но один из них незначительно смещен во времени (вдоль оси времени), то евклидова метрика может посчитать, что ряды сильно отличаются друг от друга (см. рисунок 28). Чтобы сравнить две последовательности мы должны деформировать ось времени одной или обеих последовательностей. Алгоритм DTW позволяет осуществить данное преобразование временной оси и выполнить сравнение.

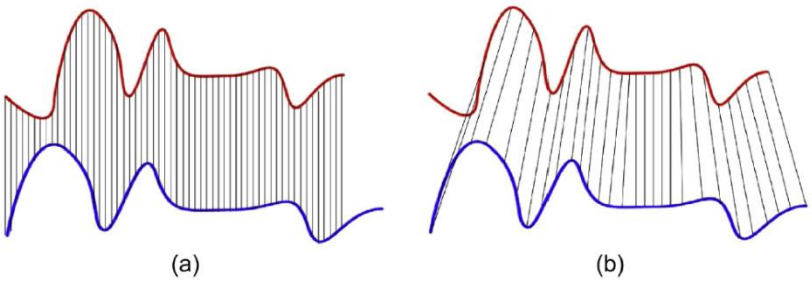


Рисунок 28 — Сопоставление двух последовательностей евклидовой метрикой (a) и алгоритмом DTW (b)[47].

На вход алгоритму DTWподаются два временных ряда ***Q*** длины ***n*** (26) и ***C*** длины ***m*** (27):

$$Q = q_1, q_2, \dots, q_i, \dots, q_n; \quad (26)$$

$$C = c_1, c_2, \dots, c_j, \dots, c_m. \quad (27)$$

Выходом является стоимость оптимального пути трансформации двух рядов, минимизирующего расстояние.

Алгоритм состоит из нескольких шагов:

1. Построение матрицы d порядка $n \times m$ (матрицы расстояний), в которой элемент d_{ij} – расстояние $d(q_i, c_j)$между двумя элементами последовательностей. В качестве метрики дистанции можно использовать евклидово расстояние ($d(q_i, c_j) = (q_i - c_j)^2$). При этом, каждый элемент d_{ij} матрицы d соответствует выравниванию между точками q_i и c_j. Для задачи аутентификации на основе динамики работы с мышью в качестве элементов последовательности используется расстояние.
2. Построение матрицы трансформации D, каждый элемент которой зависит от предыдущих:

   $$D_{ij} = d_{ij} + min(D_{i-1\,j}, D_{i-1\,j}, D_{i\,j-1}). \quad (28)$$

3. Построение оптимального пути трансформации и DTW расстояния (стоимости пути). Путь трансформации W – это набор смежных элементов матрицы трансформации. Он представляет собой путь $W = w_1, \dots, w_K$; $max(m, n) \leq K < m + n$ (K – длина пути), минимизирующий общее расстояние между Q и C (см. рисунок 29). Элементы последовательности определяются как $w_K = (i, j)_K$, вес элемента пути определяется как $d(w_K) = d(q_i, c_j)$. (29)

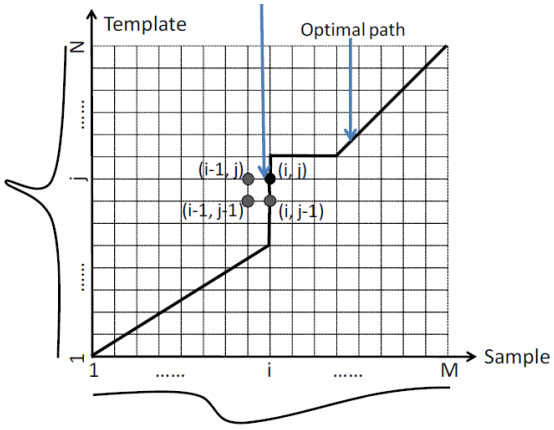


Рисунок 29 — Поиск оптимального пути на основе алгоритма DTW[50].

При этом путь трансформации должен удовлетворять следующим условиям:

– Началом пути является $w_1 = (1, 1)$, а концом $w_K = (n, m)$. Это гарантирует, что путь трансформации содержит все точки обоих временных рядов.
– Любые два смежных элемента пути W удовлетворяют неравенствам: $w_i - w_{i+1} \leq 1$ и $w_j - w_{j+1} \leq 1$. Это условие будет гарантировать, что путь трансформации будет являться непрерывным.
– Любые два смежных элемента пути $w_k = (i, j)_k$ и $w_{k-1} = (i, j)_{k-1}$ удовлетворяют следующим неравенствам: $i_k - i_{k-1} \geq 0$ и $j_k - j_{k-1} \geq 0$. Это условие будет гарантировать, что путь трансформации не возвращается назад к пройденной точке (оба индекса i и j не уменьшаются).

Стоимость пути трансформации называется DTW-расстоянием и рассчитывается исходя из оптимального пути W по формуле (30):

$$DTW(Q, C) = min\left\{\frac{\sum_{k=1}^{K} w_k}{K}\right\} \quad (30)$$

Число K в знаменателе используется для учета того, что пути трансформации могут быть различной длины.

4.5 Разработка методов хранения и управления данными поведенческой биометрической информации об особенностях работы со стандартными устройствами ввода-вывода

Для решения задач статической аутентификации и фоновой идентификации на основе данных поведенческой биометрической информации об особенностях работы со стандартными устройствами ввода-вывода использование сложных схем передачи и хранения данных, таких как реляционные базы данных или облачные хранилища, не требуется.

Собираемые данные о динамике работы пользователя с клавиатурой и мышью хранятся на файловой системе в виде текстовых файлов. Данные собираются с помощью библиотеки, перехватывающей и анализирующей системные события, приходящие от клавиатуры и мыши с помощью установки хука командой WinAPI SetWindowsHookEx. Далее данные передаются клиентскому приложению, которое преобразует их в установленный формат и сохраняет на диск.

Запись в новый файл начинается при запуске приложения сбора, имя файла содержит дату и время начала сбора, тип собираемой информации – клавиатура или мышь, и имя

пользователя, под которым происходил сбор. При остановке приложения сбора запись в файл заканчивается и файл закрывается. Сбор данных с клавиатуры и мыши можно активировать как вместе, так и по отдельности.

В ходе процедуры сбора данные сохраняются локально на машине каждого пользователя. Для использования их в рамках экспериментов файлы были собраны с машин вручную и помещены в одну директорию. Далее, данные, относящиеся к каждому пользователю, были разбиты на части в соответствии с условиями экспериментов, описанных в разделе 5, поданы на вход методам предобработки и построения моделей представления. Полученные модели также сохранялись на файловой системе и использовались для обучения и построения моделей классификации, которые сохранялись для дальнейшего применения в процессе классификации данных других пользователей.

4.6 Выводы

В данном разделе представлены результаты теоретических исследований по разработке структур данных, методов сбора, предобработки, хранения и управления для поведенческой биометрической информации об особенностях работы со стандартными устройствами ввода-вывода (клавиатура, мышь, монитор)

С учетом специфики рассматриваемых проблем и требований ТЗ была проведена декомпозиция задачи на следующие подзадачи: разработка средств для сбора данных динамики работы пользователя с мышью и клавиатурой, разработка сред для реализации сценариев сбора данных, предобработки и хранения собранных данных.

Для задачи статической аутентификации на основе динамики работы пользователя с клавиатурой предлагается осуществлять сбор событий динамики работы пользователя в специальной среде для ввода контрольных фраз, для мыши – в специальной графической среде для ввода образца (графической подписи).

Для задачи фоновой идентификации пользователя на основе динамики работы с клавиатурой и мышью предлагается в процессе обычной работы пользователя за компьютером собирать системные события нажатия клавиш клавиатуры и мыши и движений мыши.

Результаты, полученные в рамках настоящего раздела, подтверждают соответствие требованиям пп. 2.1.1, 3.4, 4.1.1.1 ТЗ.

5 Разработка методов машинного обучения и математической статистики для построения и применения поведенческих моделей на основе биометрической информации об особенностях работы со стандартными устройствами ввода-вывода (клавиатура, мышь, монитор)

5.1 Решаемые задачи

В данном разделе представлено предложенное решение для построения моделей и методов машинного обучения и математической статистики для задач статической аутентификации и фоновой идентификации пользователей на основе биометрической информации работы пользователя с клавиатурой и мышью.

Основные подзадачи, которые необходимо рассмотреть при построении решения следующие:

- предобработка данных;
- выбор метода классификации;
- выбор метода принятия решения.

5.2 Применяемые методы машинного обучения и математической статистики

Для экспериментального сравнения методов классификации были выбраны одноклассовые методы, являющиеся представителями различных типов методов машинного обучения:

- Метрические (метод ближайших соседей [50]).
- Вероятностные (метод опорных векторов [50]).
- Эвристические (метод репликаторных нейронных сетей [51, 52], нечеткий метод на основе потенциальных функций [45]).

Эти методы имеют свои достоинства и недостатки, и качество их работы может существенно зависеть от специфики прикладной задачи. Более подробно метрические и вероятностные методы рассмотрены в пункте 2.1.4 в [1].

Одним из достоинств метода ближайших соседей является возможность использования различных метрик, позволяющих подстроить его под конкретную задачу, а также небольшое количество параметров. Основной особенностью метода ближайших соседей является отсутствие у этого алгоритма стадии обучения, таким образом, достоинством такого подхода является возможность обновлять обучающую выборку без переобучения классификатора. Однако данный алгоритм осуществляет сравнение каждого анализируемого объекта со всеми объектами из обучающей выборки и поэтому главный недостаток метода k ближайших соседей заключается в длительности времени работы классификатора на этапе обучения и необходимости хранения всего обучающего набора.

Метод опорных векторов [50] для многих прикладных задач демонстрирует лучшие по точности результаты, по сравнению с другими методами машинного обучения. Для классификации данным методом достаточно лишь части обучающего набора данных (опорных векторов). Однако, скорость обучения данного алгоритма одна из самых низких. Метод опорных векторов также требует большого объема памяти и значительных затрат машинного времени на обучение.

Эвристические методы являются достаточно гибкими и позволяют моделировать очень сложные взаимосвязи между входными и целевыми переменными, благодаря чему часто демонстрируют хорошую точность. К таким методам можно отнести нечеткий метод поиска исключений, позволяющий преодолеть проблемы методов опорных векторов SVM, связанные с бинарной решающей функцией, зависящей от критических параметров, устанавливаемых априори [45]. Основная идея данного метода состоит в поиске одного общего нечеткого кластера, включающего все образы объектов исходного множества, таким образом, что степень принадлежности «основной части» высока.

Также к эвристическим методам можно отнести класс алгоритмов машинного обучения, известный как deep learning, которые пытаются моделировать высокоуровневые абстракции в данных, используя архитектуры, состоящие из множества нелинейных трансформаций. К этому классу алгоритмов можно отнести репликаторные нейронные сети, которые после построения модели не требуют хранения объектов обучающего набора [51, 52]. Однако данные методы имеют невысокую скорость работы при большом количестве входных признаков.

5.2.1 Метод ближайших соседей

В качестве конкретной реализации метода ближайших соседей было рассмотрено два варианта:

– классический: в тренировочном наборе L рассчитываются расстояния от каждого вектора до его K-го соседа, потом эти расстояния сортируются по возрастанию, полученный массив назовем L-dist. В качестве порога принятия решения о легитимности вектора берется значение элемента, имеющего позицию в массиве L-dist равную len(L-dist) * N, где N– процент, задающийся экспертно и влияющий на величину ошибки первого рода. Обычно это значение варьируется от 80 до 95%.
– модифицированный подход, основанный на локальной плотности векторов: пусть L – множество тренировочных векторов и E– вектор-кандидат. Для вычисления меры сходства вектора Eи набора L воспользуемся следующей процедурой:
  - находим K-го соседа вектора Eв тренировочном наборе L. Назовем этот вектор Z;
  - для вектора Zсчитаем сумму расстояний от него до каждого из K его соседей в тренировочном наборе L. Назовем полученную сумму Z_dist;
  - для вектора Eсчитаем сумму расстояний от него, до каждого из K его соседей в тренировочном наборе L. Назовем полученную сумму E_dist;
  - вычисляем дистанцию между тренировочным набором Lи вектором Z по формуле (31):

$$dist = \frac{|Z_{dist} - E_{dist}|}{\|Z\|} \quad (31)$$

Для определения порога легитимности разобьем случайным образом тренировочный набор L на два набора: L1 и L2, для векторов из набора L2 вычислим расстояния от каждого вектора до набора L1 и наоборот. Отсортируем полученный набор и воспользуемся процедурой, описанной для классического подхода с N в тех же границах. Полученный порог будем использовать для определения легитимности векторов из тестовых наборов.

5.2.2 Репликаторные нейронные сети

Данный подход основан на анализе отклонений и достаточно активно используется в задачах выявления исключений [52]. Репликаторная нейронная сеть (англ. Replicator Neural Networks, RNN) аппроксимирует функцию $f(x) = x$. Как правило, используется многоуровневая сеть, как минимум с тремя внутренними уровнями, последовательно

расположенными между входным и выходным (см. рисунок 30).

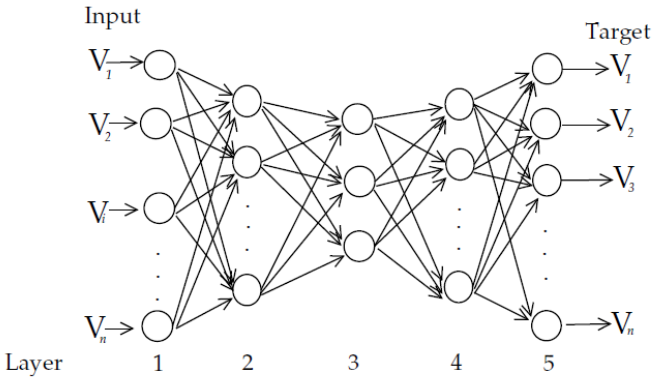


Рисунок 30 — Структура репликаторной нейронной сети [52].

Каждый узел, если он находится не на выходном уровне, соединен с каждым узлом следующего уровня. Соединение однонаправленное: выход узла меньшего уровня соединен с входом узла большего уровня. Входные и выходные данные – *n*-мерные числовые векторы. На каждый из *n* входных узлов поступает соответствующий элемент вектора, аналогичным образом полученный вектор считывается с выходных узлов. Каждому ребру моделирующего сеть графа принадлежит коэффициент w_{kij} , где *k* — номер уровня, *i* — номер узла на уровне *k*,*j* — номер связанного узла на уровне *k-1*.Сигнал,поступивший на выход узла меньшего уровня, приходит на вход узла большего уровня умноженным на коэффициент соединяющего эти узлы ребра. В каждом узле сети находится функциональный элемент S_k (θ), где θ — сумма всех входных сигналов узла. На все выходы узла поступает сигнал с выхода его функционального элемента, который, является одной из следующих функций (32):

$$
\begin{aligned}
&S_1(\theta) = \tanh(a_1\theta);\\
&S_2(\theta) = \tanh(a_2\theta);\\
&S_3(\theta) = \tanh(a_3\theta);\\
&S_4(\theta) = \tanh(a_4\theta);\\
&S_5(\theta) = a_5\theta.
\end{aligned}
\qquad (32)
$$

Здесь *N* и все a_k — параметры алгоритма. Задача функции S_3 заключается в расслоении множества возможных входных данных узла на *N* слоев.

Таким образом, количество возможных значений на выходах третьего уровня, а значит, и на последующих уровнях вплоть до выходного ограничено. Фактически репликаторная нейросеть производит кластеризацию анализируемого набора данных. Сеть в

результате обучается таким образом, чтобы выдавать центр ближайшего кластера для входного вектора. На выходе будет именно центр кластера или значение, близкое к нему, потому что формальная задача при настройке сети - минимизировать среднеквадратичную ошибку по всему набору данных.

Обучение нейронной сети происходит последовательно уровень за уровнем, при этом на каждом этапе производится оптимизация весовых коэффициентов только определенного уровня. В качестве метода оптимизации используется сопряженный градиент. Особенность этого класса нейронных сетей состоит в том, что элементы внутренних уровней можно использовать новое признаковое пространство, построенное на основе данных входного слоя. Процесс обучения нейронной сети начинается с инициализации десятью случайными точками, для которых выполняется так называемое предварительное обучение в течение десяти итераций. Далее каждый уровень нейронной сети обучается отдельно, первоначально используя оптимальные веса, полученные в результате предварительного обучения. Поочередно активируются связи между слоями, при этом обучение текущего слоя выполняется в течение максимально 1000 итераций, и затем выходы данного слоя используются в качестве входов следующего. После того как все уровни по отдельности обучены, производится результирующее обучение всех уровней нейронной сети, используя в качестве начальных весов оптимальные веса каждого уровня (обучение также выполняется в течении максимально 1000 итераций).

Исключениями в RNN методах считаются те объекты, для которых результат, выданный нейросетью, существенно отличается от входного – это означает, что точка находится далеко от центров кластеров. На самом деле алгоритм даже не разделяет все возможные точки на аномальные и нормальные, а выдает для каждой точки степень ее исключительности, основываясь на величине ошибки нейросети. Таким образом решающая функция $F_{of} := \|x - f_{RNN}(x)\|$.

5.2.3 Нечеткий метод поиска исключений с использованием потенциальных функций

Нечеткий метод поиска исключений позволяет преодолеть проблемы методов опорных векторов SVM, связанные с бинарной решающей функцией, зависящей от критических параметров, устанавливаемых априори [45]. Гибридный метод, использующий математический аппарат как теории нечетких множеств, так и потенциальных функций, может рассматриваться как нечеткий вариант метода SVM. Основная идея метода поиска

исключений близка идеям методов SVM, в частности также опирается на гипотезу о компактности (ГП), но вместо гиперплоскости или гиперсферы, описывающей компактное множество образов в пространстве характеристик, предлагается искать один общий нечеткий кластер, включающий все образы объектов исходного множества, таким образом, что степень принадлежности «основной части» высока. Размер этой «основной части» будем контролировать с помощью параметра η - радиуса нечеткого кластера. Это возможно либо напрямую задавая его, либо, определяя через число ожидаемых исключений k (используя подход, аналогичный SVM). Все объекты из исходного пространства X, с помощью потенциальной функции K неявно отображаются в пространство характеристик большой (или бесконечной) размерности. В результате объекты из X, образы которых лежат ближе к центру кластера в пространстве характеристик, будут иметь большую степень принадлежности. В этом случае степень принадлежности образа объекта нечеткому кластеру может рассматриваться как степень «типичности» объекта, то есть некоторая величина, противоположенная мере исключительности. Объекты с низкой степенью «типичности», меньшей предопределенного порога, будут считаться исключениями. Следует заметить, что изменение данного порога, то есть изменение критерия исключительности не приведет к необходимости заново применить алгоритм поиска исключений, как это происходит в методах SVM и метрических методах.

Рассмотрим более формально предложенный алгоритм. Определим целевую функцию:

$$J(u,a,\eta)=\sum_{i=1}^{N}(u_i)^m d_i^2(a)+\eta\sum_{i=1}^{N}(1-u_i)^m \qquad (33)$$

где $a=(a_1,...,a_N)\in\Re^N$, $u=(u_1,...,u_N)\in[0,1]^N$, $d_i^2(a)$ есть расстояние в пространстве характеристик от образа $\varphi(x_i)$ до a, такое что $d_i^2(a)=\left(\sum_{l,j}a_l a_j K_{lj}+K_{ii}-2\sum_j a_j K_{li}\right)$, m — параметр, определяющий скорость убывания значения степени принадлежности в зависимости от расстояния до центра кластера. Параметра η контролирует число «типичных» точек в кластере.

Степенью «типичности» произвольного объекта x из X будем называть значение функции $u(x)$, определенной следующим образом (34):

$$u(x)=\left[1+\left(\frac{\sum_{j=1}^{N}u_j{}^m\sum_{i=1}^{N}u_i{}^m K(x_i,x_j)}{\eta(\sum_{i=1}^{N}u_i{}^m)^2}-2\frac{\sum_{i=1}^{N}u_i{}^m K(x,x_i)}{\eta\sum_{i=1}^{N}u_i{}^m}+\frac{K(x,x)}{\eta}\right)^{1/(m-1)}\right]^{-1} \quad (34)$$

где u_j есть решение.

Объект $x\in X$ является исключением тогда и только тогда, когда его степень «типичности» $u(x)$ меньше заданного порога α. Таким образом, решающее правило принимает вид $F_{of}(x)=1-u(x)$. Используя терминологию теории нечетких множеств, множество исключений есть множество α-уровня нечеткого множества *U*, заданного на универсуме *X* функцией принадлежности $u(x)$.

Итак, при такой постановке задачи предложенный алгоритм поиска исключений на основе блочного покоординатного спуска можно записать в следующем виде (здесь *l* — номер итерации).

Шаг 0.Инициализация

$u_n^{(0)}=random([0,1])$ или $u_n^{(0)}=1$

$\eta^{(0)}=const$ (П31)

расчет матрицы значений потенциальной функции $\{K_{ji}\}_{1\le i\le N,1\le i\le N}$

Шаг 1. Расчет $a^{(l+1)}=\arg\min_{a\in\Psi_a} J(a,\eta^{(l)},u^{(l)})$

$$a_n^{(l+1)}=(u_n^{(l)})^m\Big/\sum_{i=1}^{N}(u_n^{(l)})^m$$

Шаг 2. Для всех *x* расчет расстояния до центра кластера**:**

$$d_n^2(a^{(l)})=\left(\sum_{i,j}a_i^{(l+1)}a_j^{(l+1)}K_{ji}+K_{nn}-2\sum_j a_i^{(l+1)}K_{ni}\right)$$

Шаг 3. Расчет $u^{(l+1)}=\arg\min_{u\in\Psi_u} J(a^{(l+1)},\eta^{(l+1)},u)$

$$u_n^{(l+1)}=\left[1+\left(d_n^2(a^{(l+1)})\big/\eta^{(l+1)}\right)^{1/(m-1)}\right]^{-1}$$

ЕСЛИ $\left\| u^{(l)} - u^{(l+1)} \right\| > \varepsilon$ **И** $l < L$ ТО **Шаг 1.** ИНАЧЕ **ВЫХОД.**

После расчета $u = (u_1, ..., u_N)$ для поиска исключений для любого x из X можно использовать функцию $u(x)$, вычисляющую нечеткую степень «типичности».

