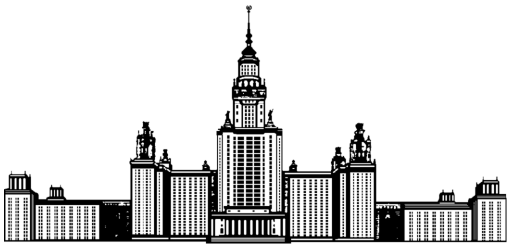


Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра интелектуальных информационных технологий

Березникер Алексей Витальевич

Динамическая аутентификация пользователей на основе анализа работы с компьютерной мышью

КУРСОВАЯ РАБОТА

Научный руководитель:
кандидат физико-математических наук,
математик М.А. Казачук

Москва, 2020 г.

Содержание

1 Аннотация

Целью данной работы является исследование существующих и разработка собственных алгоритмов динамической аутентификации пользователя на основе анализа работы с компьютерной мышью, показывающих высокое качество работы и способных работать в динамическом режиме. В работе рассматриваются существующие методы динамической аутентификации, основанные на использовании классических методов машинного обучения и нейронных сетей, а также способы построения и предобработки признакового пространства. Анализируются достоинства и недостатки различных подходов.

Мы фокусируемся на независимой от контекста системе динамической аутентификации, которая реагирует на каждое отдельное действие, выполненное пользователем.

Отметим, что динамическая аутентификация не является альтернативным решением безопасности для первоначального входа в систему, она обеспечивает дополнительную меру безопасности наряду с первоначальным логином.

2 Введение

2.1 Область применения

В настоящее время неотъемлемой частью различных сфер деятельности человека стало использование информационных систем. Огромное количество информации ограниченного доступа переносится, хранится и обрабатывается в информационных системах, что формирует потребность в обеспечении их защищенности.

Люди используют механизмы контроля доступа, такие как пароль, магнитные карты или биометрию для защиты от несанкционированного доступа другого человека. Это означает, что пользователь должен предоставить подтверждение своей личности при запуске или разблокировке системы. Однако во многих случаях люди оставляют компьютер без присмотра, временно покидая свое рабочее место, просто потому что у них отсутствует привычка выключать компьютер.

Контроль доступа к персональному компьютеру (ПК) обычно реализуется как единоразовое подтверждение личности во время первичной авторизации. Предполагается, что в течении всего сеанса в системе будет находиться зарегистрированный пользователь. К сожалению, когда компьютер оставлен без присмотра, любой человек может получить доступ к тем же источникам данных, что и легитимный пользователь.

Защита информации в информационных системах обеспечивается созданием комплексной системы защиты, одной из главных составляющих которой являются методы защиты от несанкционированного доступа [1, 2].

Основой программно-технических средств защиты от несанкционированного доступа являются процедуры идентификации и аутентификации пользователей. Идентификатором в таком случае служит уникальный признак объекта, позволяющий отличить его от других объектов. А под процедурой аутентификации подразумевается процесс проверки принадлежности субъекту доступа предъявленного им идентификатора.

2.2 Методы аутентификации

Существующие методы осуществления аутентификации можно разделить на три категории:

1. методы, основанные на обладании субъекта аутентификации некоторым секретным знанием. В качестве такого знания может выступать секретное слово, пароль или цифровой сертификат. Данный метод является самым распространенным и простым, поэтому он часто подвержен успешным атакам со стороны злоумышленников.

2. методы, основанные на наличии у субъекта идентификации некоторого физического объекта. Таким объектом может быть, например, ключ, флеш-накопитель или магнитная карта. Аутентификация по предъявлению чего-либо, чем владеет пользователь, имеет сходные недостатки с предыдущей категорией, и, кроме того, добавляется риск передачи, утери, кражи или копирования ключа. Также требуется специальное оборудование для распознавания идентификатора, используемого при аутентификации.

3. методы, основанные на собственных свойствах субъекта доступа. В качестве таких свойств могут рассматриваться биометрические данные пользователя, т.е. уникальные биологические и физиологические характеристики, которые позволяют установить личность человека. Методы аутентификации, основанные на проверке подлинности через предъявление биометрического образа называются биометрической аутентификацией.

Существующие в настоящее время методы биометрической аутентификации могут быть основаны на физиологических характеристиках человека, находящихся при нем в течение всей его жизни, или поведенческих характеристиках человека, являющихся характеристиками поведения индивидуума и отличающихся относительной устойчивостью и постоянством проявления. Данные методы разделяются на два класса:

1. Статические методы биометрической аутентификации
   Статическая аутентификация заключается в эпизодической проверке личности пользователя (например, при его входе в систему), после чего, в случае успешного прохождения, предоставляется доступ к системе. Например, проверка отпечатка пальца, сетчатки глаза или геометрии лица.

2. Динамические методы биометрической аутентификации
   Динамическая аутентификация предполагает проведение проверки личности пользователя постоянно – на протяжении всей сессии. Например, анализ голоса, клавиатурного почерка или работы с компьютерной мышью. Под «клавиатурным почерком» понимаются характеристики динамики работы пользователя с клавиатурой компьютера.

Основным недостатком методов проверки пользователей, основанных на физиологической биометрии, является то, что они требуют наличия аппаратных устройств, таких как датчики отпечатков пальцев и сканеры сетчатки, которые дороги и не всегда доступны. Хотя проверка отпечатков пальцев становится широко распространенной в ноутбуках и смартфонах, она все еще недостаточно популярна и не может быть использована в веб-приложениях. Кроме того, отпечатки пальцев могут быть скопированы.

В свою очередь, методы, основанные на поведенческой биометрии, не требуют специального оборудования, так как они используют обычные устройства ввода для сбора биометрических данных, такие как мышь и клавиатура.

Другим важным отличием между физиологической и поведенческой биометрией является временной аспект. Поведенческая биометрия может отличаться в зависимости от режима работы пользователя и времени суток, когда она была зафиксирована. Это усложняет процесс подражания для обхода системы, даже в случае перехвата части данных.

Очевидно, динамическая аутентификация пользователей является предпочтительней, так как она исключает сценарии, при которых злоумышленник получает доступ к информационной системе после того, как легитимный пользователь пройдет процедуру аутентификации. Однако данный подход затрачивает больше ресурсов компьютера, за счет непрерывной работы.

2.3 Актуальность

Таким образом, мы видим проблемы отсутствия контроля факта смены пользователя и компрометации идентификаторов, которые мы предлагаем решить использованием биометрических поведенческих характеристик пользователя для динамической аутентификации.

Достоинство нашего подхода заключается в простоте внедрения: нужно лишь устройство ввода (компьютерная мышь) и специальное программное обеспечение, позволяющее проводить анализ.

3 Постановка задачи

Задачей данной работы является исследование существующих и разработка собственных алгоритмов динамической аутентификации пользователя на основе анализа работы с компьютерной мышью, показывающих высокое качество работы и способных работать в динамическом режиме.

Наше решение должно выполнять задачи незаметно для пользователя, выявлять злоумышленника как можно быстрее, в то же время, избегая в максимально возможной степени неправильной блокировки легитимного пользователя.

4 Обзор существующих решений

4.1 Цели обзора

Целями данного обзора являются:

1. изучение показателей эффективности биометрических систем;
2. выявление достоинств и недостатков существующих подходов;
3. выявление наиболее релевантных признаков для построения модели пользователя;
4. выявление методов построения модели пользователя, показывающих наилучшее качество аутентификации;
5. поиск открытых наборов данных для проведения собственных исследований;
6. формулировка направлений дальнейших исследований.

4.2 Показатели эффективности

Согласно [1, 3], решение об аутентификации должно основываться на результате процесса сопоставления вновь представленных биометрических данных с предварительно сохраненными эталонными шаблонами. Основными метриками оценки качества работы системы аутентификации являются:

- Коэффициент ложных отклонений
  Коэффициент ложных отклонений (FRR: False Rejection Rate) – это доля случаев, когда биометрическая система не предоставляет доступ легитимному пользователю. В статистическом смысле, FRR – это ошибка I рода. FRR также известен как частота ложных несовпадений (FNMR: False Non Match Rate).

- Коэффициент ложного принятия
  Биометрическая безопасность использует коэффициент ложного принятия (FAR: False Acceptance Rate) для доли случаев, когда система предоставляет доступ неуполномоченному лицу. С точки зрения статистики, FAR – это ошибка II рода. Этот коэффициент также известен как частота ложных совпадений (FMR: False Match Rate).

- Равный коэффициент ошибок
  Одним из основных способов обобщить рабочие характеристики биометрической системы безопасности является рассмотрение коэффициента переходных ошибок (CER: Crossover Error Rate), также известного как равный коэффициент ошибок

(EER: Equal Error Rate). Это состояние, при котором ошибки FAR и FRR равны. EER дает возможность сравнивать системы. Чем меньше EER, тем лучше. Меньшее значение EER означает, что можно настроить систему так, чтобы частота ошибок как для I рода, так и для II рода была меньше.

В приложениях безопасности FAR (неавторизованный доступ) хуже, чем FRR (блокировка авторизованного пользователя). Первое может привести к нарушению конфиденциальности данных, а второе станет лишь неудобством. Конечно, может быть ситуация, в которой последствия FAR и FRR равны или что FRR хуже, но обычно это не так.

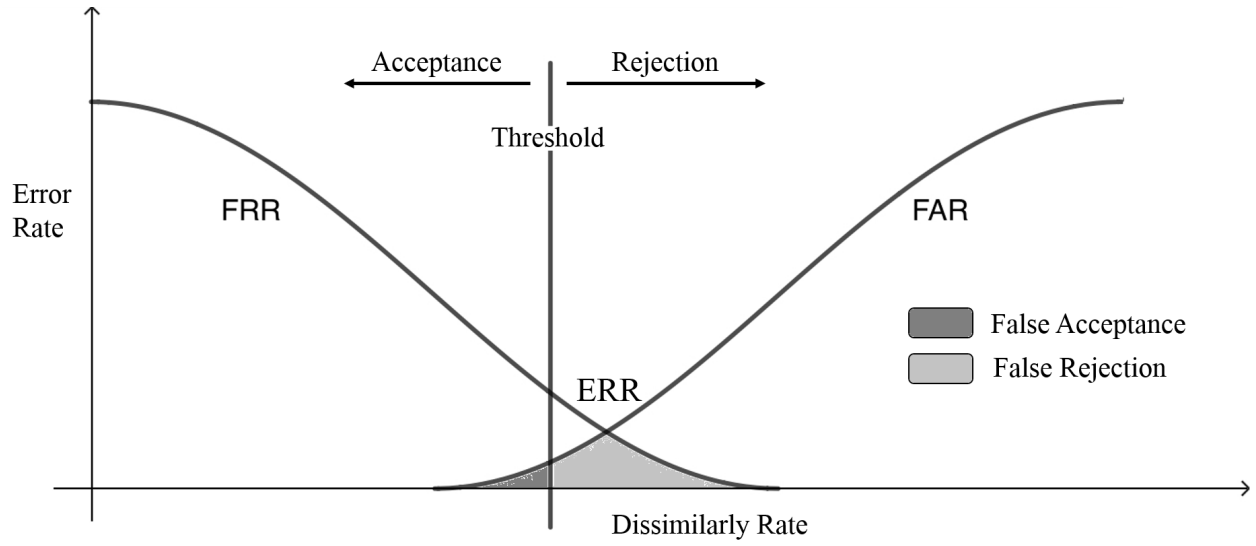


Рисунок 1. Графическая интерпретация FAR, FRR и EER

- Рабочая характеристика системы
  Рабочая характеристика системы (ROC: Receiver operating characteristic) или ROC-кривая [4] – это график компромисса между характеристиками FAR и FRR. В общем случае алгоритм принимает решение на основе порогового значения, которое определяет, насколько близко к эталону должен быть вход, чтобы его можно было считать совпадающим. Если порог уменьшится, будет меньше ложных отклонений, но больше ложных принятий. И наоборот: более высокий порог уменьшит FAR, но увеличит FRR. Качество оценивают как площадь под ROC кривой (ROC AUC: Area Under ROC Curve).

Значения метрик EER и ROC-AUC являются основными показателями эффективности для сравнения качества работы различных систем аутентификации, именно их используют в большинстве статей обзора [8 – 19].

Однако авторы ряда публикаций [5, 6, 7] верно заметили, что для системы, на самом деле, важно знать не только об обнаружении нарушителя, но и когда было совершено внедрение, т.е. сколько действий нарушитель был в состоянии совершить до обнаружения. Поэтому они предложили использовать следующие характеристики для оценки эффективности работы системы:

- Среднее количество действий злоумышленника
  Среднее количество действий злоумышленника (ANIA: Average Number of Imposter Actions) показывает, сколько действий успеет совершить нарушитель до того момента, как он будет заблокирован системой.

- Среднее количество легитимных действий
  Среднее количество легитимных действий (ANGA: Average Number of Genuine Actions) характеризует количество действий авторизованного пользователя до его ошибочной блокировки системой.

В данных терминах система будет обладать лучшими качествами при минимизации ANIA и максимизации ANGA.