Вычислительная сложность алгоритма поиска нечеткого кластера в пространстве характеристик является квадратичной. Но с другой стороны, данный метод основывается на построении модели и поэтому может быть использован совместно с методами случайной выборки (sampling), что позволяет уменьшить сложность до линейной, естественно с потерей точности. Еще одной важной особенностью данного метода является то, что он не опирается на явные предположения о распределении вероятности. Кроме того, изменение критериев исключения не ведет к необходимости решать задачу поиска исключений заново, как это происходит в большинстве методов поиска исключений, включая методы SVM, метрические методы и другие. Это происходит потому, что в данном случае решающая функция не является бинарной. С другой стороны, как и все методы анализа данных, использующие потенциальные функции, результат зависит от используемой потенциальной функции и ее параметров. Данный метод не зависит от структуры используемых данных. Основным требованием является корректно определенная потенциальная функция, адекватно отражающая структуру и семантику анализируемых данных.

5.3 Экспериментальное исследование

В данном разделе приведены результаты экспериментального сравнения методов предобработки данных, моделей представления и методов классификации событий динамики работы пользователя с клавиатурой и мышью.

Для оценки качества классификации используется ROC-кривая (Receiver Operating Characteristic) — графическая характеристика точности классификатора, отражающая зависимость доли верных положительных классификаций от доли ложных положительных классификаций при варьировании порога решающего правила. Количественной интерпретацией ROC-кривой является показатель AUC (Area Under ROC Curve) — площадь, ограниченная ROC-кривой и осью абсцисс. Чем выше показатель AUC, тем качественнее классификатор. Помимо этого, часто используется такое понятие как EER (Equal Error Rate) — точка на ROC-кривой, в которой равны доли ошибочно отвергнутых собственных векторов и ошибочно принятых чужих векторов.

Для каждого пользователя проводится серия экспериментов с каждым из оставшихся. В качестве тестового набора в каждом из экспериментов используется незадействованная при обучении классификатора часть данных этого пользователя и такой же объём данных другого пользователя. При построении векторов признаков по исходной последовательности действий используется разбиение на основе окна, включающего фиксированное количество действий. Процент некорректно классифицированных векторов рассчитывается для каждой пары пользователей и затем усредняется по всему набору. Размеры окон для экспериментов выбирались экспериментально. Также стоит отметить, что при сравнении отдельных пар пользователей встречались пары, результаты которых сильно отличались как в лучшую, так и в худшую сторону от среднестатистической картины.

5.3.1 Эксперименты по предобработке данных динамики работы пользователя с клавиатурой для случая статической аутентификации

В данном пункте представлены результаты экспериментального исследования различных методов предобработки данных для задачи статической аутентификации пользователя на основе динамики его работы с клавиатурой.

Как было сказано в пункте 4.4.4 настоящего отчета метод представления, выбранный для решения данной задачи, состоит из следующих признаков: время удержания нажатой клавиши, временной промежуток между отпусканием предыдущей клавиши и нажатием текущей и временной промежуток между нажатием предыдущей клавиши и нажатием текущей. Эти признаки разнородны и могут лежать в разных границах, в связи с этим качество классификации может быть повышено за счет нормализации данных.

Предварительный анализ данных показал, что их распределение имеет длинный правый хвост. В связи с этим было принято решение попытаться заменить значения переменных на значения их логарифма: $\ln(x + C)$. Так как x — это времена удержания или паузы между различными фазами нажатия кнопок, то возможны ситуации, когда значение параметра x будет отрицательным. Такие случаи называют «перестановками»: если упорядочить нажатия клавиш по времени их нажатия и если одна клавиша была нажата, а затем была нажата другая клавиша, но первая при этом не отпускалась, то разница во времени между нажатием текущей и отпусканием предыдущей клавиши будет отрицательной. Поэтому параметр C необходимо выбирать так, чтобы $\ln(x + C)$ был определен. Было принято решение выбирать C равным максимуму из случаев перескока + 1,

если при попытке ввода появляется большее время перескока, то такая попытка признается не легитимной.

В ходе экспериментов проводилось исследование влияния применения методов нормализации на качество классификации. Были рассмотрены следующие методы:

- нормализация делением на квадратный корень из дисперсии;
- нормализация делением на значение абсолютного отклонения;
- нормализация делением на квадратный корень из абсолютного отклонения;
- нормализация делением на межквартильное расстояние (IQR);
- нормализация делением на медиану абсолютного отклонения.

Наилучший результат был получен при нормализации по абсолютному отклонению: пусть в N попытках обучения встречается признак p, тогда для этого признака коэффициент нормализации для вектора x будет иметь вид (35):

$$W_p = \sum_{i=1}^{N} \frac{|x_i - \bar{x}|}{N}, x'_p = {}^{x_p}/_{W_p} \quad (35)$$

где $\bar{x}$ — среднее арифметическое элементов вектора x, а x' — нормализованный вектор признаков p.

5.3.2 Эксперименты по предобработке данных динамики работы пользователя с клавиатурой для случая фоновой идентификации

В данном разделе представлены результаты экспериментального исследования различных методов предобработки данных для задачи фоновой идентификации пользователя на основе динамики его работы с клавиатурой. Будет рассмотрена стратегия подбора оптимальных параметров для представления, основанного на N-граммах.

Для выделения признаков данные разбиваются на наборы из 25, 50, 75 и 100 последовательных нажатий кнопок (далее - окон), вектора признаков строятся независимо, по каждой последовательности. Для каждой характеристики рассчитывается и среднее и среднеквадратичное времени её набора в окне.

Для формирования вектора признаков необходимо определить, какие характеристики могут повлиять на них.

Во-первых, для N-грамм необходимо определить порог частоты вхождения, этот порог должен динамически меняться в зависимости от размера обучающей выборки, чтобы «случайные» N-граммы не зашумляли модель и определить, на сколько сильно влияние аномально набранных N-грамм на качество классификации.

Во-вторых, это размер окна, как было сказано выше, в настоящих экспериментах будут рассмотрены размеры окон от 25 до 100.

В-третьих, необходимо определить размер сдвига окна. Если сдвигать окно на одно действие, то в модели будет очень много почти одинаковых векторов, что увеличит её объем, но не добавит описательной способности, если сделать шаг слишком большим, то может быть потеряна описательная способность модели. Эти параметры необходимо исследовать совместно, т.к. один размер сдвига может быть существенным для окна из 25 нажатий и не существенным для окна из 100 нажатий.

Таким образом, выбор параметров будет состоять из следующих шагов:

1. Определение оптимального значения порогов отсечения N-грамм по частоте.
2. Определение оптимального размера длины последовательности и размера сдвига окна при их формировании.

В качестве классификатора для определения оптимальных параметров будет использоваться SVM с полиномиальным ядром, degree = 3, nu = 0.05, т.к. он показывает высокую скорость работы и обладает достаточно высокой точностью.

5.3.2.1 Определение оптимального значения порогов отсечения N-грамм по частоте.

Для определения порога частоты вхождения N-граммы был произведен анализ данных из набора НЭДКЗ (см. пункт 1.1.6 ОВБС), содержащих от 5000 до 35000 действий. Более длинные наборы данных обрезались до заданных величин. Такой выбор размерности обуславливается тем, что не требует длительной работы пользователя в период обучения и обладает достаточной описательной способностью. Для каждого тестируемого пользователя набор признаков формировался индивидуально (частота ди- и триграмм рассчитывалась только на основе обучающей выборки).

Необходимо было выбрать порог таким образом, чтобы можно было, с одной стороны, доверять среднему и дисперсии, а с другой, излишне не сократить пространство признаков. В среднем набор данных содержит 700 различных диграмм, из них 130-140 встречаются чаще 20 раз. При увеличении порога количество диграмм резко сокращается (см. рисунок 31, 32). Для триграмм набор в среднем содержит 1500 уникальных элементов, из них 120-140 встречаются чаще 10 раз (см. рисунок 33, 34).

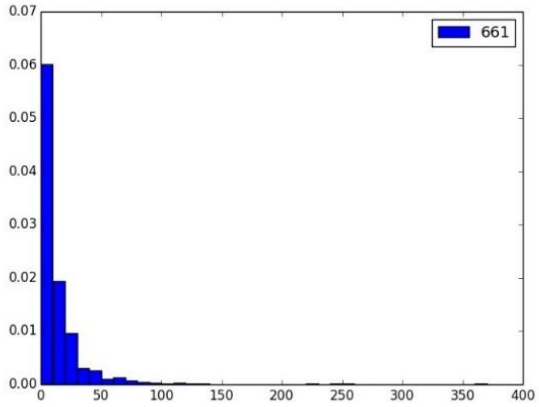


Рисунок 31 — Распределение частоты вхождения одинаковых диграмм.

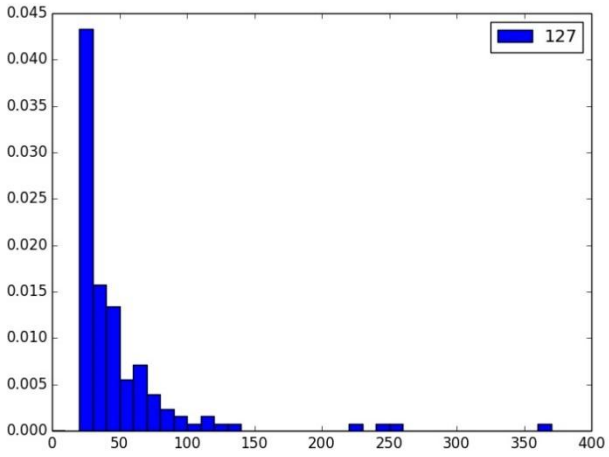


Рисунок 32 — Распределение частоты вхождения одинаковых диграмм, встречающихся более 20 раз.

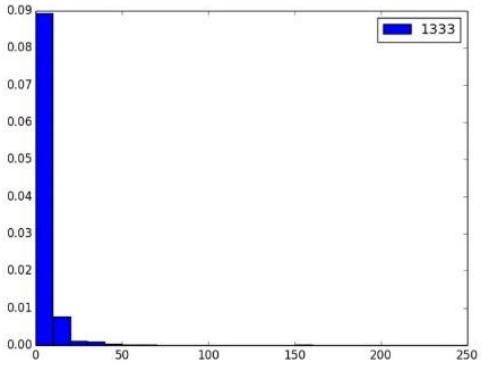


Рисунок 33 — Распределение частоты вхождения одинаковых триграмм.

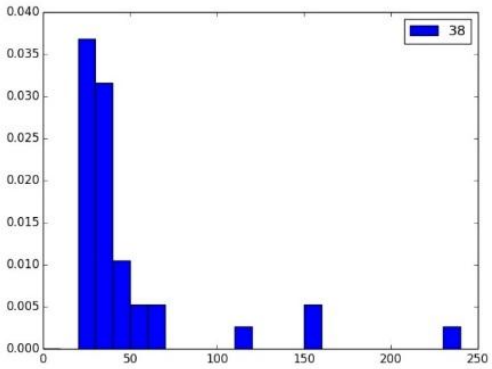


Рисунок 34 — Распределение частоты вхождения одинаковых триграмм, встречающихся более 20 раз.

Эксперименты показали, что при порогах ниже 10 качество классификации ухудшается, более высокие пороги можно использовать если в тренировочном наборе достаточно разнообразных N-грамм, что означает его значительный объем, в противном случае качество классификации тоже ухудшается. В ходе дальнейших экспериментов авторами были выбраны пороги от 0.03 до 0.012% от размера обучающей выборки (от 10 до 40 при размере обучающей выборки в 35000 действий).

5.3.2.2 Определение оптимального размера длины последовательности и размера сдвига окна при их формировании.

Для подбора параметров использовался набор данных НЭДКЗ (см. пункт 1.1.6 ОВБС), из него были выбраны данных двух человек, показывающие наихудшую точность сравнений среди всех остальных.

По итогам экспериментов было определено, что наилучшие результаты показаны при размере окна 75 и сдвиге равном 10 (см. таблицу 16):

Таблица 16 — Подбор размера окна и сдвига с использованием SVM (приведена характеристика AUC).

Размер окна/Сдвиг	1	2	5	**10**	20
25	65	65	65	66	61
50	67	68	69	69	69
75	71	69	70	**75**	71
100	72	72	74	75	73

5.3.2.3 Выводы

По итогам экспериментов для метода представления данных, основанного на часто встречаемых N-граммах было принято решение оставить для рассмотрения окна размером 75, со сдвигом 10, частота вхождения диграмм и триграмм варьировалась от 0.03 до 0.12 % от размера обучающего набора.

5.3.3 Эксперименты по предобработке данных динамики работы пользователя с мышью

В данном разделе представлены результаты экспериментального исследования различных методов предобработки для задачи фоновой идентификации. Характеристики, описывающие динамику работы пользователя с мышью, рассчитываются по фрагментам собранной поведенческой информации (временным окнам). Размер окна влияет на точность идентификации, поскольку окна большого размера более обобщенно представляют поведение пользователя и дают лучшее качество идентификации (см. рисунок 35). В то же время для этого требуется большее чисто событий, что критично при анализе действий пользователя в режиме фоновой идентификации. При использовании перекрывающихся окон для принятия решения есть возможность проанализировать большее количество прогнозов и принять более точное решение (по сравнению с неперекрывающимися окнами), имея в распоряжении некоторое ограниченное количество событий (см. рисунок 36).

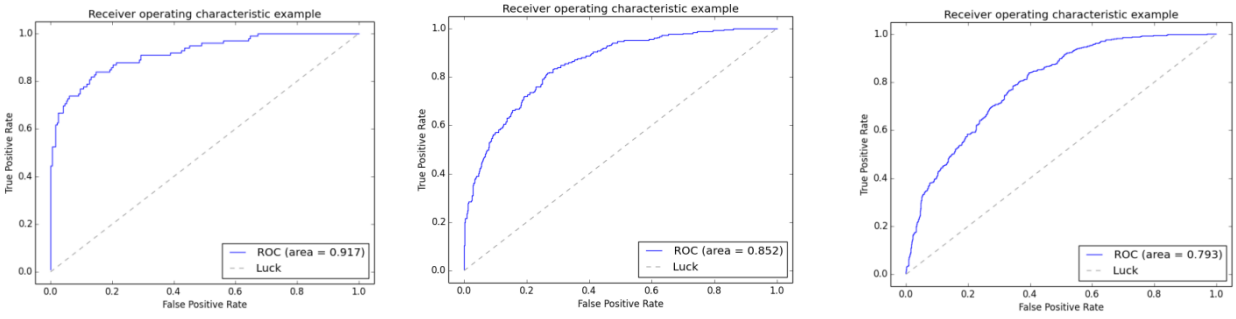


Рисунок 35 — Результаты сравнения различных размеров окон (слева — 500 событий, в середине — 200, справа — 100, метод классификации OneClassSVM: kernel='poly', degree=5, nu=0.05).

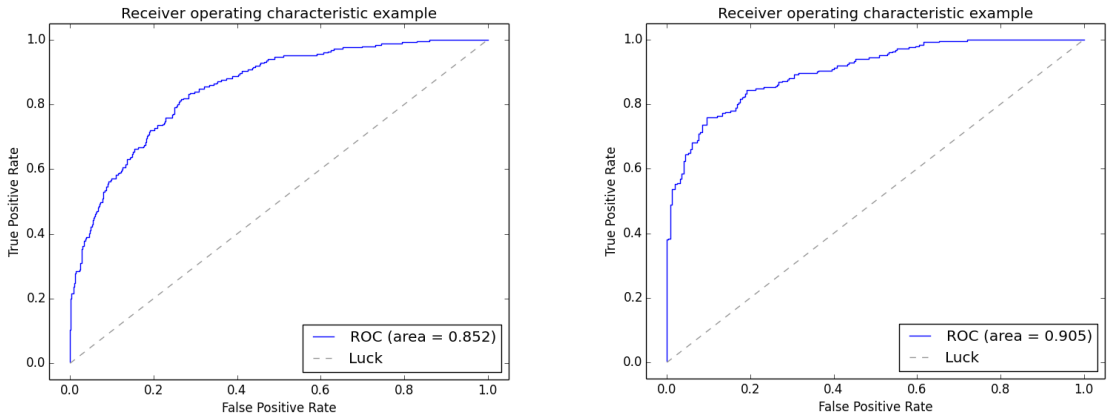


Рисунок 36 — Результаты сравнения выделение окон с перекрытием и без (слева — без перекрытия, справа — с перекрытием; размер окна — 200 событий, метод классификации OneClassSVM: kernel='poly', degree=5, nu=0.05).

При проведении экспериментального сравнения методов предобработки использовался набор данных, собранных в фоновом режиме с десяти пользователей при работе за компьютером с использованием мыши. В качестве модели представления используется комбинированная модель, включающая статистические характеристики действий при работе пользователя с мышью и распределение шаблонов поведения пользователя. В качестве алгоритма классификации используется метод опорных векторов как один из наиболее точных методов машинного обучения для широкого спектра задач (используемые параметры для SVM: kernel='poly', degree=5, nu=0.05).

Если в процессе выделения временного окна обнаруживается пауза в событиях, то часть событий до и после паузы предлагается относить к разным временным окнам. При этом, если в каком-либо из окон количество событий оказывается меньше некоторого порога, такое временное окно удаляется из анализа. Такая предобработка элементарных событий позволяет получить улучшение качества классификации (см. рисунок 37).

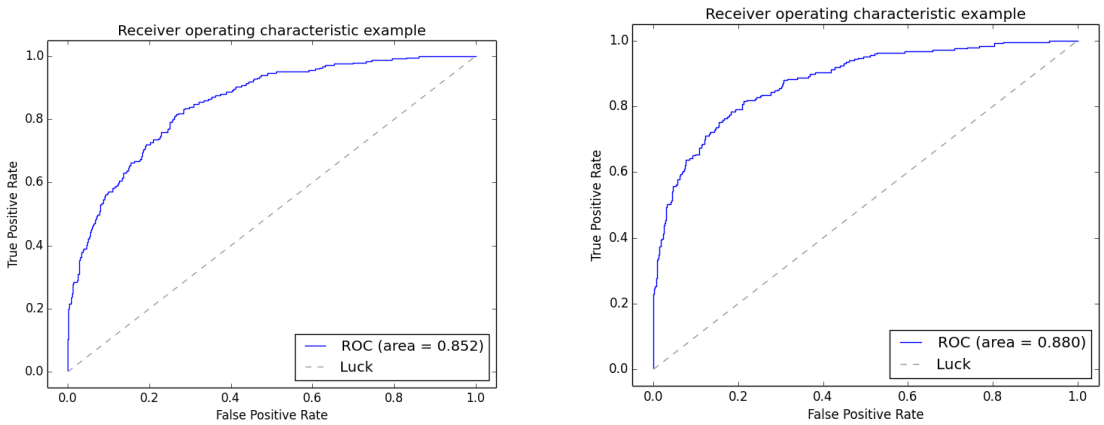


Рисунок 37 — Результаты сравнения обработки паузы в действиях пользователя (слева — без дополнительной обработки, справа — отнесение к разным окнам событий до паузы и после; размер окна — 200 событий, алгоритм классификации OneClassSVM: kernel='poly', degree=5, nu=0.05).

Если в процессе выделения временного окна пользователь перешел из одного приложения в другое, то часть событий до и после перехода может быть отнесена к разным временным окнам. При этом, если в каком-либо из окон количество событий оказывается меньше некоторого порога, такое временное окно удаляется из анализа.

Как показали эксперименты, обработка смены деятельности между приложениями (отнесение к разным окнам событий до и после перехода между приложениями без учета близости приложений по типу активности) не даёт улучшения точности (см. рисунок 38). Это может быть связано с тем, что во время перехода между приложениями пользователь совершает характерные действия, позволяющие классификатору фиксировать определённую специфику его поведения.

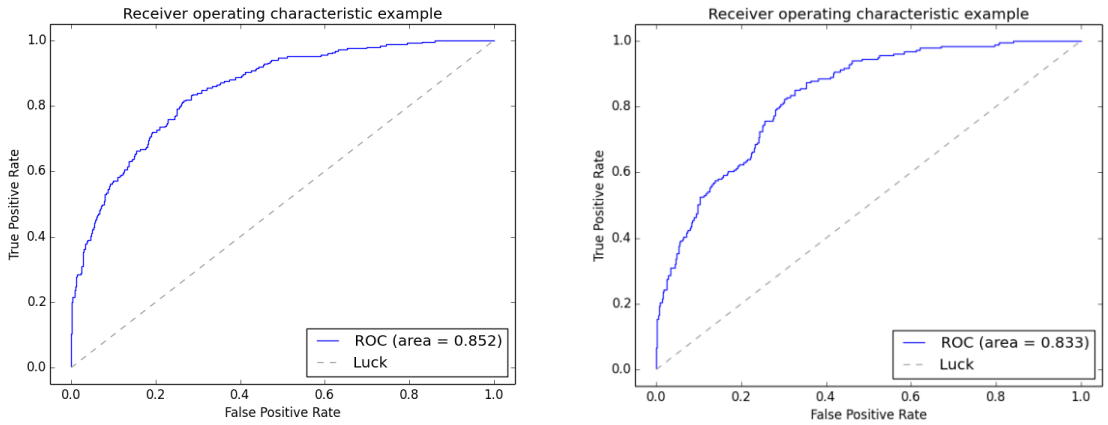


Рисунок 38 — Результаты сравнения обработки смены деятельности пользователя между приложениями (слева — без дополнительной обработки, справа — отнесение к разным окнам событий до и после перехода между приложениями; размер окна — 200 событий, параметры OneClassSVM: kernel='poly', degree=5, nu=0.05).

Если при расчете значений признаков значение какого-то из признаков аномально, его можно заменить на усредненное значение по данному признаку. Для фильтрации использовались такие статистические меры, такие как математическое ожидание и стандартное отклонение, для оценки распределений значений признаков.