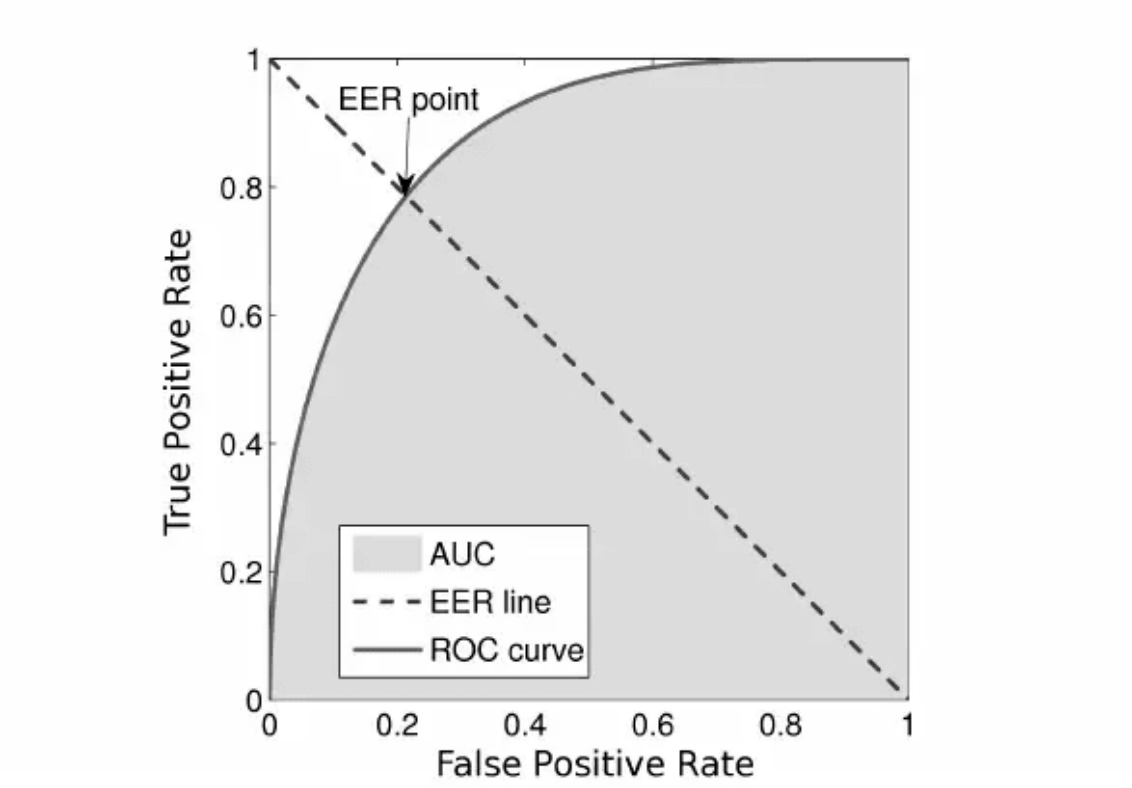


Рисунок 2. Связь между EER и ROC curve

4.3 Построение признакового пространства

Различные подходы к сбору экспериментальных данных в литературе отличаются количеством пользователей, принявших участие в исследовании, методами сбора и размерами собранных в итоге данных. Однако наиболее важным различием являются условия, в которых эти данные были собраны. Так, среда сбора данных может быть:

- Контролируемой
  Пользователь выполняет строго поставленные задачи. Например, кликает на появляющиеся на экране объекты [8, 13, 16, 17, 20], работает только с текстовым форматом данных [10, 12] или играет в компьютерные игры [14];

- Неконтролируемой
  Пользователь работает в привычной для него обстановке и выполняет повседневные для своей деятельности задачи [5, 6, 7, 9, 11, 15, 18, 19].

Сбор экспериментальных данных в контролируемой обстановке с конкретно поставленной перед пользователем задачей, возможно даже, на конкретном компьютере имеет серьезные недостатки. В этом случае пользователь будет больше сосредоточен на выполнении задачи, и его поведение не будет соответствовать его нормальному состоянию. По этой причине результаты экспериментов в контролируемой среде нельзя обобщить на реальную обстановку. Однако из-за совершенно неконтролируемого процесса сбора данных, может возникнуть проблема в различии количества данных, полученных от различных участников эксперимента.

Нами было найдено два открытых набора данных, собранных в неконтролируемой среде: BALABIT, использующийся в работах [9, 15, 18, 19], и TWOS – [15, 18, 19]. В остальных исследовательских публикациях используются собственные или недоступные наборы данных. Подробнее о структуре и происхождении полученных датасетов мы поговорим в разделе 5.1.1.

Наиболее часто используемые в статьях признаки, характеризующие особенности траектории движения мыши являются:

1. кинематические характеристики:

   (a) перемещение, длина траектории, скорость, ускорение;

2. направление движения;

3. кривизна кривой перемещения.

Авторы [5] предлагают использовать особенности траектории движения компьютерной мыши, приведенные в таблице 1, где $P_i = (x_i, y_i)$ – координаты положения курсора. Схожие признаки используют в большинстве статьей обзора для построения признаковых пространств.

Таблица 1. Исходное признаковое пространство

Признак	Формула	Признак	Формула
Direction bin	Divided into 8 bins (45°)	Curve acceleration	$\frac{Curvespeed}{\Delta t}$
Actual distance	$\sqrt{(x_n - x_0)^2 + (y_n - y_0)^2}$	Mean movement offset	$\frac{1}{n}\sum_{i=1}^{n}\begin{vmatrix} P_n - P_0 \\ P_i - P_0 \end{vmatrix} / norm(P_n - P_0)$
Actual distance bin	Divided into 20 bins	Mean movement error	$\frac{1}{n}\sum_{i=1}^{n}\left\|\begin{vmatrix} P_n - P_0 \\ P_i - P_0 \end{vmatrix} / norm(P_n - P_0)\right\|$
Curve length	$\frac{1}{n}\sum_{i=1}^{n-1}\sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}$	Mean movement variability	$\sqrt{\sum_{i=1}^{n}\frac{(y_i - movementoffset)^2}{n-2}}$
Curve length bin	Divided into 20 bins	Mean curvature	$\frac{1}{n}\sum_{i=0}^{n}\frac{\angle P(x_i, y_i)P(0,0)P(x_i, 0)}{\sqrt{x_i^2 + y_i^2}}$
Length ration	$\frac{Curvelength}{Actualdistance}$	Mean curvature change ratio	$\frac{1}{n}\sum_{i=0}^{n}\frac{\angle P(x_i, y_i)P(0,0)P(x_i, 0)}{\sqrt{(x_n - x_i)^2 + (y_n - y_i)^2}}$
Actual speed	$\frac{Actualdistance}{\Delta t}$	Mean curvature velocity	$\frac{Meancurvature}{\Delta t}$
Curve speed	$\frac{1}{n}\sum_{i=0}^{n-1}\frac{\sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}}{t_{i+1} - t_i}$	Mean angular velocity	$\frac{1}{n}\sum_{i=0}^{n-2}\frac{\angle P_i P_{i+1} P_{i+2}}{t_{i+2} - t_i}$

Также был рассмотрен еще ряд признаков из публикации [21], описанных в таблице 2, где S_{n-1} – длина траектории (см. curve length в таблице 1).

Получающиеся в итоге признаковые пространства имеют небольшую размерность и поэтому в качестве метода их предобработки зачастую используют только нормализацию [10, 14, 15], заданную формулой 6, и предварительную очистку от выбросов [15, 17].

$$\hat{x} = \frac{x - \mathsf{E}x}{\sqrt{\mathsf{D}\,x}} \tag{1}$$

где $\mathsf{E}x$ – математическое ождание наблюдения, а $\mathsf{D}\,x$ – его дисперсия.

Таблица 2. Расширение признакового пространства

Признак	Формула
minimum, maximum, mean, standard deviation and (maximum - minimum)	$x, y, v_x, v_y, v, \dot{v}, \ddot{v}$
Траектория центра масс (TCM: Trajectory Center of Mass)	$\frac{1}{S_{n-1}} \sum_{i=1}^{n-1} t_{i+1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$
Коэффициент рассеивания (SC: Scattering Coefficient)	$\frac{1}{S_{n-1}} \sum_{i=1}^{n-1} t_{i+1}^2 \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} - TCM^2$
Третий момент (M_3)	$\frac{1}{S_{n-1}} \sum_{i=1}^{n-1} t_{i+1}^3 \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$
Четвертый момент (M_4)	$\frac{1}{S_{n-1}} \sum_{i=1}^{n-1} t_{i+1}^4 \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$
Траектория кривой (TCrv: Trajectory Curvature)	$\frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{(\dot{x}^2 + \dot{y}^2)^{\frac{3}{2}}}$
Кривизна скорости (VCrv: Velocity Curvature)	$\frac{\ddot{v}}{(1 + \dot{v}^2)^{\frac{3}{2}}}$

4.4 Построение модели пользователя

Для расширения области применения и возможностей нашей системы необходимо для обучения использовать данные, собранные в неконтролируемой среде. Поэтому в дальнейшем мы рассмотрим только те исследования, в которых перед участниками экспериментов не было поставлено конкретных задач, что позволяло им работать в привычной для них обстановке. Сводная информация по таким исследованиям представлена в таблице 3.

Таблица 3. Качество работы методов из обзора

Работа	Данные	Метод	EER
[9]	BALABIT	Random Forest	0.188
[11]	28 пользователей	SVM+WV	0.125
[15]	BALABIT	LinearSVC	0.183
[18, 19]	BALABIT	CNN-LSTM	0.137
	TWOS		0.220
	BALABIT	2D-CNN	0.098
	TWOS		0.128
[5, 6, 7]	50 пользователей	OneClassSVM + TrustModel	**0.05**

Исследования, проводившиеся в Венгерском университете Трансильвании в 2019 году [9], использовали бинарный классификатор для построения модели пользователя. Они интегрировали свое решение на основе алгоритма Random Forest в ПО Weka Machine Learning toolkit [22]. Очевидно, что использование информации о поведении других пользователей приводит к повышению качества аутентификации. Однако не всегда возможно собрать «отрицательные» данные (например, динамику работы с компьютерной мышью, ритм нажатия на кнопки), поэтому мы считаем, что для обобщения работы системы недопустимо применение методов на основе многоклассовой классификации.

Случайный лес (RF: Random Forest) [23] – это алгоритм классификации, состоящий из множества деревьев решений. При построении каждого отдельного дерева он использует бэггинг (бутстрэп-агрегирование) [24] и случайные подпространства признаков, чтобы попытаться создать некоррелированный лес деревьев, предсказание которого будет точнеее прогноза любого отдельного дерева. Таким образом, в таком случайном лесу мы получаем деревья, обученные на различных подвыборках исходного набора данных и использующие различные признаки для принятия решений. Это создает некоррелированные деревья, которые защищают друг друга от ошибок.

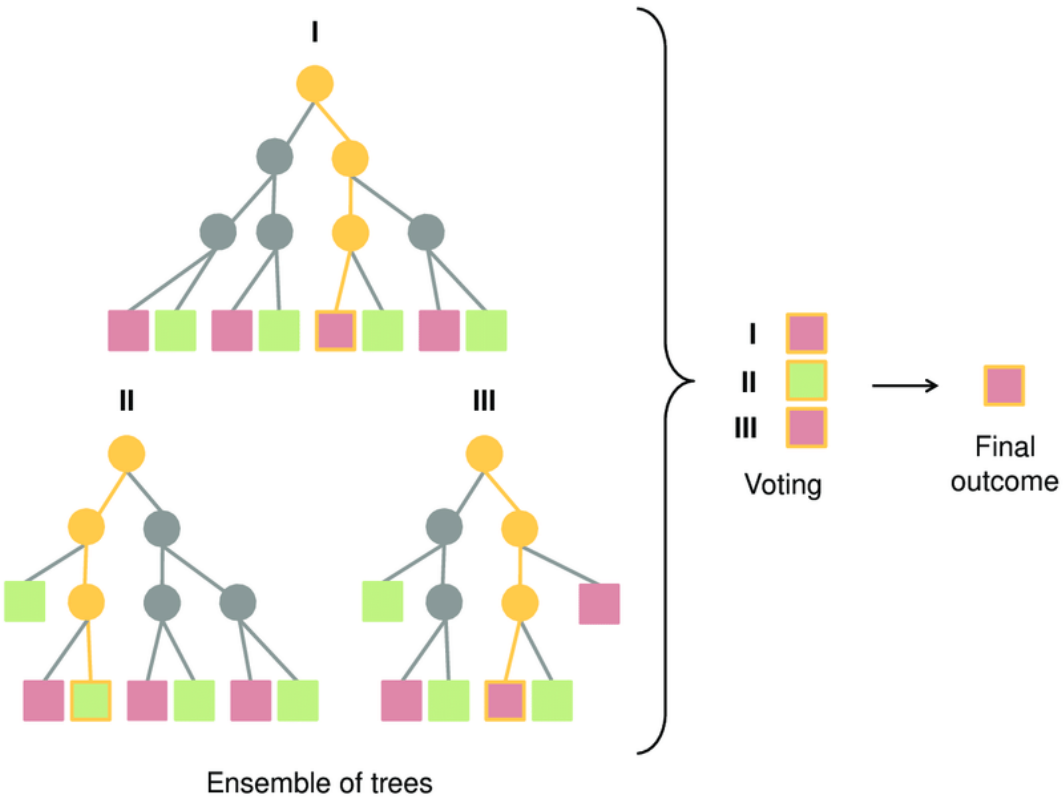


Рисунок 3. Random Forest

В следующей работе [11] рассматривается подход на основе поведенческих характеристик для классификации пользователей с использованием метода опорных векторов (SVM: Support Vector Machine) [33], дополненного методом взвешенного голосования (Weighted Voting):

$$F(b_1(x), b_2(x), \ldots, b_n(x)) = \sum_{i=1}^{n} \alpha_t b_t(x),\ x \in \mathfrak{X},\ \alpha_t \in \mathbb{R} \tag{2}$$

Основная идея SVM заключается в переводе исходного признакового пространства в пространство более высокой размерности и поиске разделяющей гиперплоскости с максимальным зазором в этом пространстве. В данной работе этот метод используется также в качестве бинарного классификатора.

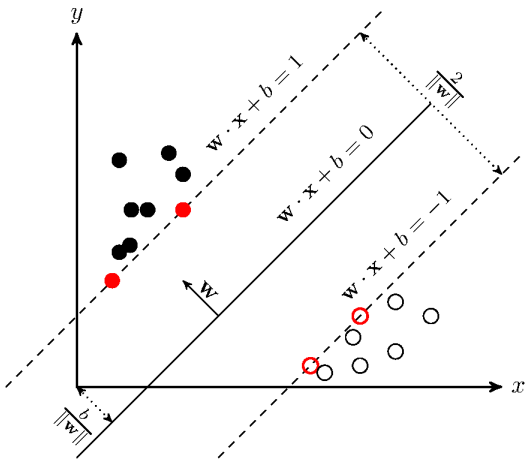


Рисунок 4. SVM

В работе [15] предложили применить разновидность SVM основанную на кластеризации данных с линейным ядром (SVC: Support Vector Clustering). Идея алгоритма аналогична SVM, только в качестве разделяющей поверхности строится гиперсфера.