Как показали эксперименты, устранение аномалий в распределениях признаков (посредством замены на усредненное значение тех признаков, значения которых отличаются от математического ожидания более чем на три стандартных отклонения) не даёт улучшения точности (см. рисунок 39). Это может быть связано с тем, что необычные значения в характеристиках динамики позволяют классификатору фиксировать определённую специфику его поведения, отличающую его от других пользователей.

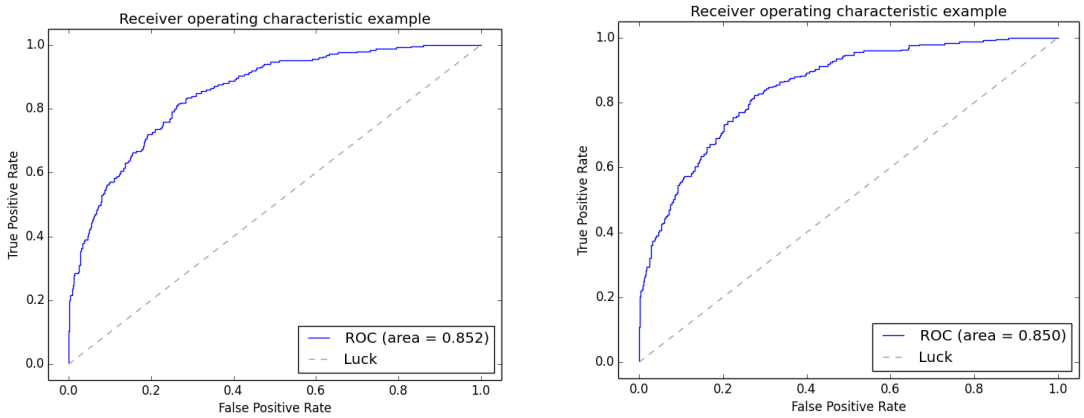


Рисунок 39 — Результаты сравнения устранения аномалий в распределениях признаков (слева — без дополнительной обработки, справа — замена на усредненное значение тех признаков, значения которых отличаются от математического ожидания более чем на три стандартных отклонения; размер окна — 200 событий, параметры OneClassSVM: kernel='poly', degree=5, nu=0.05).

5.3.4 Эксперименты по статической аутентификации на основе работы пользователя с клавиатурой

В данном пункте приведены результаты экспериментального сравнения качества работы различных методов классификации (kNN, SVM и нечеткий метод поиска исключений) для выбранных моделей представления в задаче статической аутентификации пользователей. Метод RNN не рассматривается в рамках задачи, т.к. требует значительного количества векторов для обучения, что потребует от пользователя большого количества (от нескольких сотен) обучающих вводов пароля. В качестве набора данных для экспериментального сравнения использовался набор данных, опубликованный в работе [53]. Основными его достоинствами являются:

– репрезентативность: в сборе данных участвовал 51 человек;
– соответствие решаемой задаче: данные представляют собой информацию о нажатиях клавиш во время набора сгенерированного случайным образом десятисимвольного пароля;
– значительное количество повторений: каждый из испытуемых набирал пароль 400 раз, что оказывается достаточным как для обучения классификатора, так и для его апробации.

Условия проведения эксперимента было решено взять из описания авторов набора данных:

- в качестве обучающего набора для каждого пользователя подаются 200 первых попыток набора пароля;
- для определения ошибки первого рода используются 200 следующих попыток пользователем ввода пароля;
- для определения ошибки второго рода используются по 5 попыток набора пароля 50-ю другими пользователями.

В процессе экспериментов варьировались следующие параметры алгоритмов:

- классический и модифицированные методы kNN: Кот 2*корень квадратный из размера обучающей выборки.
- ядро метода опорных векторов $kernel : rbf,\ linear,\ poly$;
- параметр степень полинома ядра $degree : 2,3,4,5,6$;
- параметр метода опорных векторов nu: от 0.1 до 0.99;
- параметр метода опорных векторов гамма: от 0.1 до 100;
- параметры нечеткого метода поиска исключений: их выбор описан наиболее подробно в отдельном разделе, так как метод показал наилучшие результаты.

5.3.4.1 Выбор параметров нечеткого метода поиска исключений

В качестве ядровых функций использовались наиболее популярные: скалярное произведение и ядро Гаусса. Использование скалярного произведения показало очень низкую точность, EER был в районе 0.3 и было исключено из дальнейшего рассмотрения.

Параметр ядра Гаусса σ варьировался в пределах от 0.1 до 20. При малых значениях ядра, порядка 0.1 – 1 качество классификации постепенно увеличивалось, но после достижения порога равного 10 снова начинало деградировать. На рисунках 40 и 41 отображены ROC-кривые при малых (от 0.1 до 1) и больших (от 1 до 20) значениях σ.

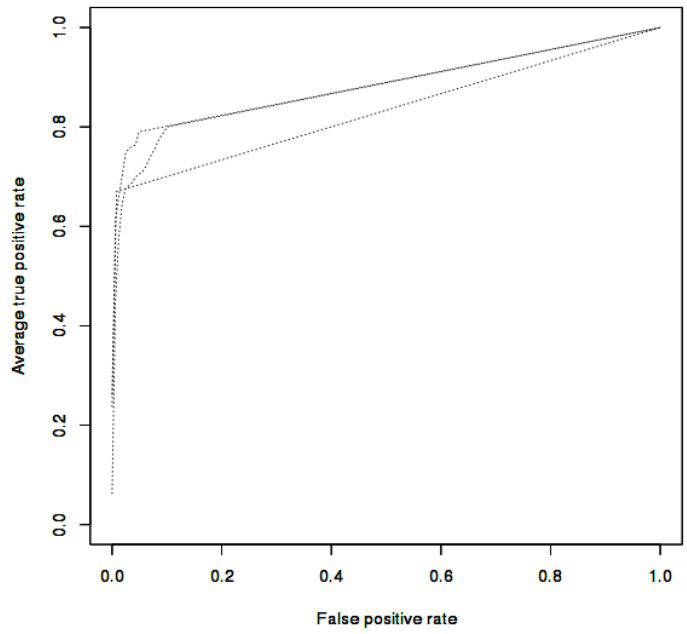


Рисунок 40 — ROC-кривые при малых значениях σ в ядре Гаусса.

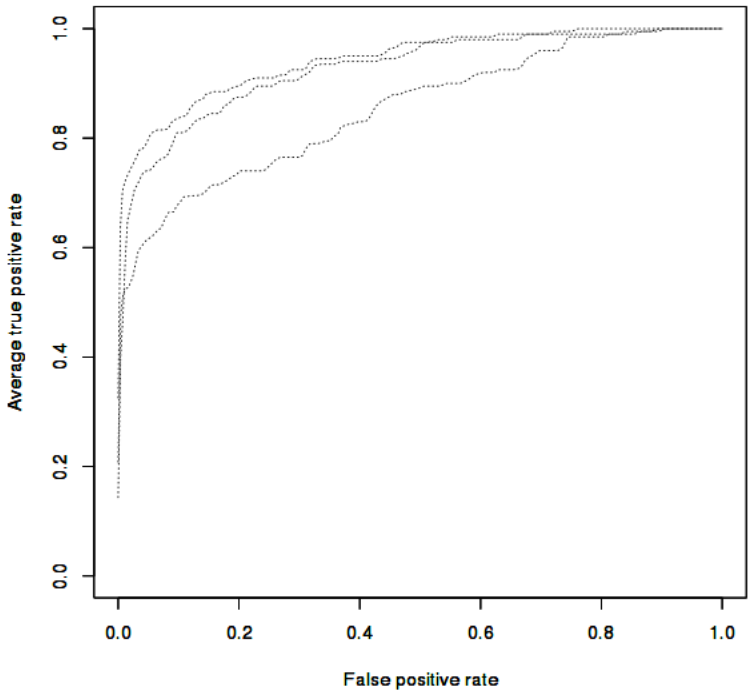


Рисунок 41 — ROC-кривые при больших значениях σ в ядре Гаусса.

Варьирование параметра степени нечеткости m в пределах от 1 до 10 не показало большого изменения точности, при значениях меньше 1 корректной классификации не происходит. В связи с этим было принято решение зафиксировать параметр m равным единице.

Подбор параметра η, являющегося расстоянием от центра кластера, при котором степень принадлежности считается равной 0.5 осуществлялся как упрощенным, так и итерационным способами. При использовании итеративного способа было необходимо задать долю выбросов. Эксперименты показали, что при доле выбросов равной 0.1 значение EERварьируется в границах 0.178-0.181, что является лучшим из полученных результатов. Использование вычисления логарифма от значений и нормализации по максимальному отклонению увеличивает полученную точность при применении метода до EER= 0.092.

На рисунке 42 приведены ROC-кривые различных пар пользователей, при упорядоченном выборе попыток ввода пароля.

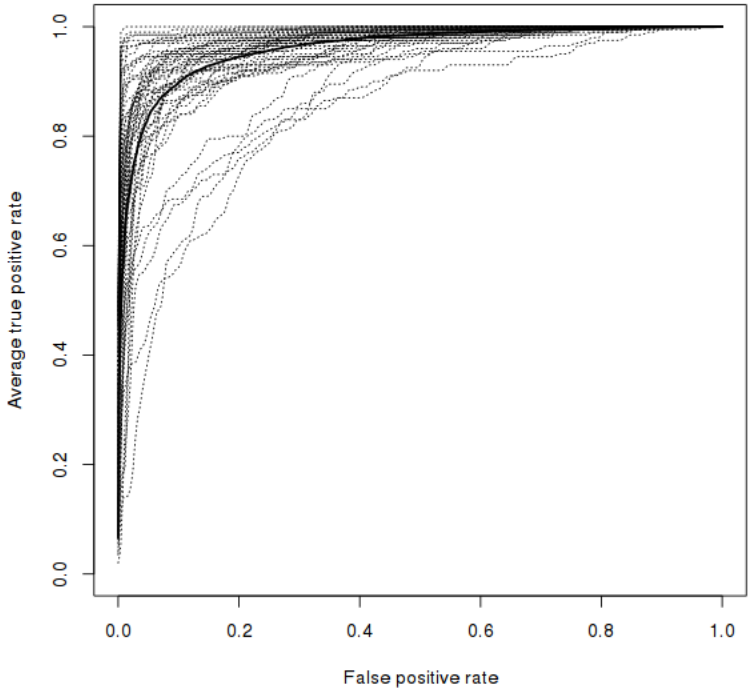


Рисунок 42 — ROC-кривые, полученные на упорядоченных данных с использованием предложенного алгоритма.

5.3.4.2 Результаты экспериментов

Таблица 17 — Результаты экспериментов по статической аутентификации для клавиатуры. Выбор наилучшего метода классификации.

Метод классификации	Параметры	EER (%)
классический kNN	K = корень квадратный из объема обучающей выборки, метрика – евклидово расстояние и манхэттен показали примерно одинаковые результаты	12.8
модифицированный kNN	K = корень квадратный из объема обучающей выборки, метрика – евклидово расстояние и манхэттен показали примерно одинаковые результаты	16.5
SVM	Ядро rbf. nu = 0.4. gamma = 46	11.2
Нечёткий метод поиска исключений	Параметр ядра Гаусса σ = 10, m = 1.5, η = 8	9.2

Как показывают эксперименты, лучшее качество классификации было получено при использовании нормализации и взятия логарифма от признаков для нечёткого метода поиска исключений, параметр EER = 9.2%.

Помимо рассматриваемого набора, авторы провели эксперименты на наборе НЭДК2 (см. подпункт 1.1.5.2 ОВБС). Результаты экспериментов с использованием нечеткого метода поиска исключений показали качество, несколько ниже, чем на первом наборе данных, EER = 0.18. Это может быть связано с тем, что было произведено всего по 100 попыток набора последовательности символов в разное время.

5.3.5 Эксперименты по статической аутентификации на основе работы пользователя с мышью

В данном подпункте приведены результаты экспериментального сравнения качества различных способов статической аутентификации, моделей представления и методов классификации.

Для экспериментов по статической аутентификации использовались следующие средства сбора данных, в которых пользователи осуществляли предопределённые действия с

помощью мыши (системные события работы с мышью фиксировались с помощью приложения для сбора данных):

- виртуальная клавиатура (пользователь осуществлял ввод заданного слова с помощью мыши);
- графическая среда для слежения за точкой (пользователь должен осуществить мышкой наведение на точку, которая появлялась в области приложения);
- среда для введения графической подписи (пользователи рисовали предопределенное слово мышкой в области приложения).

В таблице 18 и на рисунке 43 представлены результаты сравнения качества работы различных методов классификации на данных динамики работы пользователя, собранных при вводе пользователем мышью слова на виртуальной клавиатуре. Эксперимент проводился на примерах ввода десяти пользователей, каждый из которых осуществлял по двадцать вводов слов, длинной от одного до двенадцати. В качестве модели представления используется метод DTW для сопоставления траекторий (в качестве элементов сравниваемых временных рядов используется пройденная дистанция). Модель на основе базовых статистических признаков также применялась для данных, собранных при вводе пользователем мышью слова на виртуальной клавиатуре, но показала качество, хуже, чем метод DWT. Сравниваются следующие методы классификации: метод ближайших соседей (kNN), метод опорных векторов (SVM) и нечеткий метод поиска исключения. Метод репликаторных нейронных сетей не применялся в этой серии экспериментов, поскольку для его работы необходимы вектора признаков анализируемых объектов, а модель представления на основе DTW формирует уже матрицу расстояний между объектами. В качестве функции ядра для методов опорных векторов и нечеткого метода поиска исключений используется: $k(x,y) = exp^{\left(-\frac{DTW(x,y)}{\sigma^2}\right)}$.Выполнялся перебор следующих параметров методов:

- параметр ядра σ : 0.01, 0.1, 0.3, 0.4, 0.5, 0.6, 0.7, 0.9, 1, 2, 3, 5;
- параметр метода опорных векторов *nu*: 0.1, 0.3, 0.5, 0.7, 0.9, 0.91, 0.92, 0.93, 0.95, 0.97;
- параметр метода опорных векторов гамма: от 0.1 до 100;
- параметр метода ближайших соседей k: от 1 до корня квадратного из размера обучающей выборки;
- параметр нечеткого метода поиска исключений η: 0.8, 0.85, 0.9, 0.91, 0.92, 0.93, 0.95, 0.97.

По результатам экспериментов лучшее качество классификации при таком подходе сбора данных показывает метод опорных векторов ($\sigma = 0.4,\ nu = 0.9$) при сравнении траекторий введения шестибуквенных слов.

Таблица 18 — Эксперименты по статической аутентификации для собранных данных на виртуальной клавиатуры на основе сравнения траекторий с использованием метода DTW (10 пользователей, 20 примеров ввода шестибуквенных слов на каждого пользователя).

AUC (kNN), %	58.3
AUC (SVM), %	61.7
AUC (нечёткий метод поиска исключений), %	57.6

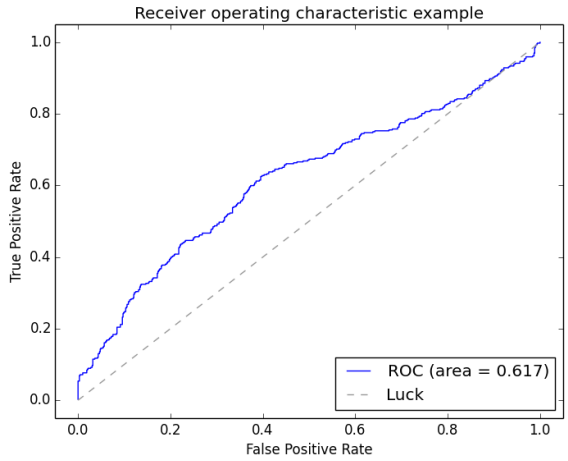


Рисунок 43 — ROC-кривые (FAR-FRR) экспериментального исследования методов статической аутентификации пользователей при работе мышью (данные ввода на виртуальной клавиатуре: 10 пользователей, 20 примеров ввода шестибуквенных слов на каждого пользователя, модель представления — DTW, алгоритм классификации — SVM).

В таблице 19 и на рисунке 44 представлены результаты сравнения качества работы различных методов классификации на данных динамики работы пользователя, собранных в приложении наведения курсора мышки на точку. Эксперимент проводился на образцах, собранных с десяти пользователей, каждый из которых осуществлял по двадцать проходов траектории слежения за точкой (включающей десять точек). В качестве модели представления используются статистические характеристики динамики работы с мышью, описанные в пункте 4.4.3. Модель на основе DWT также применялась для данных, собранных в среде слежения за точкой, но показала качество хуже, чем модель на основе

статистических признаков. Сравниваются следующие методы классификации: метод ближайших соседей (kNN), метод опорных векторов (SVM) и нечеткий метод поиска исключения. Выполнялся перебор следующих параметров методов:

– ядро метода опорных векторов $kernel: rbf,\ linear,\ poly$;
– параметр степень полинома ядра $degree: 2,3,4,5,6$;
– параметр метода опорных векторов *nu*: 0.1, 0.3, 0.5, 0.7, 0.9, 0.91, 0.92, 0.93, 0.95, 0.97;
– параметр метода опорных векторов гамма: от 0.1 до 100
– параметр метода ближайших соседей k: от 1 до корня квадратного из размера обучающей выборки;
– параметр нечеткого метода поиска исключений η: 0.8, 0.85, 0.9, 0.91, 0.92, 0.93, 0.95, 0.97.

По результатам экспериментов лучшее качество классификации при таком подходе сбора данных показывает метод опорных векторов ($kernel =' poly', degree = 5, nu = 0.05$).

Таблица 19 — Эксперименты по статической аутентификации для собранных данных перемещений мыши при слежении за точкой (10 пользователей, 20 примеров ввода на каждого пользователя для обучения, модель представления — статические характеристики работы пользователя с мышью).

AUC (kNN), %	65.1
AUC (SVM), %	76.7
AUC (нечёткий метод поиска исключений), %	63.2

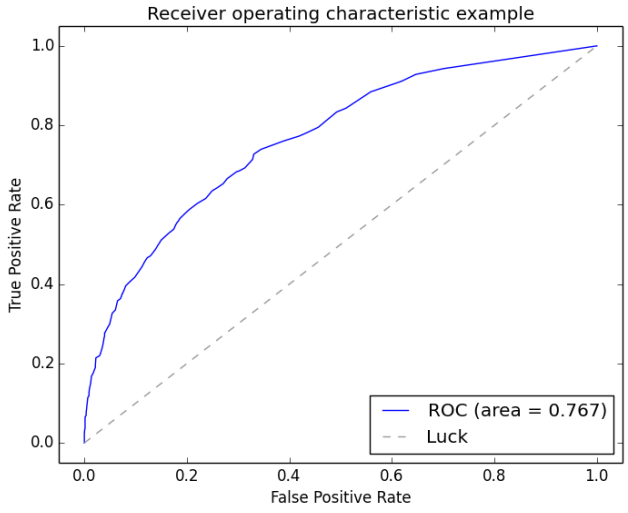


Рисунок 44 — ROC-кривые (FAR-FRR) экспериментального исследования методов статической аутентификации пользователей при работе мышью (данные перемещений мыши при слежении за

точкой, модель представления — статистические характеристики работы пользователя с мышью, алгоритм классификации — SVM, размер окна 500 событий).

В таблицах 20, 21, 22 и на рисунке 45 представлены результаты сравнения качества работы различных методов классификации на данных динамики работы пользователя, собранных в среде для ввода графической подписи. Эксперимент проводился на примерах ввода десяти пользователей, каждый из которых осуществлял по двадцать вводов слов, длиной от одного до шести. В качестве модели представления используется метод DTW для сопоставления траекторий (в качестве элементов сравниваемых временных рядов используется пройденная дистанция). Модель на основе базовых статистических признаков также применялась для данных собранных графических подписей, но показала качество, хуже, чем метод DWT. Сравниваются следующие методы классификации: метод ближайших соседей (kNN, *k=1*), метод опорных векторов (SVM) и нечеткий метод поиска исключения. Метод репликаторных нейронных сетей не применялся в этой серии экспериментов, поскольку для его работы необходимы вектора признаков анализируемых объектов, а модель представления на основе DTW формирует уже матрицу расстояний между объектами. В качестве функции ядра для методов опорных векторов и нечеткого метода поиска исключений используется: $k(x,y) = exp^{\left(-\frac{DTW(x,y)}{\sigma^2}\right)}$.Выполнялся перебор следующих параметров методов:

- параметр ядра σ : 0.01, 0.1, 0.3, 0.4, 0.5, 0.6, 0.7, 0.9, 1, 2, 3, 5;
- параметр метода опорных векторов *nu*: 0.1, 0.3, 0.5, 0.7, 0.9, 0.91, 0.92, 0.93, 0.95, 0.97;
- параметр метода ближайших соседей k: от 1 до корня квадратного из размера обучающей выборки;
- параметр нечеткого метода поиска исключений η:0.8, 0.85, 0.9, 0.91, 0.92, 0.93, 0.95, 0.97.

По результатам экспериментов лучшее качество классификации при таком подходе сбора данных показывает метод опорных векторов ($\sigma = 0.5$, $nu = 0.91$) при сравнении траекторий введения трёхбуквенных слов.

Таблица 20 — Эксперименты по статической аутентификации для собранных данных графической подписи на основе сравнения траекторий с использованием метода DTW (10 пользователей, 20 примеров ввода на пользователя, ввод четырёхбуквенных слов).

AUC (kNN), %	72.3

AUC (SVM), %	75.3
AUC (нечёткий метод поиска исключений), %	73.1

Таблица 21 — Эксперименты по статической аутентификации для собранных данных графической подписи на основе сравнения траекторий с использованием метода DTW (10 пользователей, 20 примеров ввода на пользователя, ввод трехбуквенных слов).

AUC (kNN), %	83.3
AUC (SVM), %	89.2
AUC (нечёткий метод поиска исключений), %	84.2

Таблица 22 — Эксперименты по статической аутентификации для собранных данных графической подписи на основе сравнения траекторий с использованием метода DTW (10 пользователей, 20 примеров ввода на пользователя, ввод двухбуквенных слов).