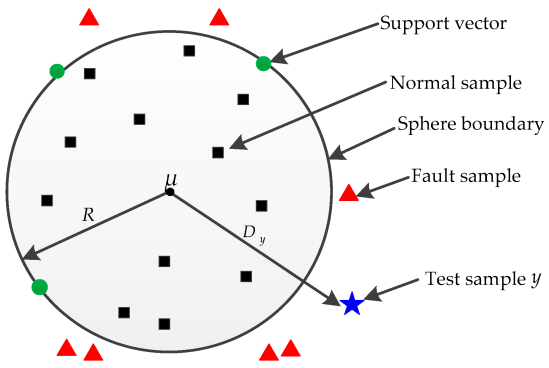


Рисунок 5. SVC

Следующими важными исследованиями в области динамической аутентификации пользователей на основе анализа работы с компьютерной мышью являются работы последних лет [18, 19], которые применили технологию глубинного обучения нейронных сетей для решения поставленной задачи. В работе предлагается гибридная архитектура нейросети на базе полносверточных и рекуррентных нейронных сетей. Учитывая последовательный характер данных движения курсора мыши, рекуррентная нейронная сеть, обычно используемая для временных рядов, кажется интуитивно понятным выбором для решения поставленной задачи.

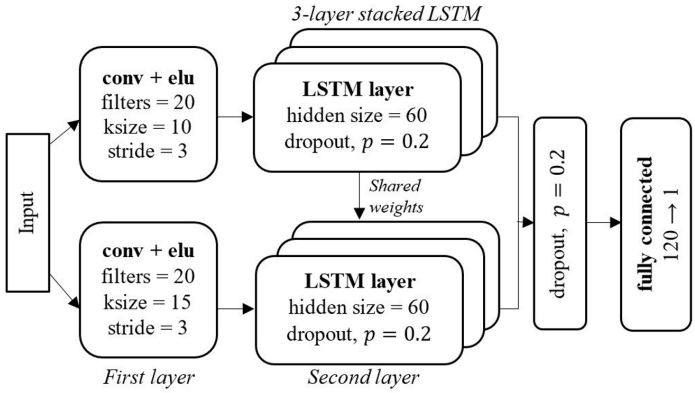


Рисунок 6. Архитектура CNN-LSTM нейросети

Также в данной работе предлагают решение проблемы небольшого количества обучающих образцов, за счет добучения (transfer learning) 2D–CNN модели. За основу была взята архитектура GoogLeNet [25]. Нейросеть была обучена в мультиклассовом режиме для всех пользователей, а затем использовалась для определения легитимности конкретного пользователя. Однако, заметим, что эта особенность влечет за собой проблемы, аналогичные использованию бинарной классификации в рассмотренных ранее работах.

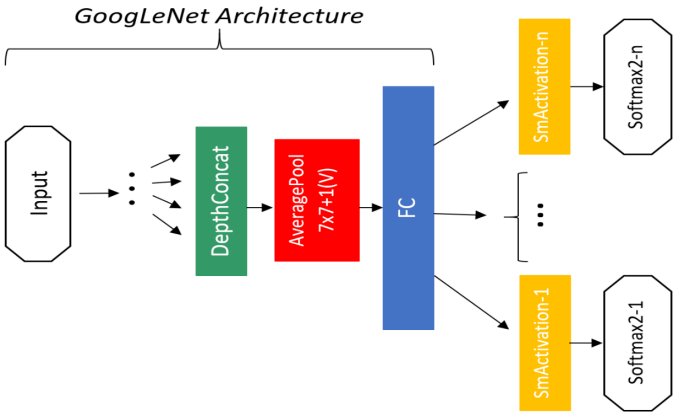


Рисунок 7. Архитектура модифицированной нейросети GoogLeNet

И, наконец, рассмотрим метод, описанный в ряде исследований [5, 6, 7], демонстрирующий наилучший результат работы в нашем обзоре. Важным вкладом этих исследований является разработка и описание т.н. модели доверия (Trust Model) в качестве динамической составляющей системы аутентификации. Основная идея данной технологии заключается в том, что доверие (или уверенность) системы в подлинности текущего пользователя зависит от его отклонений от нормального состояния. Если текущее действие выполняется в соответствии с шаблоном, хранящимся в профиле легитимного пользователя, то доверие системы к подлинности текущего пользователя будет увеличено (вознаграждение). Если между поведением подлинного и текущего пользователя будет заметное различие, то доверие системы к этому пользователю будет уменьшаться (штраф). Количество изменений уровня доверия может быть фиксированным или переменным.

Ни один человек не сможет всегда вести себя одинаково. Для легитимного пользователя это означает, что его поведение также иногда будет отклоняться от его нормального (шаблонного) поведения, что повлечет к снижению уровня доверия. Тем не менее, большинство действий легитимного пользователя будут близки к его нормальному поведению, т.е. будут приводить к повышению уровня доверия. В целом, это приведет к высокому уровню доверия системы. Для злоумышленника, однако, верно обратное. Лишь в некоторых случаях он сможет вести себя как подлинный пользователь, обманывая систему и увеличивая свой уровень доверия, но большинство его действий будут вести к снижению доверия из-за большого отклонения от поведения легитимного пользователя. Что в итоге приведет к общему снижению доверия со временем и блокировке.

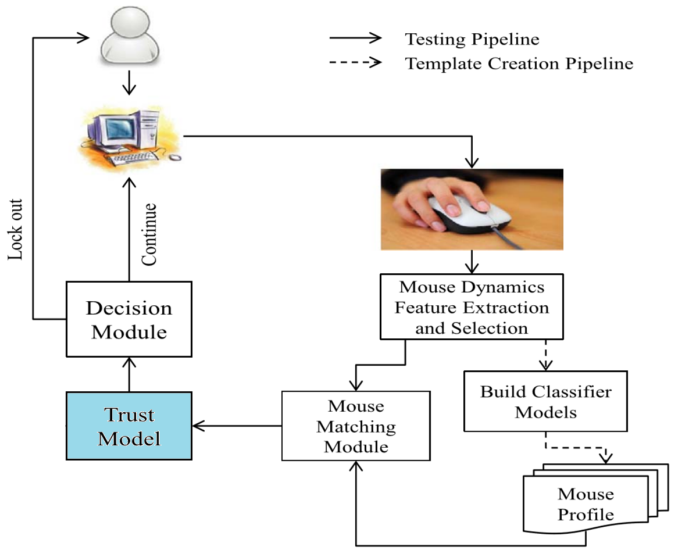


Рисунок 8. Блок-схема предлагаемой системы с Trust Model

Идеальная система должна работать таким образом, чтобы ее доверие к кому-либо, кроме легитимного пользователя, быстро уменьшалось до значения ниже заданного порогового значения блокировки T_{locout}, после чего система блокировалась и требовала статической аутентификации пользователя для продолжения работы. В такой идеальной системе легитимный пользователь также никогда не должен достигнуть уровня доверия, который привел бы к блокировке, т.е. легитимный пользователь не заметил бы присутствие системы динамической аутентификации в своей повседневной деятельности.

На рисунке 9 и 10 представлена концепция модели доверия. На рисунке 9 мы видим, как изменяется уровень доверия при сравнении профилей легитимного пользователя с тестовыми данными того же пользователя. Уровень доверия иногда падает из-за штрафов, но он никогда не достигает порога блокировки. На рисунке 10 продемонстрирована работа системы во время анализа того же пользователя с тестовыми данными злоумышленника, доверие быстро упадет ниже порога блокировки.

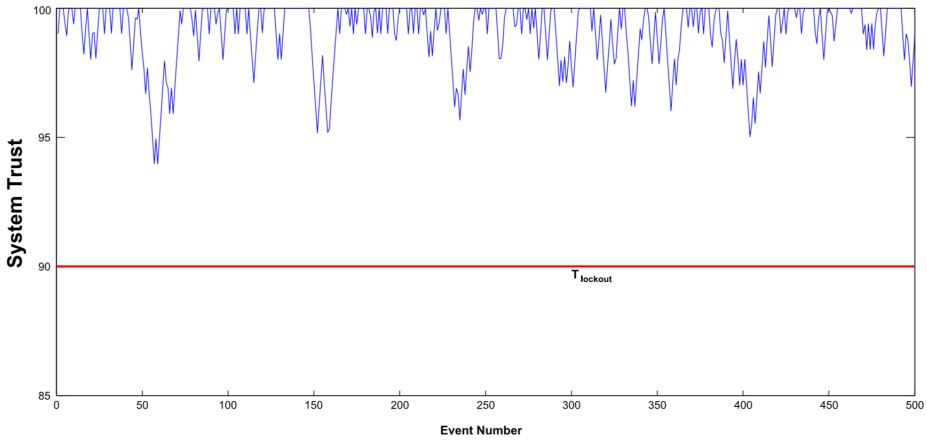


Рисунок 9. Trust Model: легитимный пользователь

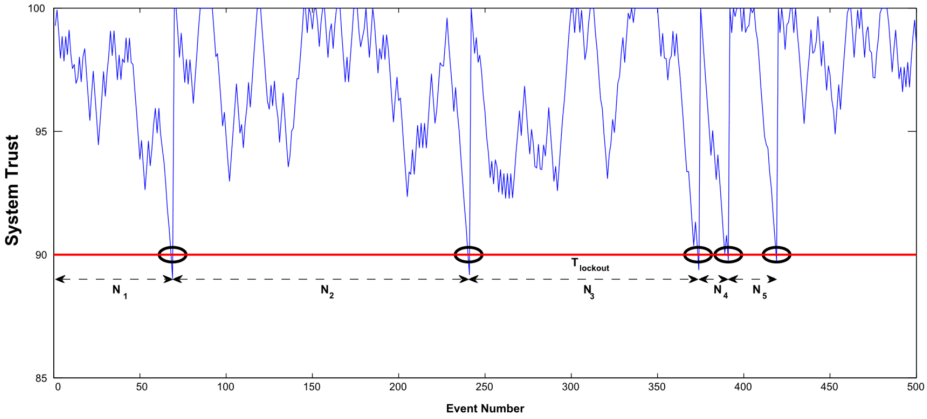


Рисунок 10. Trust Model: злоумышленник

В качестве классификатора системы авторы используют одноклассовый метод опорных векторов, что является еще одним видом рассмотренного ранее SVM. Благодаря kernel trick, модель способна проводить нелинейные разделяющие границы. Кроме того, она затачивается на обучающей выборки и поэтому отлично подходит для решения задачи поиска аномалий. Более подробный данный метод будет рассмотрен в разделе 5.3.2.

4.5 Выводы

По результатам обзора:

1. Рассмотрены существующие решения поставленной задачи и качество их работы. Основными недостатками многих работ является постановка перед участниками эксперимента конкретной задачи и использование бинарных классификаторов. Мы же хотим обеспечивать защиту пользователей в произвольной и привычной для них среде. Также проблемой для многих систем стало падение качества аутентификации после смены набора данных.

2. Были рассмотрены признаки, характеризующие траекторию движения компьютерной мыши, которые составят признаковое пространство для обучения моделей.

3. Были найдены наборы данных для проведения собственных экспериментальных исследований. Их структура будет подробно описана в разделе 5.1.1.

4. Лучший результат демонстрирует одноклассовый метод опорных векторов. Поэтому в дальнейшем мы сфокусируемся на рассмотрении классических методов машинного обучения для решения поставленной задачи. А затем рассмотрим методы, основанные на использовании нейронных сетей.

5 Исследование и построение решения задачи

5.1 Сбор данных и извлечение признаков

5.1.1 Описание наборов данных

Как было сказано ранее, для расширения области применения и возможностей нашей системы необходимо для обучения использовать данные, собранные в неконтролируемой среде.

По итогам обзора было выявлено 3 набора данных для оценки производительности поведенческих биометрических алгоритмов на основе динамики работы с компьютерной мышью в целях аутентификации пользователя.

- BALABIT [26]
  Это датасет, предоставленный компанией BalaBit IT Security, специализирующейся на разработке программного обеспечения и сервисов для информационной безопасности, в 2016 году в рамках одноименного конкурса Balabit Mouse Dynamics Challenge. Набор данных доступен для исследователей и экспертов в области IT-безопасности и науки, используется в большинстве научных статей обзора. Данные собраны в неконтролируемой среде и включают информацию о времени и позиционировании указателя компьютерной мыши. В эксперименте приняло участие 10 человек. В среднем для обучения мы имеем 43 $\pm$ 17 часов работы на каждого пользователя.

- TWOS (The Wolf Of SUTD) [27]
  Этот датасет был предоставлен во время конкурса, организованного Сингапурским университетом технологии и дизайна в марте 2017 года. Набор данных содержит действия 24 пользователей, которые собирались в неконтролируемой среде в течение 5 дней. Однако информация о взаимодействии с компьютерной мышью есть только для 4 пользователей. В среднем мы имеем 10 $\pm$ 1 час работы на каждого пользователя.

- DATAIIT
  Датасет, собранный на нашей кафедре в рамках исследования задачи динамической аутентификации пользователя в 2015 году. Данные собраны в неконтролируемой среде. В эксперименте приняло участие 20 человек. В среднем мы имеем 21 $\pm$ 5 часов работы на каждого пользователя.

5.1.2 Структура данных

Набор данных был разбит на тренировочную и тестовую части, 75% и 25% от исходных данных соответственно. Каждая часть содержит в себе записи об авторизированных сеансах всех пользователей из множества $\mathfrak{U} = \{U_1, \ldots, U_j, \ldots, U_q\}$. Работа пользователя разбита на сессии разной продолжительности $U_j = \{S_1, \ldots, S_i, \ldots, S_m\}$, где каждая сессия $S_i \in U_j$ $(i = \overline{1,m},\ j = \overline{1,q})$ содержит записи вида $(time, xpos, ypos)$. Далее каждую сессию мы разбиваем на сегменты с ограничением по временному порогу сверху (назовем эту операцию T-Time Split) и по минимальному количеству действий в этот промежуток времени снизу. По полученному сегменту строится один вектор признаков.

Графическая визуализация структуры данных представлена на рисунке 11.