AUC (kNN), %	82.1
AUC (SVM), %	85.4
AUC (нечёткий метод поиска исключений), %	83.7

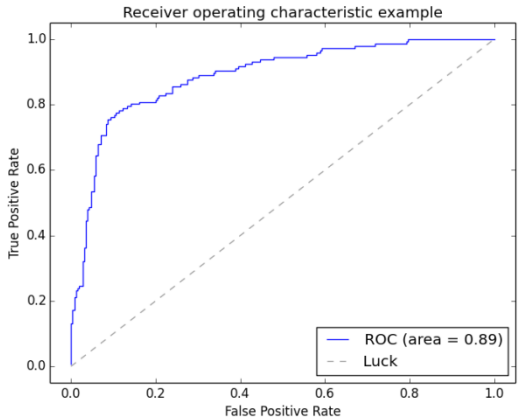


Рисунок 45 — ROC-кривые (FAR-FRR) экспериментального исследования методов статической аутентификации пользователей при работе мышью (данные трехбуквенной графической подписи, сравнение траекторий на основе DTW, алгоритм классификации SVM).

Итак, по результатам экспериментов по статической аутентификации лучшие результаты показал способ аутентификации на основе сравнения графических подписей пользователей при помощи сопоставления траекторий с применением метода динамической трансформации временной шкалы при построении модели представления и метода опорных векторов для классификации ($\sigma = 0.5,\ nu = 0.91$).

5.3.6 Эксперименты по фоновой идентификации на основе работы пользователя с клавиатурой

В данном пункте приведены результаты экспериментального сравнения качества работы различных методов классификации (kNN, SVM, RNN и нечеткий метод поиска исключений) для выбранных моделей представления в задаче фоновой идентификации пользователей. В качестве набора данных для экспериментального сравнения использовался собственный набор данных, собранных в фоновом режиме с 10 пользователей при работе за компьютером с использованием клавиатуры, НЭДКЗ (см. пункт 1.1.6 ОВБС)

Сравнивается качество работы следующих методов классификации с различными значениями параметров для выбранного представления данных:

- метод ближайших соседей:
  - номер ближайшего соседа k: от 1 до удвоенного корня квадратного из размера обучающей выборки.
  - метрики: евклидово расстояние, манхэттен.
- метод опорных векторов:
  - использовалось полиномиальное ядро со степенями от 1 до 20
  - *nu*: в промежутке от 0 до 0.99.
- нейронные сети:
  - Пусть N – размерность пространства признаков, а n – ближайшая степень двойки к числу N снизу. Тогда первые слои нейронов – это 2^n, вторые слои нейронов: 2^{n-1}, центральный слой – от 1 до 2^{n-2}. Если позволяет размерность пространства признаков, можно начинать не с ближайшего n, а с n-1, n-2 и т.д.
- нечеткий метод поиска исключений:
  - степень полинома ядра меняется в пределах от 1 до 30;
  - η в пределах от 0.01 до 1;
  - m: от 1.1 до 2.

5.3.6.1 Результаты экспериментов с представлением, основанным на выделении часто встречаемых N-грамм

В ходе предварительного анализа было определено, что оптимальными значениями для представления, основанного на N-граммах, являются следующие:

- размер окна: 75 действий.
- шаг сдвига окна — 10 действий.
- частота вхождения диграмм и триграмм в диапазоне от 0.03 до 0.12% от размера обучающего набора.

В рамках эксперимента необходимо определить оптимальный порог отсечения для диграмм и триграмм, а также выбрать алгоритм классификации показывающий наибольшую точность и его параметры (см. таблицу 23 и рисунок 46).

Таблица 23 — Результаты экспериментов по фоновой идентификации для клавиатуры. Выбор наилучшего метода классификации.

Метод классификации	Параметры	AUC (%)
Классический kNN	K = корень квадратный из объема обучающей выборки, метрика – евклидово расстояние и манхэттен показали примерно одинаковые результаты	64
Модифицированный kNN	K = корень квадратный из объема обучающей выборки, метрика – евклидово расстояние и манхэттен показали примерно одинаковые результаты	87
SVM	Степень полиномиального ядра = 8. Точность увеличивалась при увеличении nu от 0.5, по достижении 0.9 не изменялась	86
RNN	Индивидуально для каждого пользователя	81
Нечёткий метод поиска исключений	Степень полинома ядра: 6, m = 1.5, η = 0.8	83

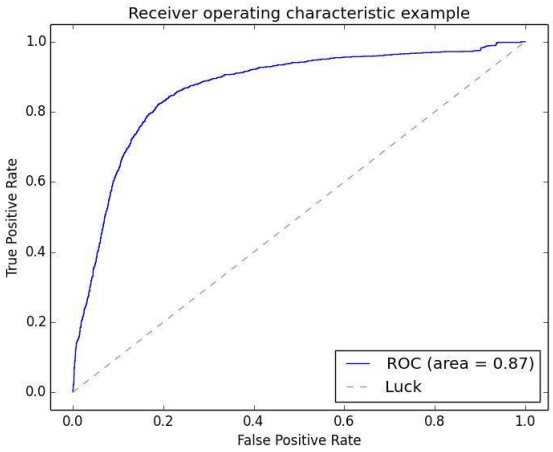


Рисунок 46 — ROC-кривая (FAR-FRR), метод представления основанный на частых N-граммах, алгоритм классификации – модифицированный kNN).

Как показали экспериментальные исследования, наилучшие результаты были получены при применении методов классификации модифицированного kNN и SVM при указанных параметрах. Модифицированный kNN хорошо подходит для представления данных, основанного на N-граммах. Это связано с тем, что векторы очень неоднородны. Если человек набирает текст, в векторе признаков будет много заполненных элементов, если человек читает документ, пролистывая его клавишами, то почти все элементы в векторе признаков будут нулевыми. Поэтому расчет дистанций для всех векторов сразу, как это делается в классическом методе, показывает низкую точность.

Стоит отметить, что уровень частоты вхождения ди- и триграмм существенно влияет на размерность пространства признаков, и, как следствие, на объем требуемой памяти и скорость работы классификатора, но результаты классификации меняются в пределах 3%, при изменении доли вхождения в указанных пределах.

5.3.6.2 Результаты экспериментов с моделью представления данных на основе потенциальных функций

В данном разделе представлены эксперименты с использованием метода представления, основанного на потенциальных функциях.

Параметрами представления являются:

- α — коэффициент отвечает за влияние времени нажатия на значение потенциала;
- σ — коэффициент затухания, отвечающий за то, как быстро потенциал будет убывать.

Кроме этого, необходимо определить минимальное количество последовательных действий (K), на основе которых строится вектор признаков и максимальную паузу T, между действиями, входящими в один вектор. Если пауза между действиями превысит T, счетчик действий K обнуляется.

В проводимых экспериментах минимальное количество действий K менялось в диапазоне от 25 до 250 с шагом в 25, максимальная пауза T в диапазоне от 20 до 100 секунд с шагом 10 секунд, коэффициент α варьировался в пределах от 0.5 до 2 с шагом в 0.1, коэффициент σ варьировался от 0.5 до 5 с шагом 0.5.

В первую очередь анализировалась связка параметров K и T. Эксперименты показали, что оптимальная точность достигается в районе K = 75 при T = 40, если параметр K увеличивать, то она стремится к AUC = 50% уже при K = 150. Полученный результат коррелирует с результатом, показанном при методе, основанном на N-граммах, там оптимальный размер окна тоже равен 75.

Как показали эксперименты, параметр σ всегда можно брать равным 1, при увеличении или уменьшении параметра качество классификации ухудшается. Параметр α можно увеличивать в связке с временем, чем больше α, тем больший промежуток времени можно дать пользователю для набора минимального количества клавиш K.

Дальнейшие исследования показали, что, варьируя параметры, можно получить высокую точность классификации для векторов признаков, полученных при активной работы пользователя с клавиатурой. С уменьшением количества нажатий, а также с увеличением повторяющихся нажатий в серии, качество ухудшается.

Точность классификации сильно зависит от выбранного подмножества действий, можно отметить, что даже если выбрать периоды с активной работой, метод, основанный на частых N-граммах будет показывать более высокий результат. На рисунке 47 приведен пример результатов классификации для одного набора данных, но с разными методами представления (на основе N-грамм и метода, основанного на потенциальных функциях). В качестве набора данных из НЭДКЗ (см. пункт 1.1.6 ОВБС) было выбрана пара пользователей, показывающих точность, выше средней.

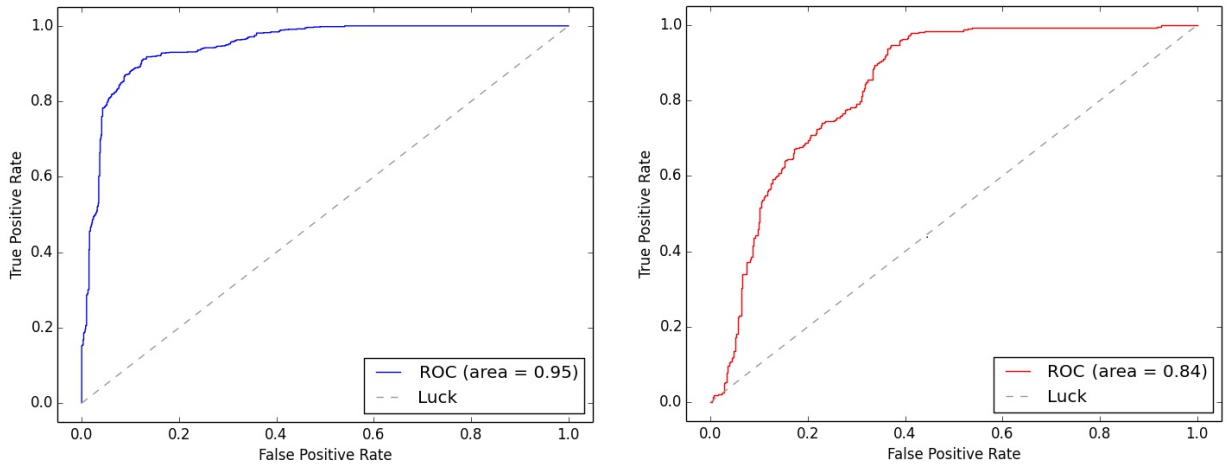


Рисунок 47 — Пример результатов классификации (классификатор- модифицированный kNN) векторов, полученных с помощью разных методов представления (на основе N-грамм и потенциальных функций).

5.3.7 Эксперименты по фоновой идентификации на основе работы пользователя с мышью

В данном подпункте приведены результаты экспериментального сравнения качества работы различных методов классификации для задачи фоновой идентификации на основе динамики работы с мышью. В таблице 24 и на рисунках 48, 49 представлены результаты проведенных экспериментов. Эксперименты проводились на наборе данных, собранных в течении нескольких недель в фоновом режиме с десяти пользователей при работе за компьютером с использованием мыши, НЭДМ2 (см. пункт 1.1.6 ОВБС).

В качестве модели представления используется комбинированная модель, включающая статистические характеристики действий при работе пользователя с мышью и распределение шаблонов поведения пользователя.

Таблица 24 — Результаты экспериментов по фоновой идентификации для данных, собранных с рабочих мест пользователей.

AUC (kNN), %	83.1
AUC (SVM), %	85.4
AUC (RNN), %	85.6
AUC (модифицированный kNN), %	86.2
AUC (нечёткий метод поиска исключений), %	86.9

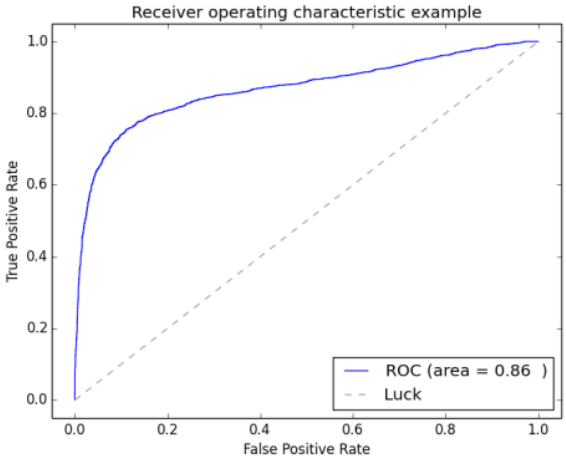


Рисунок 48 — ROC-кривые (FAR-FRR) экспериментального исследования методов фоновой идентификации пользователей при работе мышью (алгоритм классификации – модифицированный kNN, размер окна 500 событий).

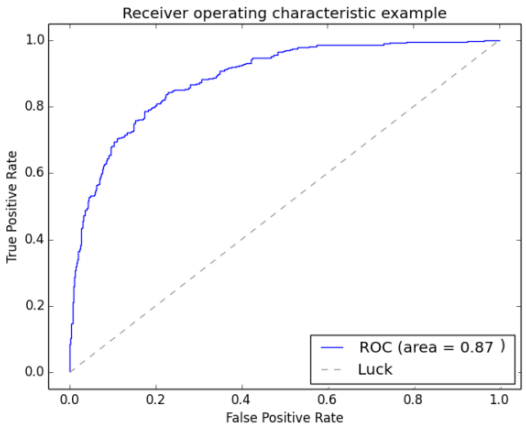


Рисунок 49 — ROC-кривые (FAR-FRR) экспериментального исследования методов фоновой идентификации пользователей при работе мышью (алгоритм классификации - нечеткий метод поиска исключений; размер окна 500 событий)

Сравниваются следующие методы классификации: метод ближайших соседей (kNN), модифицированный метод ближайших соседей, метод опорных векторов (SVM), нечеткий метод поиска исключений и репликаторные нейронные сети (RNN). Выполнялся перебор следующих параметров методов:

– ядро $kernel: rbf,\ linear,\ poly$;

– параметр степень полинома ядра $d: 2,3,4,5,6$;

– параметр метода ближайших соседей k: от 1 до корня квадратного из размера обучающей выборки;
– параметр метода опорных векторов *nu*: 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9;
– параметр метода опорных векторов гамма: от 0.1 до 100;
– параметр нечеткого метода поиска исключений η: 0.8, 0.82, 0.85, 0.87, 0.9, 0.91, 0.92, 0.93, 0.95, 0.97.
– параметр нечеткого метода поиска исключений *m*: 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2;
– архитектура нейронной сети: 32-16-2-16-32, 32-16-4-16-32, 32-16-8-16-32.

Итак, по результатам экспериментов лучшее качество классификации показали модифицированный метод ближайших соседей (с параметром k, равным корню квадратному из размера обучающей выборки) и нечеткий метод поиска исключений (с параметрами $kernel = 'poly', d = 4,\ \eta = 0.82, \mathrm{m} = 1.6$).

5.3.8 Описание комбинированных экспериментов по динамике работы пользователя с клавиатурой и мышкой

В настоящее время всё большее распространение стал получать комбинированный подход, при котором анализируется как динамика работы с мышью, так и динамика работы с клавиатурой [49, 54–56]. Данный метод позволяет учитывать больше индивидуальных особенностей пользователя и позволяет проводить аутентификацию непрерывно, независимо от того, с каким устройством ввода работает пользователь.

5.3.8.1 Цели и перечень проводимых экспериментов

Существует два основных подхода к комбинированному анализу работы с клавиатурой и мышкой [49, 55], в первом случае составляется общий вектор признаков и подается на вход классификатору, во втором вектора признаков формируются отдельно, проводится их независимая классификация и результаты передаются на вход решающей функции, которая, в свою очередь, тоже может быть классификатором.

Из-за специфики рассматриваемых в данной работе представлений мы остановимся на втором варианте. Будут взяты данные динамики работы с клавиатурой и мышкой за один период времени, рассчитаны векторы признаков, произведена их независимая классификация и получены оценки достоверности, которые будут поданы на вход решающей функции, описанной ниже, в хронологическом порядке событий, на которых оканчиваются вектора

признаков. Кроме этого, будет проведено «смешивание» ROC-кривых, как предложено в работе [55, 56].

Будут использоваться методы представления и алгоритмы классификации, признанные лучшими для клавиатуры и мышки.

Целью эксперимента является проверка гипотезы о том, что комбинированный анализ показывает результат лучший, чем методы, основанные на раздельном анализе активностей.

5.3.8.2 Метод принятия решения

Метод был представлен в работе [57]. Он основан на динамическом изменении уровня доверия к пользователю, что позволяет быстро реагировать на попытки несанкционированного доступа. В ходе работы пользователь характеризуется уровнем доверия C. В начале сессии уровень равен 0. После каждого нажатия он изменяется в зависимости от информации, содержащейся в шаблоне пользователя. Если ритм нажатий или время нажатия-отпускания клавиш соответствуют шаблону пользователя, значение уровня доверия уменьшается, в противном случае — увеличивается.

Пока уровень не превышает заданного порога, пользователь считается корректным. Если значение C превышает этот порог, необходимо проинформировать систему безопасности и, возможно, предпринять дополнительные действия для идентификации пользователя. Значение C никогда не должно быть меньше 0. Таким образом, функция расчета значения C может быть представлена следующей формулой:

$$C_k = \begin{cases} 0, k = 0, \textit{начало сессии} \\ \max(C_{k-1} - R, 0), d \le T \\ C_{k-1} + d - T, d > T \end{cases} \quad (36)$$

Здесь R — величина поощрения пользователя за нажатие, соответствующее шаблону, d — расстояние от очередного нажатия до шаблона пользователя, T — доверительный порог расстояния. Необходимо определить максимальное значение порога C, при превышении которого пользователь признается нарушителем. Это значение может быть уникальным для каждого пользователя. Оно определяется в ходе анализа данных пользователя (тренировочных данных) и минимизирует ошибку первого рода.

Доверительный порог T выбирался равным порогу в точке, где ошибка первого рода, равна ошибке второго рода. Поощрение R пользователя за нажатие, соответствующее шаблону, бралось равным 0.75 от доверительного порога. Увеличивая или уменьшая эту

величину, можно настраивать систему на более мягкий или более жесткий режим контроля за аномалиями в наборе.

5.3.8.3 Результаты экспериментов

Для эксперимента был выбран набор данных, на котором классификаторы показывали результаты в 87% и 86% AUC для мыши и клавиатуры соответственно, это одни из наилучших полученных показателей. На рисунках 50, 51 изображены ROC-кривая и значения уровня доверия для динамики работы с мышью, на рисунках 52 и 53 — для клавиатуры.

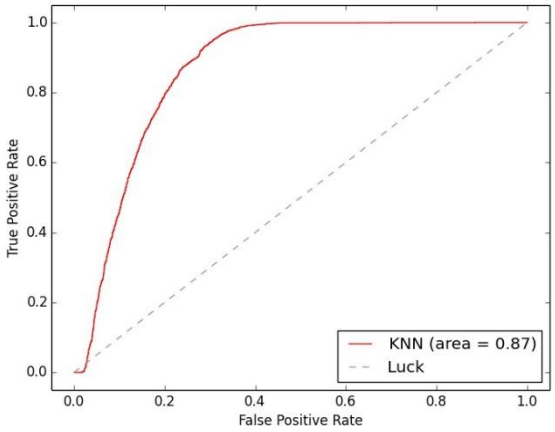


Рисунок 50 — ROC-кривая для представления динамики работы пользователя с мышью.

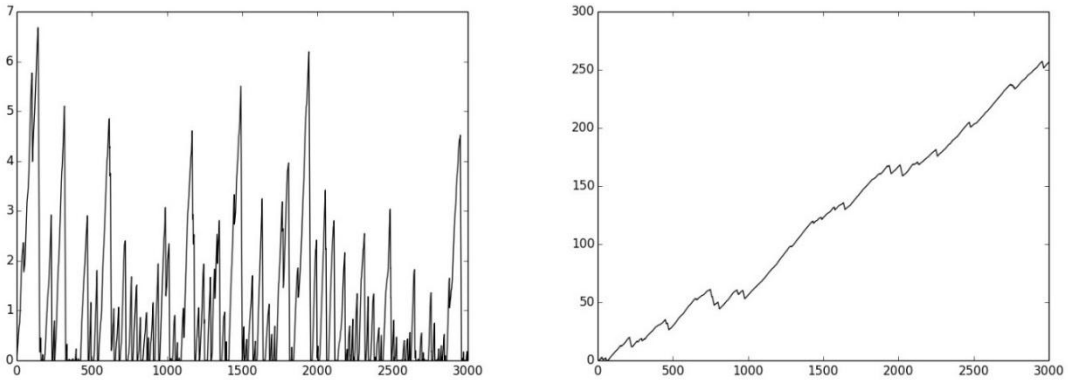


Рисунок 51 — Динамика работы пользователя с мышью. Уровень доверия для легитимного и не легитимного пользователей.

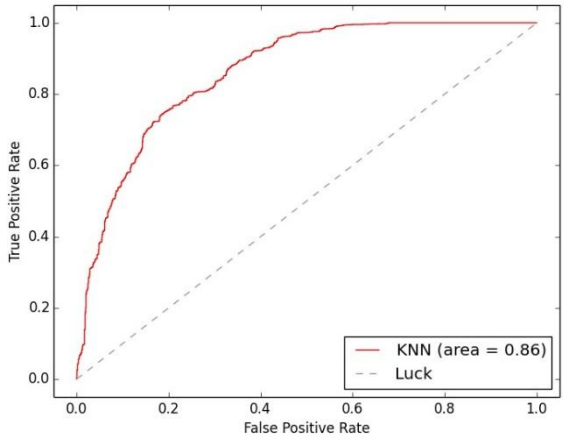


Рисунок 52 — ROC-кривая для представления динамики работы пользователя с клавиатурой.

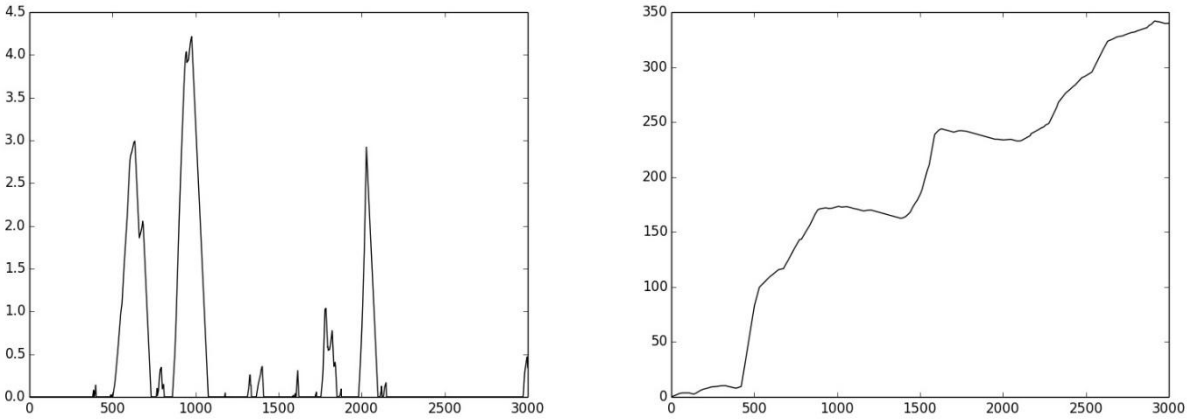


Рисунок 53 — Динамика работы пользователя с клавиатурой. Уровень доверия для легитимного и не легитимного пользователей.