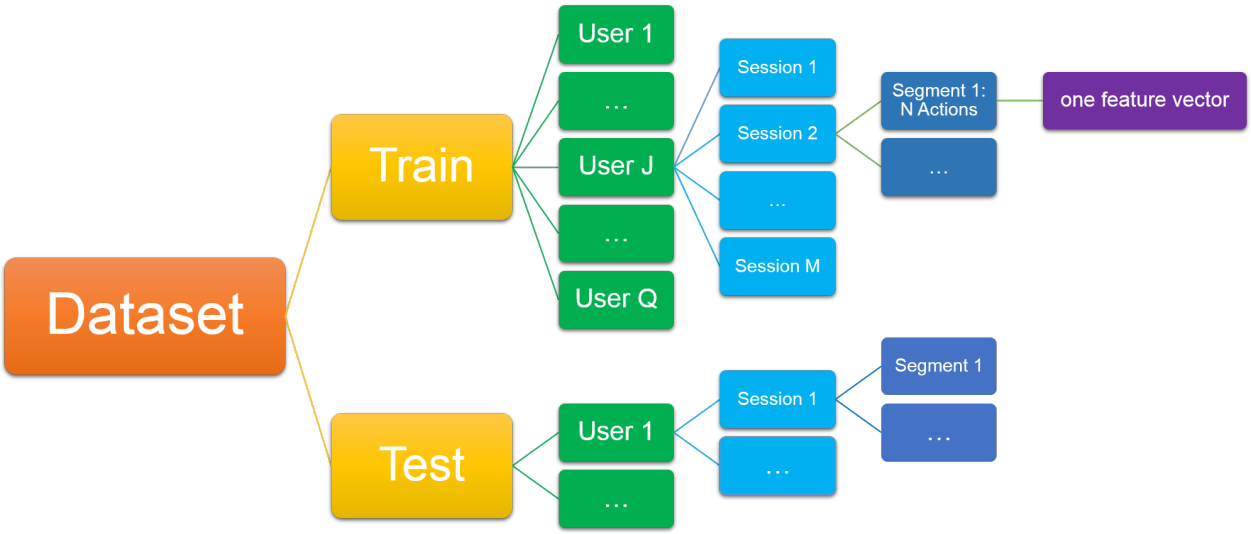


Рисунок 11. Структура данных

5.1.3 Выявленные особенности данных

Во время анализа данных были обнаружены следующие особенности:

1. полностью дублирующиеся записи в таблице;
2. дубликаты временных меток;
3. множественные дубликаты положения мыши (зацикливание);
4. сверхбольшие значения координат;

5. в наборе данных BALABIT:

   (a) в состоянии прокрутки колеса (Scroll) положение мыши перемещается в начало координат;

   (b) основная работа пользователей происходит в левом нижнем углу экрана.

Все особенности были устранены из наборов данных.

5.2 Построение признакового пространства

Так как 16 признаков, объявленных в таблице 1, может оказаться недостаточно для уникальности отдельно взятого вектора признаков, мы сформируем признаковое пространство, как конкатенацию признаков из таблицы 1 и 2. Получившееся признаковое пространство имеет размерность 78 признаков на вектор.

Для обработки признакового пространство нами было предложено использовать следующие методы:

5.2.1 Квантование

Квантование [29] – процесс предобработки данных, при котором непрерывные данные преобразуются в дискретные путем замены значений интервалами, каждый из которых представляет некоторый диапазон. Различают два основных метода квантования:

1. Интервальный
   Диапазон изменения значений признака разделяется на равные интервалы. Данный метод используется, если значения равномерно распределены на всей области значений, т.е. в результате квантования не будет интервалов, в которых значения почти отсутствуют или, наоборот, плотных интервалов.

2. Квантильный
   Ширину интервалов выбирают таким образом, чтобы в каждый из них попало примерно одинаковое количество значений.

Согласно [30], одним из наиболее популярных методов предобработки признаков, используемых для мультимодальных распределений, является именно квантильная дискретизация. Этот подход ранее не применялся к анализу данных работы с компьютерной мышью.

На рисунке 12 продемонстрировано разбиение данных на 4 квантиля (т.н. квартили).

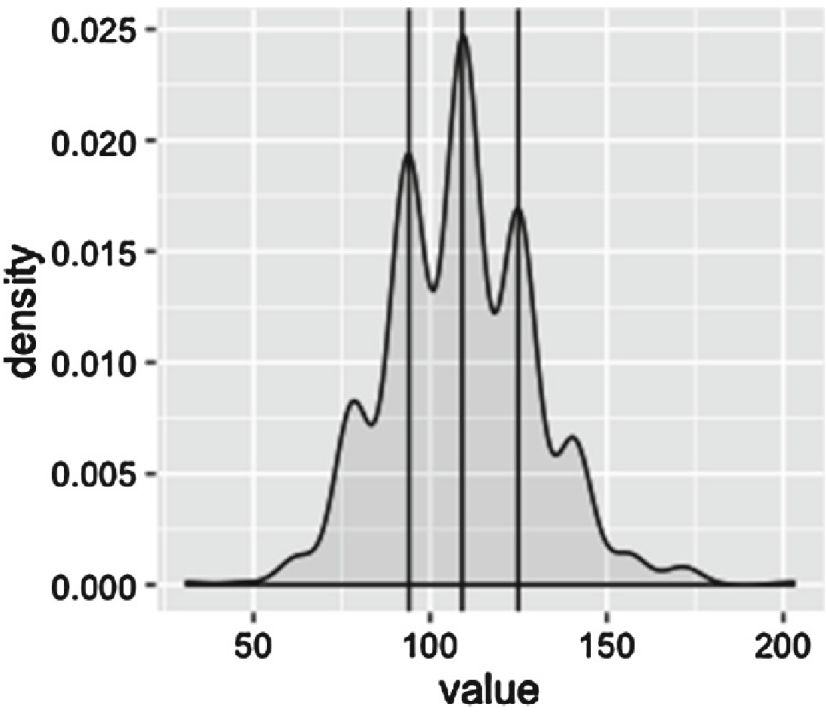


Рисунок 12. Квантильная дискретизация данных

5.2.2 One-Hot Encoding

One-Hot Encoding – это кодировка, с помощью которой категориальные признаки преобразуются в дискретную форму для лучшего качества прогнозирования методами машинного обучения. Под категориальным признаком подразумевается признак, значения которого обозначают принадлежность объекта к какой-то категории. Например, город со значением из множества {Москва, Анапа, Сургут, ...}.

Пусть $\mathfrak{V} = \{V_1, \ldots, V_i, \ldots, V_s\}$ – множество уникальных значений категориального признака V. Тогда каждое $V_i \in \mathfrak{V}$ $(i = \overline{1, s})$ преобразуется в s новых признаков, где i-й признак будет принимать значение 1, а все остальные – 0.

Данный подход устраняет проблему иерархии (порядка), возникающую при вещественной кодировке категориальных признаков, когда $V_i \mapsto i \in \mathbb{N}(\mathbb{R})$.

Однако данный метод также имеет недостаток, заключающийся в добавлении большого количества признаков в набор данных. Это может привести к значительному увеличению признакового пространства, если категориальный признак будет иметь много уникальных значений.

5.2.3 Градиентный бустинг

Преимущество использования ансамблей деревьев решений, таких как градиентный бустинг [28], заключается в том, что они могут предоставлять оценку важности признаков из обученной модели.

Как правило, важность обеспечивает оценку, которая указывает, насколько полезным был каждый признак при построении деревьев решений в модели. Чем больше атрибут используется для принятия ключевых решений с деревьями решений, тем выше его относительная важность.