Для оценки качества объединения модели использовался метод усреднения результатов классификации: каждая оценка классификатора мышки суммируется с результатом оценки ближайшего по времени вектора клавиатуры и делится на два. На рисунках 54 и 55 представлены результаты комбинированного подхода.

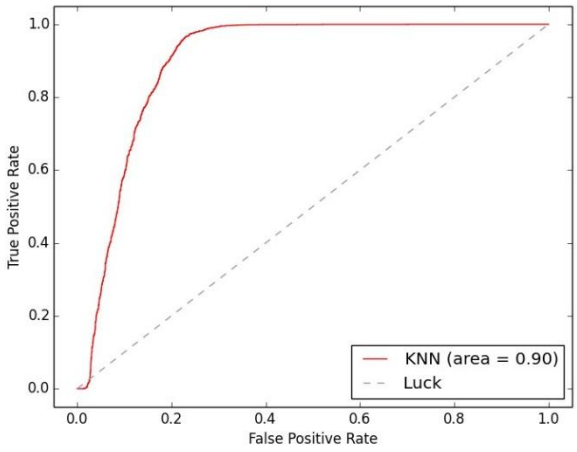


Рисунок 54 — Объединенная ROC-кривая.

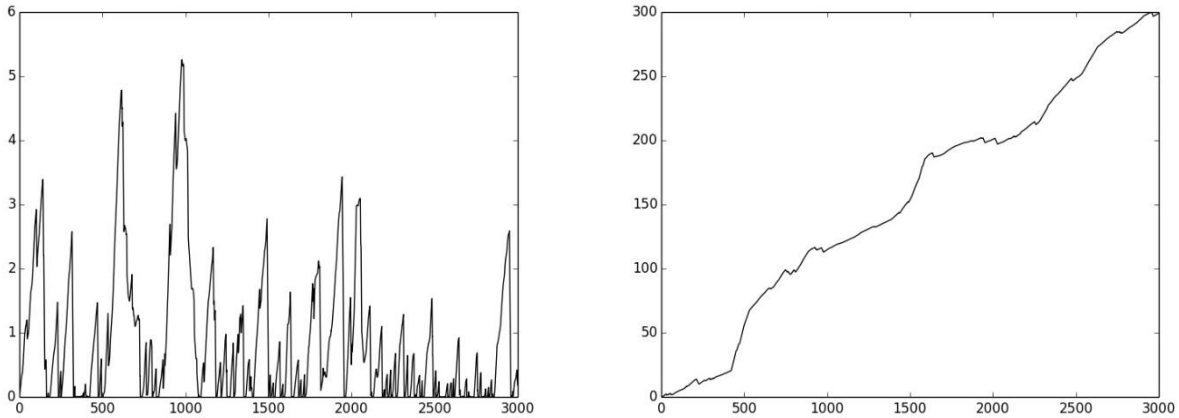


Рисунок 55 — Динамика работы пользователя с клавиатурой и мышкой. Уровень доверия для легитимного и не легитимного пользователей.

Как показали проведенные исследования, совместное использование анализа динамики работы пользователя с мышкой и клавиатурой улучшает качество классификации, сглаживая оценку локальных аномалий в работе пользователя.

5.4 Предлагаемое решение

Для решения задачи статической аутентификации на основе динамики работы пользователя с клавиатурой предлагается применять метод представления данных, основанный на фиксации времен удержания нажатой клавиши и временных интервалов между нажатиями и отпусканиями двух последовательных клавиш, а также нового метода

классификации, основанного на нечетких множествах. Лучшие результаты были получены при использовании методов предобработки данных, в качестве ядровой функции использовалось ядро Гаусса с параметром σ = 10, параметры метода основанного на нечетких множествах: m = 1, η подбирается итеративным методом при ожидаемой доле выбросов 0.1. Применение описанного метода классификации для задачи статической идентификации является новым и ранее в литературе не представлено.

Для решения задачи статической аутентификации на основе динамики работы пользователя с мышью предлагается применять способ аутентификации на основе сравнения графических подписей (состоящих из трех букв) пользователей при помощи сопоставления траекторий с применением метода динамической трансформации временной шкалы (DTW) и метода опорных векторов SVM ($\sigma = 0.5,\ nu = 0.91$) для классификации.

Для решения задачи фоновой идентификации на основе динамики работы пользователя с клавиатурой предлагается применять метод представления данных, основанный на часто встречаемых N-граммах с окном действий равным 75, формируемым по общему набору данных со сдвигом в 10 действий, в качестве признаков выступают среднее и среднеквадратичное отклонение времени наборов N-грамм. Частоту выбора N-грамм можно варьировать от 0.03 до 0.12%. В качестве метода классификации лучшую точность показал модифицированный kNN (в качестве параметра K выбирается значение, равное корню квадратному из размера обучающей выборки), метод SVM показал близкую точность.

Для задачи фоновой идентификации на основе динамики работы с мышью предлагается в процессе обычной работы пользователя за компьютером собирать системные события динамики движений мыши, делать их предобработку путем разбиения на перекрывающиеся временные окна с учетом пауз в действиях с мышью, и далее строить модель представления на основе комбинации базовых статистических характеристик перемещений пользователем мыши и распределении шаблонов поведения пользователя. В качестве метода классификации предлагается использовать нечеткий метод поиска исключений (с параметрами $kernel = 'poly', d = 4,\ \eta = 0.82, \mathrm{m} = 1.6$) или модифицированный метод ближайших соседей (с параметром k, равным корню квадратному из размера обучающей выборки), показавшие близкие результаты по точности и достаточно высокую скорость работы. Применение этих методов для задачи фоновой идентификации является новым и ранее в литературе не представлено.

5.5 Выводы

В данном разделе представлены результаты разработки методов машинного обучения и математической статистики для построения и применения поведенческих моделей на основе биометрической информации об особенностях работы со стандартными устройствами ввода-вывода (клавиатура, мышь, монитор).

В ходе решения поставленных задач были разработаны следующие методы:

- метод статической аутентификации пользователя по динамике его работы с клавиатурой на основе классификатора, использующего нечеткие множества, показавшего одну из самых высоких точностей среди существующих классификаторов, при использовании представления данных, основанного на фиксации времен удержания нажатой клавиши и временных интервалов между нажатиями и отпусканиями последовательных клавиш;
- метод статической аутентификации пользователя по динамике его работы с мышью на основе классификатора SVM и метода динамической трансформации временной шкалы (DTW) для сравнения графических подписей, состоящих из трех символов;
- метод фоновой идентификации пользователя по динамике его работы с клавиатурой на основе классификаторов модифицированный kNN или SVM, показавших близкую точность, и нового метода представления, основанного на использовании часто встречаемых N-грамм;
- метод фоновой идентификации пользователя по динамике его работы с мышью на основе нечеткого метода поиска исключений или модифицированного метода ближайших соседей, показавших близкую точность, и представления данных, основанного на комбинации базовых статистических характеристик перемещений пользователем мыши и распределении шаблонов поведения пользователя

Проведённые экспериментальные исследования предложенных методов представления и алгоритмов классификации для решения задач статической аутентификации и фоновой идентификации на собранных наборах экспериментальных данных подтвердили полученные выводы.

Результаты, полученные в рамках настоящего раздела, подтверждают соответствие требованиям пп. 2.1.2, 3.5, 4.1.1.2 подпункт 1) ТЗ.

6Формирование наборов экспериментальных данных

Настоящий раздел посвящен описанию работ и полученных результатов по формированию наборов экспериментальных данных.

Согласно п.4.1.2.12 ТЗ формируемые наборы экспериментальных данных должны быть предназначены для проведения экспериментальных исследований ЭО ПК, разрабатываемого не следующем этапе настоящих ПНИ, для демонстрации соответствия результатов теоретических исследований требованиям технического задания.

Согласно п. 4.1.2.10 ТЗ в качестве исходных поведенческих биометрических данных для ЭО ПК должны быть использованы данные, собранные в рамках работы пользователя со стандартным человеко-машинным интерфейсом (клавиатура, мышь, монитор) без использования дополнительного сканирующего и записывающего оборудования (камеры, сканеры, микрофоны и т.д.), а именно:

1. данные о динамике работы с устройствами ввода-вывода;
2. системные и прикладные журналы работы с информационными и вычислительными ресурсами;
3. динамика обработки и содержание создаваемой и потребляемой пользователем текстовой информации.

Для проведения экспериментальных исследований с ЭО ПК в 4ом этапе настоящих ПНИ формируемые в рамках настоящих работ наборы экспериментальных данных также должны удовлетворять требованиям п.4.1.2.10 ТЗ.

С другой стороны, согласно п.4.1.1.2 ТЗ формируемые наборы экспериментальных данных должны быть предназначены для экспериментальных исследований в рамках решения следующих задач:

- задач аутентификации без использования секретной информации и постоянной фоновой идентификации пользователей на основе поведенческой биометрической информации об особенностях работы со стандартными устройствами ввода-вывода;
- задач постоянной фоновой идентификации пользователей и раннего обнаружения внутренних вторжений на основе поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы;

– задач постоянной фоновой идентификации пользователей и раннего обнаружения попыток хищения конфиденциальной информации на основе поведенческой биометрии работы пользователя с текстовыми данными.

Отталкиваясь от решаемых в настоящих ПНИ перечисленных выше задач и требований к исходным поведенческим биометрическим данным для ЭО ПК проводились следующие работы:

– Формирование наборов экспериментальных данных для проведения экспериментальных исследований в рамках решения задач аутентификации без использования секретной информации и постоянной фоновой идентификации пользователей на основе поведенческой биометрической информации об особенностях работы со стандартными устройствами ввода-вывода.
– Формирование наборов экспериментальных данных для проведения экспериментальных исследований в рамках решения задач постоянной фоновой идентификации пользователей и раннего обнаружения внутренних вторжений на основе поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы.
– Формирование наборов экспериментальных данных для проведения экспериментальных исследований в рамках решения задач постоянной фоновой идентификации пользователей и раннего обнаружения попыток хищения конфиденциальной информации на основе поведенческой биометрии работы пользователя с текстовыми данными.

6.1 Формирование наборов экспериментальных данных для проведения экспериментальных исследований в рамках решения задач аутентификации без использования секретной информации и постоянной фоновой идентификации пользователей на основе поведенческой биометрической информации об особенностях работы со стандартными устройствами ввода-вывода

В рамках проведенных работ требовалось сформировать наборы данных, содержащие информацию о динамике работы пользователя с клавиатурой и мышью, для решения задач аутентификации без использования секретной информации и постоянной фоновой идентификации пользователей на основе поведенческой биометрической информации об

особенностях работы со стандартными устройствами ввода-вывода, в соответствии с пунктом 4.1.1.2 ТЗ.

6.1.1 Требования к формируемым наборам данных

Сформированные наборы данных должны были позволять провести эксперименты по каждой из перечисленных ниже задач:

- задача статической аутентификации пользователя на основе информации об особенностях его работы с мышью;
- задача статической аутентификации пользователя на основе информации об особенностях его работы с клавиатурой;
- задача фоновой идентификации пользователя на основе информации об особенностях его работы с мышью;
- задача фоновой идентификации пользователя на основе информации об особенностях его работы с клавиатурой.

Для задач статической аутентификации было необходимо провести сбор информации по следующим сценариям:

1. ввод пароля на физической клавиатуре: фиксируется динамика работы с клавиатурой;
2. ввод изображения (подписи) в область на экране: фиксируется динамика работы с мышкой.

Для задач фоновой аутентификации было необходимо провести сбор информации с одновременной фиксацией динамики работы пользователя с клавиатурой и мышью.

По сформулированным критериям для решения задачи статической аутентификации на основе динамики работы пользователя с клавиатурой был выбран набор данных с условным обозначением «НЭДК», представленный в статье [53], более подробно этот набор описан ниже в пункте 6.1.4.

Принимая во внимание то, что для решения оставшихся трех задач не было найдено подходящих наборов данных, было принято решение о формировании дополнительных наборов данных по каждой из указанных задач. Сведения о сформированных наборах для статической аутентификации приводятся ниже в пункте 6.1.5, сведения о наборах, сформированных для фоновой идентификации приводятся ниже в пункте 6.1.6.

6.1.2 Формат сбора и хранения данных

Формирующиеся наборы данных должны были быть представлены в виде текстовых файлов, отдельно для динамики работы с клавиатурой, отдельно для динамики работы с мышью. Каждая строчка в файле соответствует одному фиксируемому событию. Перечень фиксируемых событий и формат их записи должен был соответствовать формату, выбранному в пункте 4.2.1 отчета.

Для клавиатуры фиксируется следующая информация о событии:

– *key* — код клавиши,
– *action* — тип события:
  - 1 – клавиша нажата,
  - 2 – клавиша отпущена,
– *time* — время, когда произошло событие,
– *modifiers* — признак удержания клавиши-модификатора при наборе,
– *process* — имя процесса в котором произошло событие.

Данные о событии формировались в строку в следующем порядке:

`key,action,time,modifiers,process`

Для мышки фиксировалась следующая информация о событии:

– *button* — отображает, нажата ли кнопка мыши:
  - 0 — никакая кнопка не используется,
  - 1 — левая кнопка,
  - 2 — правая кнопка,
  - 3 — обе,
– *buttons* — отображает, какая кнопка мыши была нажата в предыдущий момент времени:
  - 0 — никакая кнопка не была нажата,
  - 1 — левая кнопка,
  - 2 — правая кнопка,
  - 3 — обе,
– *x* — горизонтальная координата мыши на экране,
– *y* — вертикальная координата мыши на экране,
– *wheel* — отображает, использовалось ли колесо прокрутки:
  - 0 — не использовалось,
  - 1 — использовалось,

– *action* — тип действия:
  - 1 — зажата кнопка,
  - 2 — кнопку отпустили,
  - 4 — перемещение (кнопки не использованы),
– *time* — время, когда произошло событие,
– *process* — название используемого приложения (в виде пути к исполняемому файлу).

Данные о событии формировались в строку в следующем порядке:

```
button,buttons,x,y,wheel,action,time,process
```

6.1.3 Используемый инструментарий

Согласно пункту ТЗ 4.1.2.1 процедура сбора данных должна была обеспечивать:

– обработку события нажатия любых кнопок клавиатуры, включая служебные, с фиксацией времени нажатия и времени удержания кнопок;
– фиксацию траектории движения мыши в рамках выбранных приложений;
– функционирование процедуры сбора в фоновом режиме.

Для обеспечения этих требований при реализации приложений по сбору данных необходимо было использовать библиотеку для перехвата событий операционной системы, приходящих от клавиатуры и мыши, подробно описанную в пункте 4.2.2 отчета.

6.1.4 Набор экспериментальных данных НЭДК

Выбранный набор данных был сформирован авторами статьи [53] и выложен в свободный доступ. Достоинствами этого набора, на основе которых было принято решение о его использовании являются:

– репрезентативность: в сборе данных участвовал 51 человек;
– соответствие решаемой задаче: данные представляют собой информацию о нажатиях клавиш во время набора сгенерированного случайным образом десятисимвольного пароля;
– значительное количество повторений: каждый из испытуемых набирал пароль 400 раз, что оказывается достаточным как для обучения классификатора, так и для его апробации;
– информация о существующих результатах на этих данных: в своей работе авторы привели описание нескольких экспериментов с использованием этого набора данных и их результаты.

Данные в наборе представлены файлом в формате Comma Separated Value, что позволяет легко обработать их без использования дополнительных программных средств. Каждая строка файла, начиная со второй, описывает одну попытку набора пароля одним из испытуемых и содержит описание следующих временных промежутков: времени удержания клавиши, временного промежутка между двумя нажатиями, временного промежутка между нажатием каждой клавиши и отпусканием предыдущей (последние два значения отсутствуют для первой клавиши).

6.1.5 Набор экспериментальных данных для статической аутентификации

Как уже было сказано выше, в рамках работ по формированию наборов данных для решения задач статической аутентификации необходимо было осуществить сбор данных по следующим сценариям:

1. ввод пароля на физической клавиатуре: фиксируется динамика работы с клавиатурой;
2. ввод изображения (подписи) в область на экране: фиксируется динамика работы с мышкой;

Для сбора данных по этим сценариям было разработано специальное приложение.

6.1.5.1 Описание приложения для сбора данных для статической аутентификации

Общий принцип работы приложения состоит в сборе активности мышки и/или клавиатуры при попытке аутентификации пользователя, данные активности сохраняются в текстовые файлы: одна попытка, один файл с записью зафиксированных событий. Таким образом, набор файлов образует «модель» работы пользователя при выбранном способе аутентификации. На основе этих данных можно обучить классификатор и провести тестовую аутентификацию с отображением полученной меры сходства.

После запуска приложения (см. рисунок 56) пользователь может создать новую модель данных с помощью кнопки «Добавить модель», дообучить существующую модель, выбрав её и нажав кнопку «Обучить», либо провести пробный ввод для уже обученной модели с использованием подключенного метода аутентификации, также может удалить модель нажав кнопку «Удалить модель».

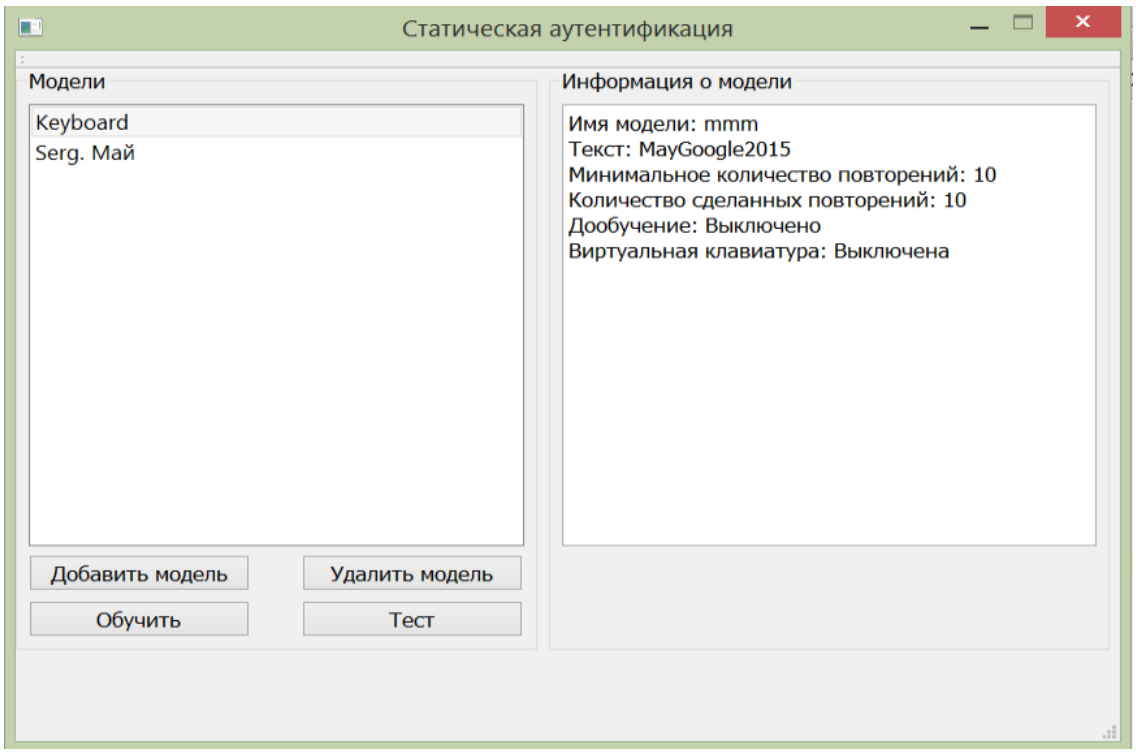


Рисунок 56 — Приложение для сбора данных при статической аутентификации.

Если пользователь нажимает кнопку «Добавить модель», появляется окно, изображенное на рисунке 57. В нем можно задать имя модели, текст, который нужно будет вводить пользователю, минимальное количество обучающих вводов, после которого будет возможно тестирование модели, дообучать или нет модель в случае успешного ввода при тестировании модели, включать или нет режим графической подписи (рисунок 58).

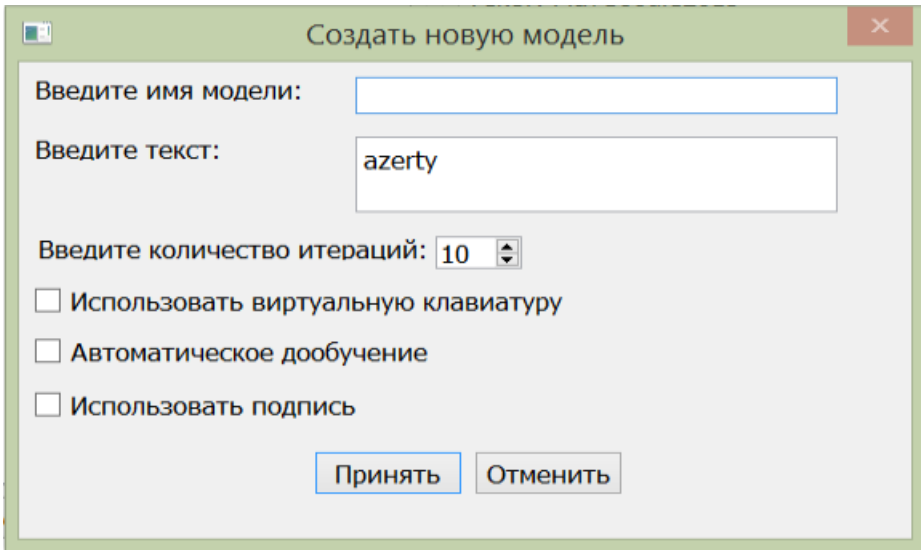


Рисунок 57 — Создание новой модели.

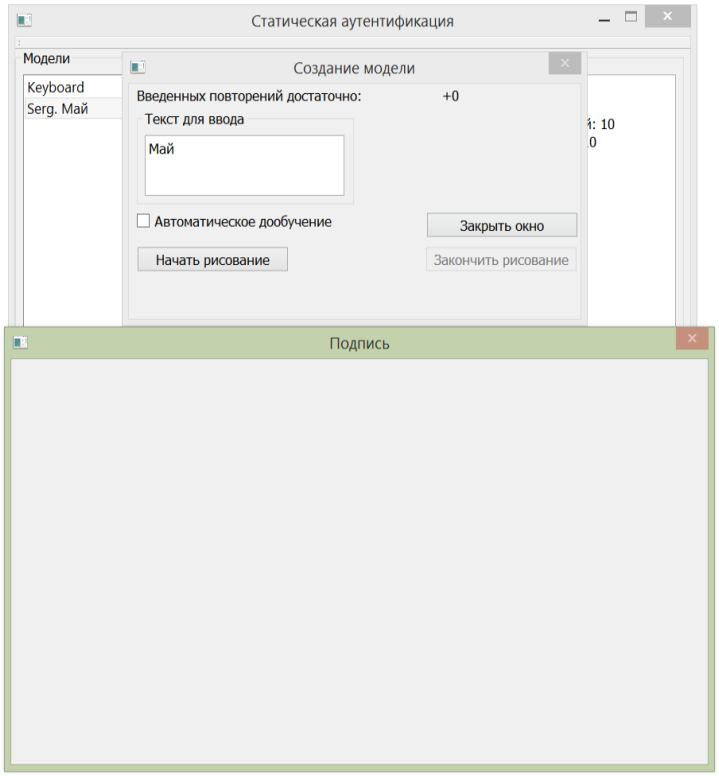


Рисунок 58 — Графическая подпись.

В ходе ввода обучающего набора или теста готовой модели, приложение определяет начало и конец сессии и фиксирует активность пользователя в этот период времени, активность пользователя вне периода ввода игнорируется.

По умолчанию используется режим стандартного клавиатурного ввода последовательности символов, что соответствует первому сценарию сбора данных.

При включении режима использования подписи пользователь должен нарисовать предопределенное слово (подпись) мышкой в области приложения, что соответствует второму сценарию сбора данных.

Если выбрать опцию «Тест», то пользователю отображается окно ввода и, после попытки аутентификации, отображается мера сходства наборов, специфичная для подключенного алгоритма аутентификации (см. рисунок 59).

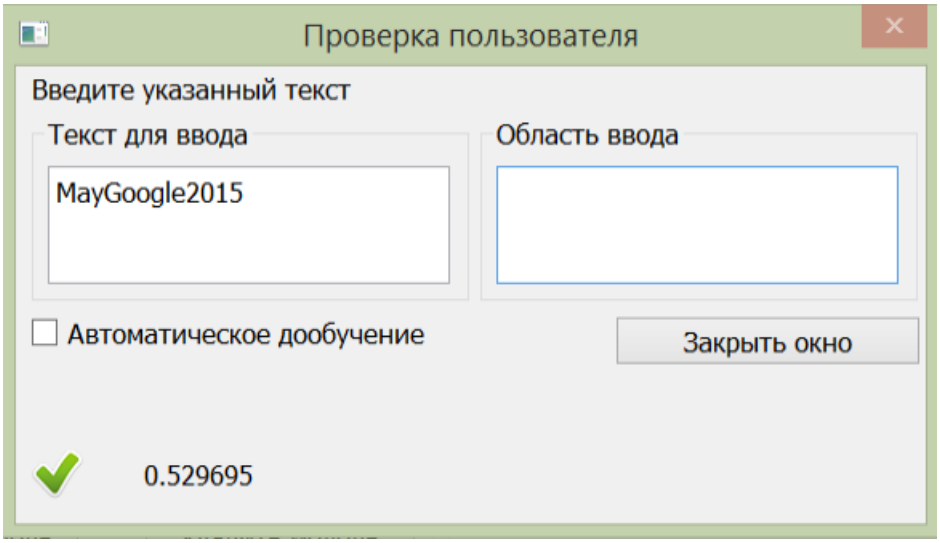


Рисунок 59 — Проверка пользователя по клавиатурному почерку.

6.1.5.2 Данные, собранные при выполнении сценариев по статической аутентификации на основе работы пользователя с клавиатурой

Сбор данных проходил на протяжении нескольких месяцев на одной машине, в сборе участвовало 10 человек. Каждый пользователь периодически делал от 10 до 20 повторений ввода различных последовательностей символов, содержащих цифры и буквы в различных регистрах. В итоге каждым пользователем было произведено по 100 вводов восьми различных последовательностей. Назовем этот набор НЭДК2. Ниже перечислены использованные последовательности символов:

– MayGoogle2015
– microsoft
– google
– azerty
– O7edzf
– ZPk9qv
– LcpOhBi3
– ozwifq0a
– ru5nsonmnx
– Imx4yyOfhq

6.1.5.3 Данные, собранные при выполнении сценариев по статической аутентификации на основе работы пользователя с мышью

Сбор данных для статической аутентификации на основе графической подписи осуществлялся на одной машине, в сборе участвовало 10 человек. Образцы для графической подписи включали в себя слова, длиной от одной до шести букв (см. рисунок 60). При вводах пользователя приложением для сбора данных фиксируются системные события динамики действий пользователя с мышкой: нажатия и отпускания кнопок мышки, перемещения мышки. Каждое событие мыши характеризуется его типом, координатами курсора и временем, когда это событие произошло. Каждый пользователь делал 20 повторений ввода каждого из различных слов.

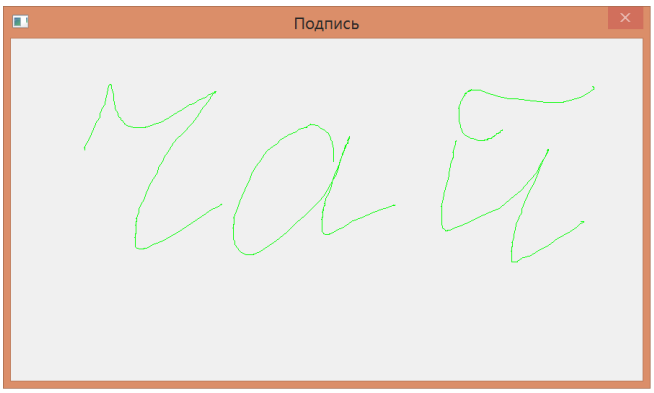


Рисунок 60 — Ввод подписи.

6.1.6 Набор экспериментальных данных для фоновой идентификации

Приложение для сбора данных, используемых в экспериментах по фоновой идентификации, должно было быть незаметным для пользователя, при этом позволяя останавливать сбор в любой момент времени. При запуске приложения появляется окно, изображенное на рисунке 61, где пользователю предлагается начать или остановить сбор данных для клавиатуры и мышки, нажав соответствующую кнопку. При нажатии кнопки минимизации, приложение сворачивается в трей, откуда может быть восстановлено по двойному щелчку мышкой на иконку приложения.

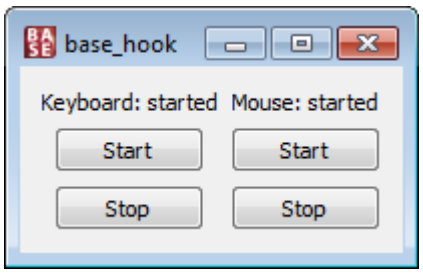


Рисунок 61 — Приложение для фонового сбора динамики работы пользователя с клавиатурой и мышкой.

Сбор данных проходил на протяжении нескольких недель у десяти пользователей. Сбор продолжался до тех пор, пока пользователь не набирал минимум 125.000 нажатий, что соответствует, в среднем, пяти дням активной работы. Параллельно собирались данные о динамике работы пользователя с мышкой, среднее соотношение зафиксированных событий клавиатуры к событиям мышки 1 к 35. Собранные наборы данных для клавиатуры назовем НЭДК3, для мышки НЭДМ2.

В итоге было собрано около 700 тысяч нажатий клавиш и в 35 раз больше событий от мышки. Общий объем собранных данных превысил 3.7 гигабайта.

6.1.7 Выводы

В ходе проведенных работ были реализованы три программных средства:

- приложение для сбора данных для статической аутентификации по первому сценарию: реализовано на языке C++ с использованием фреймворка Qt 5.4.1, объем кода порядка 2500 строк;
- приложение для сбора данных для статической аутентификации по второму сценарию: реализовано на языке Python с использованием библиотеки tkinter, объем кода порядка 900 строк;
- приложение для сбора данных для фоновой идентификации: реализовано на языке C++ с использованием фреймворка Qt 5.4.1, объем кода порядка 800 строк.

С помощью реализованных программных средств были сформированы наборы данных, предназначенные для решения задач статической аутентификации и фоновой идентификации пользователя на основе информации об особенностях его работы с клавиатурой и мышью. Данные собирались при вводе пароля на клавиатуре и вводе изображения (подписи) в области на экране. Собранные данные удовлетворяют требованиям,

указанным в пункте 4.1.2.1 ТЗ, и сохранены в формате, выбранном в пункте 4.2.1 настоящего отчета.

6.2 Формирование наборов экспериментальных данных для проведения экспериментальных исследований в рамках решения задач постоянной фоновой идентификации пользователей и раннего обнаружения внутренних вторжений на основе поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы

Для решения задач постоянной фоновой идентификации пользователей и раннего обнаружения внутренних вторжений на основе поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы требуются данные из системных и прикладных журналов работы с информационными и вычислительными ресурсами.

Исходя из формулировок решаемых задач были предъявлены следующие критерии к формируемому набору экспериментальных данных:

- данные о работе пользователей в рамках корпоративной среды;
- возможность сопоставления записей журналов с пользователями;
- возможность определения времени каждой записи в журналах.

При этом для верификации решения задачи раннего обнаружения внутренних вторжений дополнительно предъявляется следующий критерий:

- возможность выделения записей журналов, свидетельствующих о вероятном вторжении.

По сформулированным критериям для формирования набора экспериментальных данных в рамках решения задачи раннего обнаружения внутренних вторжений за основу был выбран набор DARPA 1999 [37], находящийся в свободном доступе. Более детальная информация об исходном наборе DARPA 1999, а также о полученном на его основе наборе экспериментальных данных с условным обозначением «НЭД1» приводится ниже в пункте 6.2.1.

Однако набор DARPA 1999 не содержит информации о работе пользователей с такими информационными и вычислительными ресурсами компьютерной системы, как сведения о работе с файловой системой, работой по сети, а также информации о работе с устройствами ввода — клавиатурой и мышью. Более того, указанный основополагающий набор ограничен по числу машин, с которых собирались данные, и числу пользователей, работавших на них.

Поэтому было принято решение о формировании дополнительного набора данных, удовлетворяющих перечисленным выше критериям, для обеспечения проведения экспериментальных исследований решения задачи постоянной фоновой идентификации пользователей. Сведения о сформированном наборе с условным обозначением «НЭД2» приводятся ниже в пункте 6.2.2.

6.2.1 Набор экспериментальных данных НЭД1

6.2.1.1 Основополагающий набор данных DARPA 1999

За основу набора НЭД1 был выбран набор данных DARPA 1999 [37]. Указанный набор был получен в рамках исследований «DARPA Intrusion Detection Evaluation», проводимых в 1998-2000 гг. MIT Lincoln Lab [58] совместно с Defense Advanced Research Projects Agency (DARPA ITO) [59] и Air Force Research Laboratory (AFRL/SNHS) с целью разработки методики оценки систем обнаружения вторжений. Методика подразумевает использование набора данных, описывающего модель вторжений в локальную сеть. Для получения этих данных была воспроизведена типичная локальная сеть, используемая в военно-воздушных силах США, кроме того, эта сеть была подвергнута множеству различных атак.

Набор данных DARPA 1999 состоит из двух частей: из обучающего набора и тестового. Каждый набор содержит в т.ч. файлы журналов аудита Windows NT для машины hume (см. рисунок 62) — журналы приложений (Application Event Log), безопасности (Security Event Log) и системы (System Event Log). Каждый файл журнала содержит события, зафиксированные за один день работы указанной машины. Тренировочный набор содержит сведения за три последовательные недели, а тестовый — за следующие две (т.е. данные 4-ой и 5-ой недель).

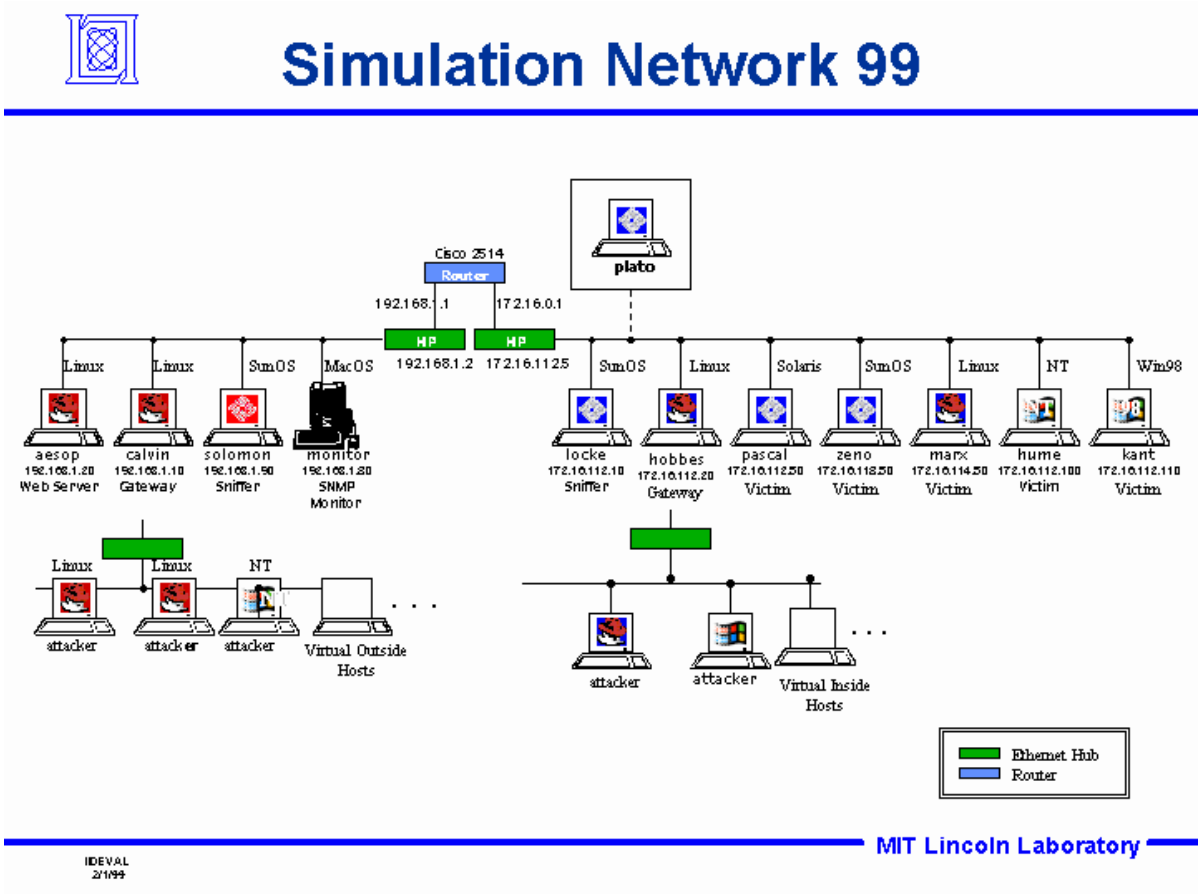


Рисунок 62 — Модель корпоративной сети в рамках исследований DARPA Intrusion Detection Evaluation.

Собранные в течение первой и третьей недели тренировочные данные не содержат никаких атак, в то время как в течение второй недели некоторое подмножество атак присутствует — это сделано с целью иллюстрации того, как можно сообщать о выявленных случаях атак [37].

Данные, собранные в течение последующих 4-ой и 5-ой недель, содержат сведения относительно того, как на фоне рутинной нормальной работы происходили атаки: т.е. данные из журналов являются результатом воздействия на систему со стороны атакующего. Используя выходные данные и сведения о проведенных атаках, возможно проводить оценку средств выявления аномалий: рассчитывать вероятность обнаружения атак и вероятность ложного срабатывания средств обнаружения вторжений на данном тестовом наборе. При этом определяется и список атак, которые выявляются средствами, и атаки, которые не выявляются.

Исходный набор DARPA 1999 ограничен в плане представленности типов событий, описывающих работу пользователей с вычислительными и информационными ресурсами компьютерной системы. Наибольшую ценность с точки зрения выявления вторжений

представляют записи журнала безопасности, несущие сведения о запуске процессов. Распределение числа зафиксированных событий в журналах аудита в течение каждой недели представлено в таблице 25.

Таблица 25 — Распределение числа событий в журналах аудита набора DARPA 1999.

Неделя	День	Журнал приложений	Журнал безопасности	Журнал системы
1	1	114110	405646	210
	2	20063	Не представлен	15
	3	10508	16041	25
	4	66448	159505	31
	5	51501	1093373	9
2	1	57563	1127617	8
	2	57563	1127617	8
	3	57563	1127617	8
	4	20882	243257	20
	5	15428	23937	13
3	1	54965	990213	Не доступен
	2	38328	1028064	8
	3	9455	8421	8
	4	36652	1540598	Не доступен
	5	9709	7495	8
	6	38620	7287	Не доступен
	7	24613	9313	Не доступен
	8	75220	22026	14
4	1	25668	32095	14
	2	30447	27122	7
	3	36874	954905	8
	4	64075	1268736	9
	5	53524	10068	2
5	1	56210	1148739	17
	2	38102	10730	15
	3	52452	1202387	1
	4	66697	42670	11

	5	70872	71982	35

В ходе анализа представленных в наборе данных было выявлены следующие недочеты и особенности. Во второй обучающей неделе данные за первые три дня абсолютно идентичны (т.е. содержат копии одних и тех же файлов журналов). Также авторы набора сами выявляли недочеты в ходе накопления данных, поэтому в набор были включены журналы для дополнительных дней (см. журналы для недели 3 дни 6–9). Помимо этого, в тренировочных неделях 4 и 5 происходили сбои, в итоге в набор были включены данные, собранные за другие даты. Авторы набора рекомендовали фильтровать события, входящие в заданные временные интервалы, определенные для каждой недели. Таким образом, за основу набора НЭД1 брались события для временных интервалов, представленных в таблице 26.

Таблица 26 — Характеристики временных интервалов сбора событий набора DARPA 1999.

	Неделя	Начало	Окончание
Обучающий набор	1	01.03.1999 00:00:00	07.03.1999 23:59:59
	2	08.03.1999 00:00:00	14.03.1999 23:59:59
	3	15.03.1999 00:00:00	28.03.1999 23:59:59
Тестовый набор	4	29.03.1999 00:00:00	04.04.1999 23:59:59
	5	05.04.1999 00:00:00	11.04.1999 23:59:59

6.2.1.2 Вторжения. Программное средство маркировки вторжений

Исходный набор данных DARPA 1999 содержит записи журналов, собранных в ходе воздействия на целевые машины различными вторжениями. Для разметки исходного набора на предмет осуществления атаки к набору DARPA 1999 прилагается скрипт на ЯП Perl, позволяющий на основе предопределенных сигнатур выявлять следующие виды атак (описания приводятся согласно [40]):

- netbus — атака заключается в установке на машине жертвы NetBus-сервера. В последствии атакующий может подключаться к серверу удаленно и выполнять практически любые действия от имени работающего в данный момент пользователя.
- yaga — атака, направленная на создание нового пользователя в группе администраторов путем взлома реестра.
- NTinfoscan (или ntis) — атака, заключающаяся в NetBIOS-сканировании с целью сбора сведений информационной безопасности. На целевой NT-машине собирается вся

доступная информация, включая имена всех пользователей, работающих сервисов и пр., и сохраняется, например, в html-файл, который впоследствии скачивается злоумышленником.

– CaseSen — атака, направленная на получение администраторских прав пользователем. Атака использует чувствительность к регистру каталога объектов NT.

Помимо этого, указанный скрипт позволяет выделять события, которые косвенно могут свидетельствовать о возможном проведении атаки (например, аварийная перезагрузка компьютера может указывать о возможных вторжениях, использующих уязвимости ОС при старте компьютера).

6.2.1.3 Формирование набора экспериментальных данных НЭД1

При формировании набора НЭД1 потребовалось провести предобработку данных, представленных в исходном наборе DARPA 1999. Напомним, что набор представляет собой набор файлов аудита ОС Windows NT. Детальную информацию об организации системных журналов ОС Windows можно найти в [31].

Для возможности применения программного средства маркировки вторжений необходимо представить собранные события журналов аудита NT в виде текстового файла, в котором события следуют в хронологическом порядке. Для выгрузки данных в указанном формате было реализовано программное средство.

В дальнейшем результаты применения средства маркировки вторжений — выделенные события, отвечающие сигнатуре искомой атаки — сохраняются в экспериментальной базе данных, речь о которой пойдет ниже.

Для формирования набора НЭД1 было разработано программное средство выгрузки событий, представленных в исходном наборе DARPA 1999. Данное средство позволяет производить выгрузку данных, предоставляя возможность задавать фильтры для выгружаемых событий — указывать типы журналов, типы событий, задавать имена пользователей и/или компьютеров, типы источников и соответствующие им идентификаторы событий, задавать границы временных интервалов для времени фиксации событий. Помимо этого, разработанное программное средство для каждого события по возможности формирует текст его сообщения, используя файл с локализованными строками сообщений (параметр EventMessageFile, задаваемый для каждого журнала ОС Windows) и значения параметров целевого сообщения. Результат выгрузки программное средство может сохранять в нескольких форматах:

- В виде набора текстовых файлов — в этом случае для каждого выгружаемого события формируется текстовый файл, содержащий полученное текстовое сообщение данного события, а заголовок формируется на основе подстановки значений атрибутов текущего события в заранее заданный пользователем шаблон заголовка (например, заголовок может содержать информацию о времени фиксации события, названии журнала и т.п.). Такой подход позволяет в дальнейшем удобно оперировать множеством событий, сохраненных в файловой системе. В частности, данный подход позволяет сопоставлять каждой единице времени или временному интервалу (например, минута/час/сутки) множество событий, записанных в соответствующих файлах.
- В экспериментальной базе данных — в этом случае каждое выгружаемое событие сохраняется в таблице БД (использовалась СУБД Microsoft SQL Server 2014), имеющей структуру, приведенную на таблице 27. Сделано это с целью подготовки базиса экспериментальных исследований с использованием аналитического инструментария Microsoft SQL Server 2014 Analysis Services.

Таблица 27 — Структура таблицы выгруженных событий.

Наименование столбца	Описание
ID	Идентификатор записи в таблице, соответствующей выгруженному событию
TIMECREATED	Дата и время фиксации события
ISTRAIN	Признак, принадлежит ли событие обучающему набору
HASINTRUSION	Признак, что данное событие соответствует какой-либо атаке, выявленной с помощью программного средство маркировки вторжений
EVENTID	Значение атрибута «Идентификатора события»
LEVEL	Значение атрибута «Тип события»
LOGNAME	Указание журнала-источника события
RECORDID	Номер записи в журнале событий
PROVIDER	Источник события
USERNAME	Имя пользователя, от имени которого зафиксировано событие
DESCRIPTION	Текстовое описание события

С помощью разработанного программного средства были выгружены события из исходного набора DARPA 1999 и сохранены в обоих указанных форматах, что и являет собой содержательную часть набора НЭД1.

Таким образом, результатом выполнения работ, описанных в настоящем пункте, является сформированный набор экспериментальных данных, включающий:

- данные, извлеченные из исходного набора DARPA 1999 и сохраненные в двух представленных форматах;
- программное средство, реализующее дополнительный функционал, используемый в ходе проведения экспериментальных исследований.

6.2.2 Набор экспериментальных данных НЭД2

6.2.2.1 Организация сбора данных. Исходные данные

Как отмечалось выше, набор НЭД1 не обладает широтой представленных сведений о работе пользователей с информационными и вычислительными ресурсами защищаемой компьютерной системы, круг которых определен в ТЗ настоящих ПНИ. Поэтому было принято решение о формировании дополнительного набора экспериментальных данных НЭД2, при этом в расчет брался следующий список типов фактов работы пользователя с информационными и вычислительными ресурсами, перечисленный в п.4.1.2.3.1 ТЗ:

- факты работы с локальными файлами;
- факты работы с внешними носителями;
- факты работы с разделяемыми сетевыми ресурсами;
- факты работы с удаленными сервисами и приложениями по сетевым протоколам TCP/IP;
- факты входа и выхода в/из системы;
- факты локального и удаленного запуска и установки приложений и сервисов;
- факты изменения программной конфигурации системы;
- факты подключения дополнительного оборудования.

Для каждого факта в п.4.1.2.3.2 ТЗ дополнительно указан список обязательных атрибутов:

- информация о времени и, где необходимо, длительности действия;
- информация об имени пользователя;
- информация об имени хоста;
- информация об осуществляющем действие программном приложении;

– информация о размере передаваемой информации в случае операций с файлами или соединений с удаленными сервисами.

Для организации сбора данных было принято решение об использовании стандартных механизмов аудита ОС семейства Microsoft Windows с дополнительной реализацией экспериментальных программных средств для покрытия всего указанного множества типов фактов работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы.

В качестве основы сбора данных использовались журналы ОС Windows. Основные журналы и идентификаторы событий в Windows XP/Vista/2008/7/8 одинаковы. Для сбора событий безопасности используется журнал Security, в котором обрабатываются события, указанные в таблице 28.

Таблица 28 — Обрабатываемые идентификаторы событий журнала Security.

Идентификатор события	Описание
512	запуск операционной системы Windows
513	запуск выключения Windows
528	успешное начало сеанса работы
538	завершение сеанса работы
551	начало выполнения завершения сеанса работы
592	создание нового процесса
593	окончание работы процесса

В журнале System собираются системные события, для построения требуемых фактов потребуются события со следующими идентификаторами (см. таблицу 29):

Таблица 29 — Обрабатываемые идентификаторы событий журнала System.

Идентификатор события	Описание
6005	событие запуска Event log

В журнале Application собираются события прикладного уровня, для построения требуемых фактов потребуются события со следующими идентификаторами (см. таблицу 30):

Таблица 30 — Обрабатываемые идентификаторы событий журнала Application.

Идентификатор события	Описание
1033	установка программы
1034	удаление программы

Также дополнительно были разработаны следующие экспериментальные программные модули, позволяющие собирать данные о работе пользователя с информационными и вычислительными ресурсами компьютерной системы:

- модуль мониторинга работы с сетью;
- модуль мониторинга отсылки данных через веб-обозреватель Microsoft Internet Explorer как в качестве передаваемого файла, так и в качестве содержимого HTML-формы;
- модуль мониторинга работы с файловой системой;
- модуль мониторинга активности использования клавиатуры и манипулятора типа мышь в приложениях.

При этом для унификации процесса получения исходных данных экспериментальные программные модули сбора реализованы таким образом, что собранная дополнительная информация об активности помещается в специально созданный в операционной системе журнал Activity. Разработанные модули предназначены для установки на целевую машину в «ручном» режиме (т.е. без использования специальных средств автоматизации пакетов установки) и не подразумевают каких-либо механизмов защиты собранной информации и передачи ее на другие узлы компьютерной сети.

В дальнейшем собранные в журналах ОС Windows данные выгружаются дополнительно реализованным экспериментальным программным средством в специальном формате в файловую систему (далее — промежуточное хранилище). При выгрузке данных дополнительно происходит их обработка с целью расчета вычисляемых показателей (например, длительность работы процесса вычисляется на основе информации событий его старта и останова).

Выгруженные данные являются исходными для формирования набора экспериментальных данных НЭД2.

6.2.2.2 Исходные данные для набора НЭД2

При формировании набора данных НЭД2 в качестве исходных брались данные, полученные из двух независимых источников — таким образом, набор НЭД2 состоит из

двух частей, идентичных по структуре, но различных по содержимому. В дальнейшем обе части будем обозначать как «НЭД2.1» и «НЭД2.2».

6.2.2.3 Исходные данные НЭД2.1

В качестве исходных данных для набора НЭД2.1 использовались сведения, полученные по договоренности с одним учреждением (далее — Учреждение), при условии, что Учреждение останется неназванным, а используемые данные будут обезличены.

Корпоративная сеть Учреждения представляет собой локальную вычислительную сеть (ЛВС), которая имеет плоскую архитектуру с сегментами VLAN и использует стек протоколов TCP/IP. Скорость передачи данных 100 Мбит/сек. Доменная структура базируется на платформе MS AD Windows Server 2003. Структурированная кабельная сеть построена на многомодовых оптических линиях (62,5/125 и 50/125) и объединяет автоматизированные рабочие места (АРМ) пользователей с централизованным хранилищем данных и серверной группой. В состав ЛВС входят до 1250 АРМ.

На АРМ установлено общее программное обеспечение:

- операционные системы (далее – ОС) Windows 2000 Pro, Windows XP Pro;
- офисное программное обеспечение Microsoft Office 2000, Microsoft Office XP, Microsoft Office 2003;
- дополнительное прикладное программное обеспечение для выполнения функциональных задач.

Отличительной особенностью корпоративной сети Учреждения является то, что за каждым отдельным компьютером (АРМ) работает один пользователь.

Согласно договоренности, исходные данные были собраны на 20 АРМ Учреждения в течение 9 подряд идущих календарных дней и в последующем обезличены.

6.2.2.4 Исходные данные НЭД2.2

В качестве исходных данных для набора НЭД2.2 использовались сведения, собранные в рамках Лаборатории технологий программирования факультета вычислительной математики и кибернетики МГУ имени М.В. Ломоносова (далее — Лаборатория ТП). Локальная вычислительная сеть Лаборатории ТП использует стек протоколов TCP/IP. Скорость передачи данных 1 Гбит/сек. Доменная структура базируется на платформе MS AD Windows Server 2008 R2. В состав ЛВС входят порядка 10 физических машин и порядка 10 виртуальных.

На компьютерах установлено общее программное обеспечение:

- операционные системы (далее – ОС) Windows XP Pro, 2003, 2008, 7, 8;
- офисное программное обеспечение Microsoft Office 2007;
- дополнительное прикладное программное обеспечение для выполнения функциональных задач.

Отличительной особенностью корпоративной сети Лаборатории ТП является то, что за каждым отдельным компьютером может работать несколько пользователей.

Исходные данные были собраны на 7 компьютерах в течение двух последовательных календарных месяцев.

6.2.2.5 Формирование набора данных НЭД2

Для обработки исходных данных НЭД2.1 и НЭД2.2 было реализовано программное средство, предназначенное для выгрузки событий из промежуточного хранилища, при этом имеется возможность задавать фильтры для выгружаемых событий — указывать типы журналов, задавать имена пользователей и/или компьютеров, типы источников и соответствующие им идентификаторы событий, задавать границы временных интервалов для времени фиксации событий. Результатом выгрузки является набор текстовых файлов, сохраненных в файловой системе с разбиением по компьютерам (т.е. для каждого компьютера создается папка, в которую помещаются файлы с информацией о событии: для каждого события — отдельный файл). Заголовок генерируемого файла формируется на основе подстановки значений атрибутов текущего события в заранее заданный пользователем шаблон заголовка (например, заголовок может содержать информацию о времени фиксации события, названии журнала и т.п.). Содержимое текстового файла в зависимости от режима выгрузки является текстовым описанием события, зафиксированного в журнале ОС, либо набором значений числовых параметров, разделенных запятой (т.н. CSV-формат — Comma Separated Values, дословно «значения, разделенные запятой»). Фиксируются следующие числовые параметры:

- длительность;
- число переданных/полученных байт;
- число байт, участвующих в обмене;
- количество нажатий клавиш клавиатуры/действий с мышью.

В случае отсутствия информации в событии о каком-либо параметре, ему присваивается нулевое значение.

Реализованный подход позволяет в дальнейшем удобно оперировать множеством событий, сохраненных в файловой системе. В частности, данный подход позволяет сопоставлять каждой единице времени или временному интервалу (например, минута/час/сутки) множество событий, записанных в соответствующих файлах. Характеристики полученного набора НЭД2 приведены в таблице 31.

Таблица 31 — Характеристики набора НЭД2.

	НЭД2.1	НЭД2.2
Число компьютеров	20	7
Среднее число (несистемных) пользователей, работающих на одном компьютере	1	6
Общее число событий	~ 248 000	~ 385 000

Результатом выполнения работ, описанных в настоящем пункте, является сформированный набор экспериментальных данных, включающий:

- данные, собранные в различных условиях в рамках групп НЭД2.1 и НЭД2.2, обработанные и сохраненные в двух представленных форматах;
- программное средство, реализующее дополнительный функционал, используемый в ходе проведения экспериментальных исследований.

6.3 Формирование наборов экспериментальных данных для проведения экспериментальных исследований в рамках решения задач постоянной фоновой идентификации пользователей и раннего обнаружения попыток хищения конфиденциальной информации на основе поведенческой биометрии работы пользователя с текстовыми данными

Для решения задач постоянной фоновой идентификации пользователей и раннего обнаружения попыток хищения конфиденциальной информации на основе поведенческой биометрии работы пользователя с текстовыми данными требуются данные о динамике обработки и о содержании создаваемой и потребляемой пользователем текстовой

информации. Исходя из формулировок решаемых задач были предъявлены следующие критерии к формируемому набору экспериментальных данных:

- текстовая информация из корпоративной среды;
- возможность сопоставления текстовых данных с пользователями;
- возможность определения времени обработки текстовых данных.

Также в расчет брался следующий перечень типов и форматов документов, для которых должны обеспечиваться сбор и обработка фактов работы пользователя, указанный в п.4.1.2.5.1 ТЗ:

1. любые электронные документы в виде локальных файлов, файлов на внешних носителях и разделяемых сетевых ресурсах в защищаемой компьютерной системе, в текстовых форматах;
2. незакодированные сообщения электронной почты, получаемые и передаваемые по протоколам IMAP и HTTP (на почтовые Web-системы) с использованием одного из веб-обозревателей.

По сформулированным выше критериям для формирования набора экспериментальных данных за основу был выбран набор Enron [4]. Набор Enron содержит электронную почту 150 сотрудников (главным образом, из высшего руководства) американской энергетической компании, обанкротившейся в конце 2001 года. Кроме того, данный набор широко распространён в работах, посвящённых тематическому анализу текстовых данных [4, 14].

Однако стандартный набор Enron содержит электронные письма без вложенных файлов (англ. attachment), т.е. данные стандартного набора Enron покрывают только п.2 из требуемого перечня типов и форматов документов. В связи с этим наряду со стандартным набором писем также рассматривалась и его версия со всеми вложениями [24], которую далее будем обозначать Enron Attachment. Прикреплённые к электронным письмам документы текстовых форматов также являются текстовой информацией, обрабатываемой пользователями в защищаемой компьютерной системе, а данные о письме, такие как время отправки/получения и адресаты, служат для описания использования текстовых документов. Поэтому для удовлетворения п.1 требуемого перечня типов и форматов документов было принято решение о формировании дополнительного набора экспериментальных данных, базирующегося на наборе Enron Attachment.

На основе стандартного набора данных Enron и набора данных Enron Attachment были сформированы два набора экспериментальных данных, используемые в настоящих ПНИ:

– набор текстов электронных писем (далее — набор текстовых экспериментальных данных 1, или НТЭД1);
– набор текстовых документов, прикреплённых к письмам (далее — набор НТЭД2).

Описание процессов формирования наборов НТЭД1 и НТЭД2 приведено ниже в следующих пунктах.

6.3.1 Формирование набора НТЭД1

Архив исходного стандартного набора данных Enron [4] представлен в виде набора папок, каждая папка соответствует отдельному почтовому ящику пользователя. Электронные письма пользователей хранятся в виде отдельных файлов. Для обеспечения полноты и репрезентативности формируемого набора экспериментальных данных необходимо выбрать пользователей, чьи электронные письма войдут в набор, и интервал времени, когда велась наиболее активная работа с электронной почтой. Для этого было разработано программное средство, которое для каждого электронного письма из набора Enron сохраняло в базе данных запись, содержащую информацию о пользователе, в чьём почтовом ящике находится письмо, и о времени обработки письма. С помощью SQL-запросов к сформированной базе данных было получено, что подавляющее большинство электронных писем относится к 2000 и 2001 годам. Далее были выбраны все пользователи, у которых число писем за 2000 и 2001 годы в почтовом ящике не меньше 10000. Для большей полноты к их числу был добавлен пользователь nemec-g, т.к. в его почтовом ящике также содержится большое число писем (в количестве 9940), и данный пользователь был включен в сформированный набор НТЭД2 (см. пункт 6.3.2). В таблице 32 приведены итоговые характеристики выбранных пользователей для набора НТЭД1.

Таблица 32 — Характеристики пользователей, выбранных из стандартного набора Enron.

Номер	Имя пользователя	Число электронных писем за 2000 и 2001 годы
1	beck-s	11330
2	dasovich-j	27640
3	farmer-d	12644
4	germany-c	10177
5	jones-t	18691
6	kaminski-v	27091
7	kean-s	24154

Продолжение таблицы 32

8	mann-k	23335
9	nemec-g	9940
10	shackleton-s	15584
11	symes-k	10823
12	taylor-m	11818
Итого	**12 пользователей**	**203227 писем**

Далее необходимо было осуществить разбор файлов электронных писем, т.е. для каждого файла электронного письма требовалось выделить текстовое содержимое и время его обработки на почтовом сервере. Для этого было разработано программное средство, которое из файла электронного письма выделяло время обработки письма на почтовом сервере, тему письма и текст тела письма, при этом если тело письма было представлено в HTML-формате, то выполнялось извлечение текста из HTML содержимого.

Таким образом, для каждого из выбранных пользователей были получены следующие экспериментальные данные: текст электронных писем (т.е. тема и текст тела); время обработки электронных сообщений.

Для проведения экспериментальных исследований (см. раздел 1) при решении задач постоянной фоновой идентификации пользователей и раннего обнаружения попыток хищения конфиденциальной информации на основе поведенческой биометрии работы пользователя с текстовыми данными также требовался следующий функционал от набора экспериментальных данных:

1. Возможность группировки текстовых данных электронных писем пользователя по заданным временным интервалам.
2. Для каждого электронного письма необходимо получать: текст письма; имя пользователя, совершившего операцию с письмом; время операции с письмом.
3. Для анализа полученных в ходе экспериментальных исследований результатов требуется вычислять число общих писем пользователей за заданные интервалы времени. Например, для пользователей dasovich-j, kaminski-v, kean-s требуется найти число их общих писем из набора за первое полугодие 2001 года (результаты выполнения данного запроса представлены в таблице 33).

Для реализации перечисленного функционала было разработано программное средство, которое обрабатывало сформированные пользовательские данные, полученные из набора Enron.

Таблица 33 — Статистика распределения общих писем между пользователями из набора за первое полугодие 2001 года.

Письма пользователя	Общие письма с пользователем		
	dasovich-j	kaminski-v	kean-s
dasovich-j	11480	56	1730
kaminski-v	16	8757	36
kean-s	2628	91	10043

Результатом выполнения работ, описанных в настоящем пункте, является сформированный набор экспериментальных данных, включающий:

– данные для 12 пользователей об обрабатываемых электронных письмах;
– программное средство, реализующее дополнительный функционал, требующийся в ходе проведения экспериментальных исследований.

6.3.2 Формирование набора НТЭД2

Архив исходного набора данных Enron Attachment [24] представлен в виде набора папок, каждая папка соответствует отдельному почтовому ящику пользователя. Электронные письма пользователей хранятся в виде PST-файлов (файлы данных ПО Microsoft Outlook). Для обеспечения полноты и репрезентативности формируемого набора экспериментальных данных были выбраны все пользователи, у которых суммарный размер PST-файлов был не меньше 1Гб. Таким образом, были выбраны 15 пользователей, представленные в таблице 34.

Таблица 34 — Пользователи набора экспериментальных данных НТЭД2.

Номер	Имя пользователя	Суммарный размер PST-файлов пользователя
1	chris_germany	1.0Гб
2	daren_farmer	1.6Гб
3	darron_c_giron	1.2Гб
4	gerald_nemec	1.2Гб
5	john_lavorato	1.2Гб
6	kate_symes	1.5Гб

Продолжение таблицы 34

7	louise_kitchen	1.4Гб
8	mark_taylor	1.1Гб
9	matthew_lenhart	1.6Гб
10	phillip_m_love	1.2Гб
11	richard_sanders	2.2Гб
12	richard_shapiro	1.1Гб
13	sally_beck	2.4Гб
14	sara_shackleton	1.7Гб
15	vkaminski	3.3Гб

Далее было необходимо из имеющихся PST-файлов извлечь данные об электронных письмах и прикреплённым к ним файлах. Для этого использовалась программа `readpst` [60] со следующими входными параметрами:

```
readpst -D -S -b -j 1 <имя_pst-файла>
```

После выполнения данной команды для каждого PST-файла пользователя создавалась папка, содержащая письма и соответствующие им прикреплённые файлы.

Поскольку не все прикреплённые к письмам файлы содержат текстовую информацию, необходимо было выбрать типы (форматы) файлов и для каждого выбранного типа файла — средство извлечения текста.

Наиболее популярными форматами документов для представления текстовой информации являются DOC, RTF и PDF. В ходе дальнейшего анализа было получено, что во многих PDF-файлах из набора данных Enron Attachment текстовая информация представлена в виде графических элементов или с использованием неизвестной кодировки, что приводит к невозможности корректного извлечения текста стандартными программными средствами, например, `pdftotext` [61]. В связи с этим при дальнейшем формировании набора экспериментальных данных рассматривались только документы форматов DOC и RTF, а для извлечения текста использовалась программа `catdoc` [62].

Таким образом, для каждого из выбранных пользователей были получены следующие экспериментальные данные: текст обрабатываемых по средствам электронной почты документов форматов DOC и RTF; время обработки документа. В таблице 35 приведены характеристики полученных данных.

Таблица 35 — Характеристики данных, полученных из набора Enron Attachment.

Номер	Имя пользователя	Число текстовых документов за 2000 и 2001 года
1	chris_germany	51
2	daren_farmer	261
3	darron_c_giron	40
4	gerald_nemec	2516
5	john_lavorato	108
6	kate_symes	70
7	louise_kitchen	243
8	mark_taylor	1182
9	matthew_lenhart	6
10	phillip_m_love	31
11	richard_sanders	868
12	richard_shapiro	707
13	sally_beck	603
14	sara_shackleton	2926
15	vkaminski	616
Итого	**15 пользователей**	**10228 текстовых документов**

Для проведения экспериментальных исследований (см. раздел 1) при решении задач постоянной фоновой идентификации пользователей и раннего обнаружения попыток хищения конфиденциальной информации на основе поведенческой биометрии работы пользователя с текстовыми данными также требовался следующий функционал от набора экспериментальных данных:

1. Возможность группировки текстовых документов пользователя по заданным временным интервалам.
2. Для каждого текстового документа необходимо получать: текст документа; имя пользователя, совершившего операцию с документом; время операции с документом.
3. Для анализа полученных в ходе экспериментальных исследований результатов требуется вычислять число общих писем пользователей за заданные интервалы времени. Например, для пользователя sara_shackleton требуется найти число его

общих писем с другими пользователями из набора за 2000 и 2001 годы (результаты выполнения данного запроса представлены в таблице 36).

Для реализации перечисленного функционала было разработано программное средство, которое обрабатывало сформированные пользовательские данные, полученные из набора Enron Attachment.

Таблица 36 — Число общих писем пользователя sara_shackleton с другими пользователями из набора за 2000 и 2001 годы.

	mark_taylor	phillip_m_love	sally_beck
sara_shackleton	335	59	10

Результатом выполнения работ, описанных в настоящем пункте, является сформированный набор экспериментальных данных, включающий:

- данные для 15 пользователей о текстовых документах, обрабатываемых посредством электронной почты;
- программное средство, реализующее дополнительный функционал, требующийся в ходе проведения экспериментальных исследований.

6.4 Выводы

В рамках работ по формированию наборов экспериментальных данных был проведен анализ ТЗ, определены решаемые в рамках настоящих ПНИ задачи, сформулированы требования к исходным поведенческим биометрическим данным для ЭО ПК, и на их основе были проведены следующие работы:

- Формирование наборов экспериментальных данных для проведения экспериментальных исследований в рамках решения задач аутентификации без использования секретной информации и постоянной фоновой идентификации пользователей на основе поведенческой биометрической информации об особенностях работы со стандартными устройствами ввода-вывода.
- Формирование наборов экспериментальных данных для проведения экспериментальных исследований в рамках решения задач постоянной фоновой идентификации пользователей и раннего обнаружения внутренних вторжений на основе поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы.

– Формирование наборов экспериментальных данных для проведения экспериментальных исследований в рамках решения задач постоянной фоновой идентификации пользователей и раннего обнаружения попыток хищения конфиденциальной информации на основе поведенческой биометрии работы пользователя с текстовыми данными.

По результатам выполненных работ получены наборы экспериментальных данных в объеме, достаточном для демонстрации соответствия результатов теоретических исследований требованиям технического задания в ходе проведения экспериментальных исследований ЭО ПК.

Результаты, полученные в рамках настоящего раздела, подтверждают соответствие требованиям п. 4.1.2.12 ТЗ.

7 Обеспечение работоспособности рабочих станций, общего системного и специального программного обеспечения для проведения работ по проекту, проведение регламентных работ по обеспечению сохранности данных и программ, относящихся к проводимым работам по проекту

Регламентное обслуживание обеспечивает необходимый набор услуг для поддержания оборудования и программного обеспечения в работоспособном и актуальном состоянии в процессе развития и эксплуатации инфраструктуры, включая обеспечение сохранности программ и данных, относящихся к проводимым работам по проекту.

В рамках работ по обеспечению работоспособности серверов и рабочих станций проводятся следующие процедуры:

- Обновление программного обеспечения (установка пакетов обновления поставляемых производителями программного обеспечения такими как service-pack или другие обновления). Периодичность работ — автоматические обновления там, где это возможно, ручная проверка обновлений и систем работоспособности систем автоматических обновлений — 1 раз в месяц.
- Поддержка антивирусной защиты. Периодичность работ по проверке корректного функционирования — 1 раз в месяц. Обновление лицензий антивирусного ПО, периодичность – 1 раз в год.
- Восстановление работоспособности сервера/рабочей станции после сбоя. Включает в себя повторную инсталляцию и настройку программного обеспечения, восстановление данных.
- Профилактические работы (чистка системных блоков, блоков питания, кулеров и их замена при необходимости. Периодичность работ — каждые 3 месяца для рабочих станций, 1 раз в месяц — для серверов.
- Установка дополнительного программного обеспечения. Периодичность — по мере необходимости.

- Проверка состояния жестких дисков (согласно SMART информации) в рабочих станциях и RAID-массивах серверов. При необходимости - замена жестких с дисков с восстановлением данных/консистентности RAID-массивов. Периодичность — 1 раз в месяц.
- Диагностика прочих физических неисправностей (включая периферийные устройства и соответствующий ремонт). Периодичность — по мере необходимости.

В рамках работ по обеспечению сохранности программ и данных, относящихся к проводимым работам по проекту, используются стратегии копирования и архивирования оперативных данных проекта в составе разрабатываемых программных средств/документов и отчетных материалов/экспериментальных данных на базе следующих возможностей:

- Физический уровень — настройка критических данных, хранимых на серверах, в режиме RAID1 ("зеркалирование") — массив из двух дисков, являющихся полными копиями друг друга. Периодичность работ по проверке консистентности RAID1-массива — 1 раз в месяц.
- Архивирование и восстановление данных на уровне файловой системы серверов — реализуется инкрементным копированием по расписанию средствами Windows Server Backup, встроенными в ОС семейства Windows Server. Дополнительно, используется резервное копирование и восстановление среды системы управления проектом на базе SharePoint: используются встроенные средства Microsoft SharePoint для защиты критических объектов среды (веб-приложение, сайт, база данных контента, библиотека документов, настройки, параметры конфигурации). Также, на сервере настроено автоматическое создание резервных копий баз данных системы контроля версий с целью обеспечения сохранности данных в случае программных либо аппаратных сбоев. Периодичность работ по инкрементальному архивированию критических данных проекта — ежедневно в автоматическом режиме (на базе возможностей соответствующего ПО). Периодичность работ по сохранению состояния среды — еженедельно в автоматическом режиме.
- Восстановление работоспособности рабочих станций на уровне файловой системы — реализуется на базе интегрированного в семейство ОС Microsoft Workstation средства System Restore (точки восстановления системы). Периодичность работ по созданию точек восстановления: еженедельно в автоматическом режиме, каждый раз после установки/обновления нового ПО.

– Обеспечение бесперебойного электропитания. Для уменьшения вероятности потери/повреждения критических данных все сервера и рабочие станции подключены к источникам бесперебойного питания (ИБП) с возможностью автономной работы не менее 15 минут, необходимых для автоматического корректного выключения системы. Периодичность проверки состояния ИБП (и замены батарей в случае необходимости) — каждые 6 месяцев.

ЗАКЛЮЧЕНИЕ

В рамках работ на текущем этапе прикладных научных исследований проведены теоретические исследования 2-ой очереди и получены следующие основные результаты:

- Разработаны структуры данных, методы сбора, предобработки, хранения и управления для поведенческой биометрической информации об особенностях работы со стандартными устройствами ввода-вывода (клавиатура, мышь, монитор).
- Разработаны методы машинного обучения и математической статистики для построения и применения поведенческих моделей на основе биометрической информации об особенностях работы со стандартными устройствами ввода-вывода (клавиатура, мышь, монитор).
- Разработаны структуры данных, методы сбора, предобработки, хранения и управления для поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы.
- Разработаны методы машинного обучения и математической статистики для построения и применения поведенческих моделей на основе биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы.
- Разработаны методы машинного обучения и математической статистики для построения и применения поведенческих моделей на основе поведенческой биометрии работы пользователя с текстовыми данными.

Проведенные теоретические исследования 2-ой очереди выполнялись на основе проведённого аналитического обзора, осуществлённого выбора направления исследований и предложенных подходов, представленных в отчёте за предыдущий этап настоящих ПНИ.

Кроме того, выполнены работы, финансируемые за счет средств внебюджетных источников, по формированию наборов экспериментальных данных.

Уровень полученных результатов соответствует мировому. Поставленные на заданный отчетный период задачи выполнены полностью.

Сведения о ходе выполнения настоящих ПНИ размещены в открытом доступе на официальном сайте МГУ имени М.В.Ломоносова (система ИСТИНА — Интеллектуальная система тематического исследования научно-технической информации) по адресу: http://istina.msu.ru/projects/7964619/.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1 Машечкин И.В. и др. Исследование и разработка инновационной технологии построения программных средств обеспечения компьютерной безопасности, основанных на использовании методов машинного обучения и математической статистики для анализа данных поведенческой биометрии пользователей при работе в рамках стандартного человеко-машинного интерфейса, для решения задач активной аутентификации и идентификации пользователей, обнаружения внутренних вторжений и предотвращения попыток хищения конфиденциальной информации // Отчет о прикладных научных исследованиях (промежуточный) по теме «Выбор направления исследований. Теоретические исследования (1-ой очереди) поставленных перед ПНИ задач». — М., 2014.

2 Временной ряд (Time Series) [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2011. — Режим доступа: http://www.machinelearning.ru/wiki/index.php?title=Временной_ряд. — 28.05.2015.

3 В.Ю. Королёв, А.Ю. Корчагин, И.В. Машечкин, М.И. Петровский, Д.В. Царёв. Применение временных рядов в задаче фоновой идентификации пользователей на основе анализа их работы с текстовыми данными. // Труды Института системного программирования РАН (электронный журнал), 27(1):151–172, 2015.

4 Enron Email Dataset [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2015. — Режим доступа: http://www.cs.cmu.edu/~./enron/. — 28.05.2015.

5 И.В. Машечкин, М.И. Петровский, Д.В. Царёв. Методы вычисления релевантности фрагментов текста на основе тематических моделей в задаче автоматического аннотирования. Вычислительные методы и программирование. Том 14, 2013. 91-102.

6 I.V. Mashechkin, M.I. Petrovskiy, D.S. Popov, D.V. Tsarev. Automatic text summarization using latent semantic analysis. Programming and Computer Software, 2011, pp. 299-305.

7 D.V. Tsarev, M.I. Petrovskiy, I.V. Mashechkin. Using NMF-based text summarization to improve supervised and unsupervised classification. 11th International Conference on Hybrid Intelligent Systems (HIS), 2011. Malacca, MALAYSIA. P. 185-189.

8 D.V. Tsarev, M.I. Petrovskiy I.V. Mashechkin. Supervised and Unsupervised Text Classification via Generic Summarization. International Journal of Computer Information Systems and Industrial Management Applications. MIR Labs, Volume 5, 2013, pp. 509-515.

9 I.V. Mashechkin, M.I. Petrovskiy, D.S. Popov, D.V. Tsarev. Applying Text Mining Methods for Data Loss Prevention. Programming and Computer Software. January 2015, Volume 41, Issue 1, pp 23-30.

10 C.D. Manning, P. Raghavan, H. Schutze. Introduction to Information Retrieval. Cambridge University Press, 2008.

11 Mirzal A. Converged algorithms for orthogonal nonnegative matrix factorizations // Computing Research Repository. Vol. 1010. 2010.

12 Wei Xu, Xin Liu, Yihong Gong. Document clustering based on non-negative matrix factorization. Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, Toronto, Canada, 2003.

13 Chris Ding, Tao Li, Wei Peng, Haesun Park. Orthogonal Nonnegative Matrix Tri-Factorizations for Clustering. SIGKDD, 2006.

14 M.W. Berry, M. Browne, A.N. Langville, V.P. Pauca, R.J. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. Computational Statistics and Data Analysis, pp. 155-173, 2007.

15 J. Yoo, S. Choi. Orthogonal Nonnegative Matrix Factorization: Multiplicative Updates on Stiefel Manifolds. Intelligent Data Engineering and Automated Learning – IDEAL 2008, vol. 5326 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, pp. 140–147.

16 C. Meek, D.M. Chickering, and D. Heckerman, Autoregressive Tree Models for Time-Series Analysis [Электронный ресурс]. — Электрон. дан. — [Б. м.] : Microsoft, 2002. — Режим доступа: http://research.microsoft.com/en-us/um/people/dmax/publications/dmart-final.pdf. — 28.05.2015.

17 Технический справочник по алгоритму временных рядов (Майкрософт) [Электронный ресурс]. — Электрон. дан. — [Б. м.] : Microsoft, 2014. — Режим доступа: http://msdn.microsoft.com/ru-ru/library/bb677216.aspx. — 28.05.2015.

18 Tsarev Dmitry, Kurynin Roman, Petrovskiy Mikhail, and Mashechkin Igor. Applying non-negative matrix factorization methods to discover user’s resource access patterns for computer security tasks. In Proceedings of the 2014 International Conference on Hybrid Intelligent Systems (HIS 2014), pages 43–48. IEEE Computer Society [New York], United States, 2014.

19 T. Hastie, R. Tibshirani, G. Sherlock, M. Eisen, P. Brown, D. Botstein. Imputing Missing Data for Gene Expression Arrays. Technical report, Stanford Statistics Department. 1999.

20 O. Troyanskaya. Missing value estimation methods for DNA microarrays. Bioinformatics, vol. 17, no. 6, 2001. pp. 520-525.

21 D. Lee, S. Seung. Learning the parts of objects by non-negative matrix factorization. Nature, 401, 1999. pp. 788-791.

22 Natural Language Toolkit (NLTK) [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2015. — Режим доступа: http://www.nltk.org. — 28.05.2015.

23 Кривая ошибок (Receiver Operating Characteristic, ROC curve) [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2013. — Режим доступа: http://www.machinelearning.ru/wiki/index.php?title=ROC-кривая. — 28.05.2015.

24 New EDRM Enron Email Data Set [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2015. — Режим доступа: http://www.edrm.net/resources/data-sets/edrm-enron-email-data-set. — 28.05.2015.

25 R: Анализ и визуализация данных. Базовые графические возможности R: диаграммы размахов [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2011. — Режим доступа: http://r-analytics.blogspot.ru/2011/11/r_08.html. — 28.05.2015.

26 Upton, Graham; Cook, Ian (1996). Understanding Statistics. Oxford University Press. p. 55. ISBN 0-19-914391-9.

27 Трошин С.В. Мониторинг работы пользователей корпоративных сетей // Диссертация на соискание ученой степени к.ф.-м.н. (научные руководители Машечкин И.В. и Петровский М.И.). М.: МГУ имени М.В.Ломоносова, ф-т ВМК. 2010.

28 Царёв Д.В. Исследование и разработка системы мониторинга потоков корпоративной электронной текстовой информации // Программные системы и инструменты / Под ред. Королев Л.Н., Корухова Л.С. и др. — Т. 13. — Издательский отдел факультета ВМиК МГУ Москва, МГУ, 2012. — С. 159–173.

29 Глазкова В.В. Исследование и разработка методов построения программных средств классификации многотемных гипертекстовых документов // Диссертация на соискание ученой степени к.ф.-м.н. М.: МГУ имени М.В.Ломоносова, ф-т ВМК. 2008.

30 М. ван Стеен, Таненбаум Э. Распределенные системы. Принципы и парадигмы. - Питер, 2003.

31 Windows Event Log Reference [Электронный ресурс]. — Электрон. дан. — [Б. м.]: 2015. — Режим доступа: https://msdn.microsoft.com/en-us/library/windows/desktop/aa385785%28v=vs.85%29.aspx. — 28.05.2015.

32 Filtering IRPs and Fast I/O [Электронный ресурс]. — Электрон. дан. — [Б. м.] : Microsoft, 2014. — Режим доступа: http://msdn.microsoft.com/en-us/library/windows/hardware/ff540511(v=vs.85).aspx. — 28.05.2015.

33 IRPs Are Different From Fast I/O [Электронный ресурс]. — Электрон. дан. — [Б. м.] : Microsoft, 2014. — Режим доступа: http://msdn.microsoft.com/en-us/library/windows/hardware/ff548576(v=vs.85).aspx. — 28.05.2015.

34 Filter Manager Concepts [Электронный ресурс]. — Электрон. дан. — [Б. м.] : Microsoft, 2014. — Режим доступа: http://msdn.microsoft.com/en-us/library/windows/hardware/ff541610(v=vs.85).aspx. — 28.05.2015.

35 Communication Between User Mode and Kernel Mode [Электронный ресурс]. — Электрон. дан. — [Б. м.] : Microsoft, 2014. — Режим доступа: http://msdn.microsoft.com/en-us/library/windows/hardware/ff539277(v=vs.85).aspx. — 28.05.2015.

36 OpenSSL Project [Электронный ресурс]. — Электрон. дан. и прогр. — [Б. м.] : 2014. — Режим доступа: http://www.openssl.org/. — 28.05.2015.

37 1999 DARPA Intrusion Detection Evaluation Data Set [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 1999. — Режим доступа: http://www.ll.mit.edu/ideval/data/1999data.html. — 28.05.2015.

38 Microsoft Association Algorithm [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2015. — Режим доступа: https://msdn.microsoft.com/en-us/ms174916.aspx. — 28.05.2015.

39 Microsoft Association Algorithm Technical Reference [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2015. — Режим доступа: https://msdn.microsoft.com/en-us/cc280428.aspx. — 28.05.2015.

40 Intrusion Detection Attacks Database [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 1999. — Режим доступа: http://www.ll.mit.edu/ideval/docs/attackDB.html. — 28.05.2015.

41 Авторегрессионное интегрированное скользящее среднее (autoregressive integrated moving average, ARIMA) [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2010. — Режим доступа: http://www.machinelearning.ru/wiki/index.php?title=Autoregressive_Integrated_Moving_Average. — 28.05.2015.

42 George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, Box Jeninks, Time Series Analysis: Forecasting and Control. // A JOHN WILEY & SONS, INC. PUBLICATIONS. 4th Edition. 2008.

43 Sim T., Janakiraman R. Are digraphs good for free-text keystroke dynamics? //Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. — IEEE, 2007. — С. 1-6.

44 Kaganov V. et al. Hybrid method for active authentication using keystroke dynamics //Hybrid Intelligent Systems (HIS), 2014 14th International Conference on. – IEEE, 2014. – С. 61-66.

45 Mikhail Petrovskiy. Fuzzy Kernel-based Method for Real-time Network Intrusion Detection // Springer-Verlag, Lecture Notes in Computer Science, 2003, vol. 2877, pp. 189-200.

46 Ahmed A. A. E., Traore I. A new biometric technology based on mouse dynamics //Dependable and Secure Computing, IEEE Transactions on. - 2007. - Т. 4. - №. 3. - С. 165-179.

47 Bailey K. O., Okolica J. S., Peterson G. L. User identification and authentication using multi-modal behavioral biometrics //Computers & Security. – 2014. – Т. 43. – С. 77-89.

48 K. Jain, F. D. Griess, S. D. Connell. On-line Signature Verification. // Pattern Recognition, vol. 35, no. 12, pp. 2963–2972, 2002.

49 Yurtman A., Barshan B. Automated evaluation of physical therapy exercises using multi-template dynamic time warping on wearable sensor signals //Computer methods and programs in biomedicine. – 2014. – Т. 117. – №. 2. – С. 189-207.

50 Tax, D. One-class classification: Concept-learning in the absence of counter-examples. Doctoral Dissertation, University of Delft, The Netherlands, 2001. - 146 p.

51 Baxter R., Gu L., Hawkins S., He H., Williams G. A Comparative Study of RNN for Outlier Detection in Data Mining // IEEE ICDM 2002, pp.709-712.

52 Baxter R., Gu L., Hawkins S., He H., Williams G. Outlier Detection Using Replicator Neural Networks // Proceedings of DaWaK 2002, pp. 170-180.

53 Kevin S. Killourhy and Roy A. Maxion. Comparing Anomaly Detectors for Keystroke Dynamics. IEEE Computer Society Press, Los Alamitos, California, 2009.

54 Liu J. et al. uWave: Accelerometer-based personalized gesture recognition and its applications //Pervasive and Mobile Computing. – 2009. – Т. 5. – №. 6. – С. 657-675.

55 Shen C. et al. User authentication through mouse dynamics //Information Forensics and Security, IEEE Transactions on. - 2013. - Т. 8. - №. 1. - С. 16-30.

56 Traore I. et al. Combining mouse and keystroke dynamics biometrics for risk-based authentication in web environments //Digital Home (ICDH), 2012 Fourth International Conference on. — IEEE, 2012. С. 138-145.

57 Bours P. Continuous keystroke dynamics: A different perspective towards biometric evaluation //Information Security Technical Report. – 2012. – Т. 17. – №. 1. – С. 36-43.

58 Официальный сайт Massachusetts Institute of Technology. Lincoln Laboratory [Электронный ресурс]. — Электрон. дан. — [США] : 2015. — Режим доступа: http://www.ll.mit.edu/. — 28.05.2015.

59 Официальный сайт Defense Advanced Research Projects Agency [Электронный ресурс]. — Электрон. дан. — [США] : 2015. — Режим доступа: http://www.darpa.mil. — 28.05.2015.

60 Ubuntu Manpage: readpst - convert PST (MS Outlook Personal Folders) files [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2010. — Режим доступа: http://manpages.ubuntu.com/manpages/raring/man1/readpst.1.html. — 28.05.2015.

61 Ubuntu Manpage: pdftotext - Portable Document Format (PDF) to text converter [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2010. — Режим доступа: http://manpages.ubuntu.com/manpages/lucid/man1/pdftotext.1.html. — 28.05.2015.

62 Ubuntu Manpage: catdoc - reads MS-Word file [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2010. — Режим доступа: http://manpages.ubuntu.com/manpages/lucid/man1/catdoc.1.html. — 28.05.2015.

ПРИЛОЖЕНИЕ А

Акт №1 исполнения обязательств по работам на этапе №2 Плана-графика, выполненных за счет внебюджетных средств

ПРИЛОЖЕНИЕ А

УТВЕРЖДАЮ

"__" июня 2015 г.

Декан факультета ВМК МГУ

Академик РАН Е.И.Моисеев

АКТ №1

от 29 июня 2015 г.

исполнения обязательств по работам на этапе № 2 Плана-графика по Соглашению о предоставлении субсидии № 14.604.21.0056 от 27.06.2014г., выполненных за счет внебюджетных средств, по теме: «Исследование и разработка инновационной технологии построения программных средств обеспечения компьютерной безопасности, основанных на использовании методов машинного обучения и математической статистики для анализа данных поведенческой биометрии пользователей при работе в рамках стандартного человеко-машинного интерфейса, для решения задач активной аутентификации и идентификации пользователей, обнаружения внутренних вторжений и предотвращения попыток хищения конфиденциальной информации»

Настоящий акт составлен в том, что работы по соглашению о предоставлении субсидии № 14.604.21.0056 от 27.06.2014г. предусмотренные планом-графиком исполнения обязательств, а именно:

1. Обеспечение работоспособности рабочих станций, общего системного и специального программного обеспечения для проведения работ по проекту, проведение регламентных работ по обеспечению сохранности данных и программ, относящихся к проводимым работам по проекту;
2. Формирование наборов экспериментальных данных;

проведены в полном объеме и надлежащем качестве за счет внебюджетных источников на сумму 1 250 000 (Один миллион двести пятьдесят тысяч) рублей.

Научный руководитель проекта

Профессор И.В.Машечкин

Главный бухгалтер факультета ВМК МГУ М.В.Сидорова