Важность рассчитывается для отдельного дерева решений, затем значения характеристик усредняются по всем деревьям решений в модели.

5.3 Построение модели пользователя

5.3.1 Задача поиска аномалий

Согласно [31, 32], в анализе данных есть два направления, которые занимаются поиском аномалий: детектирование выбросов (Outlier Detection) и детектирование новизны (Novelty Detection). Как и выброс, новый объект — это объект, который отличается по своим свойствам от объектов обучающей выборки. Но в отличие от выброса, его в самой выборке пока нет. Задачей обнаружения новизны является идентификация новых или неизвестных данных, о которых система не знает во время обучения. Это означает, что задача поиска аномалий относится к классу unsupervised learning, т.е является задачей обучения без учителя.

Например, при анализе замеров температуры и отбрасывании аномально больших или маленьких значений, происходит борьба с выбросами. А при создании алгоритма, который для каждого нового замера оценивает, насколько он похож на предыдущие, и выбрасывает аномальные — детектирование новизны.

Областей, где возникает задача поиска аномалий, достаточно много:

1. обнаружение вторжений;
2. обнаружение подозрительных банковских операций;
3. обнаружение неполадок в механизмах;
4. обнаружение инсайдров на бирже;
5. медицинская диагностика;
6. сейсмология.

Особенности этой задачи заключаются в том, что присутствует явный дисбаланс классов (аномалии достаточно редки), а также в том, что тренировочные данные уже могут содержать аномальные наблюдения, о которых мы можем не знать. Такие наблюдения нужно идентифицировать и удалить из тренировочных данных на этапе предобработки признаков, чтобы система не приняла аномальные наблюдения за нормальные.

Стоит отметить, что, в общем случае, мы не можем свести задачу поиска аномалии к бинарной классификации, т.к. в реальной жизни у нас может не оказаться размеченных аномальных наблюдений для обучения.

В следующих разделах мы рассмотрим классические методы машинного обучения для решения задачи поиска аномалий.

5.3.2 One Class SVM

Метод опорных векторов (SVM: Support Vector Machine) [33] – это семейство мощного статистического обучения методов классификации и регрессии. Они доказали свою эффективность во многих практических применениях. SVM основывается на индуктивном принципе минимизации структурных рисков (SRM: Structural Risk Minimization).

Одноклассовый метод опорных векторов (OC-SVM: One Class SVM) [34] был предложен как расширение SVM, для выявления новизны или выбросов в наборах данных. Важную роль OC-SVM играет в области обнаружения вторжений.

Основная идея алгоритма заключается в поиске гиперплоскости в признаковом пространстве, которая максимально отдаляет данные от начала координат.

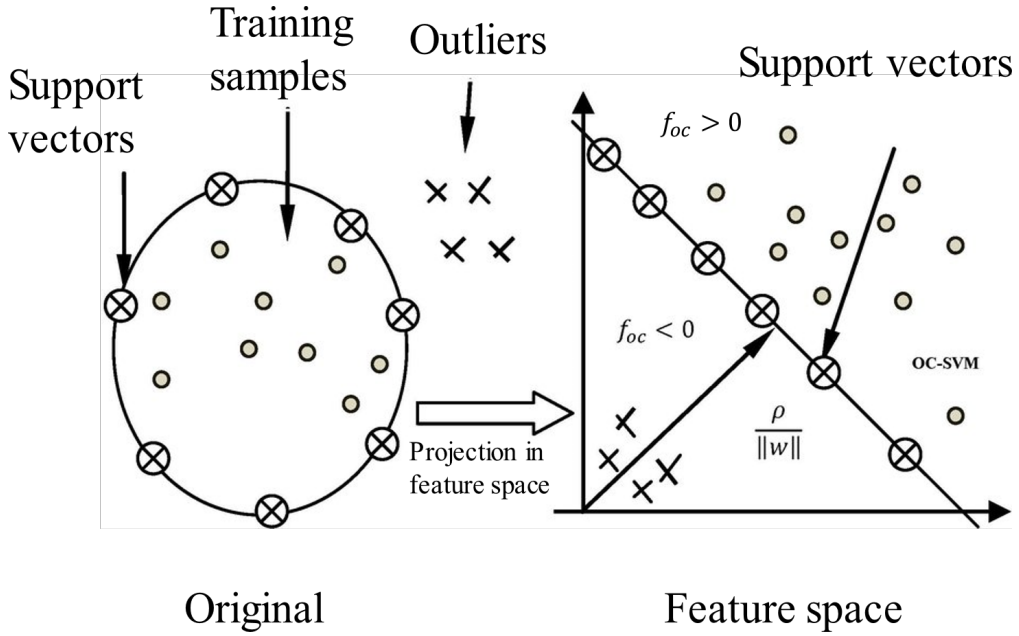


Рисунок 13. One Class SVM: принцип работы

5.3.3 Isolation Forest

Изоляционный лес (IF: Isolation Forest) [35], как и любой другой метод ансамбля деревьев, построен на основе деревьев решений (DT: Decision Tree). Алгоритм обучения строит ансамбль изоляционных деревьев на основе рекурсивной и рандомизированной процедуры структурированного разбиения: сначала случайным образом выбирается объект, а затем выбирается случайное значение между минимальным и максимальным значением выбранного объекта.

Выбросов в данных немного и, зачастую, они находятся дальше от обычных наблюдений в пространстве признаков. Вот почему при использовании такого случайного разбиения они должны быть идентифицированы ближе к корню дерева. В случае аномальных наблюдений необходимо меньше расщеплений.

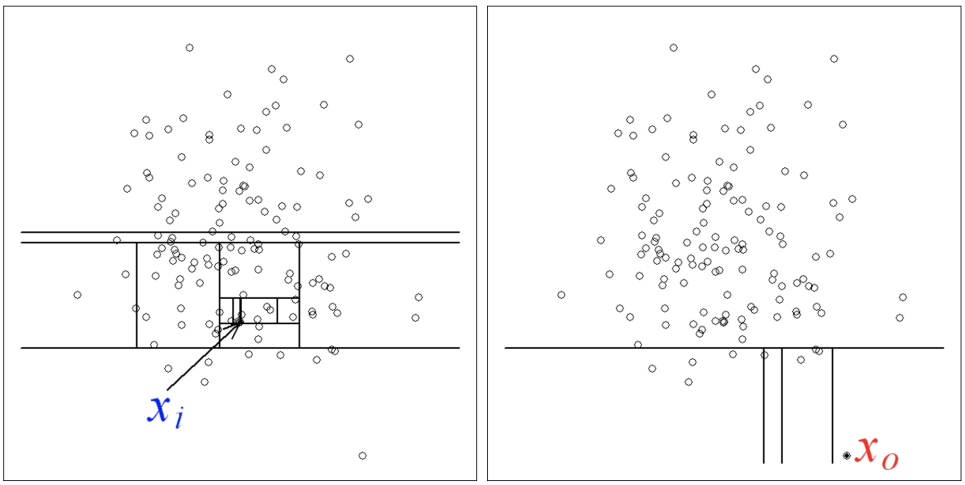


Рисунок 14. Isolation Forest: Определение нормальных и аномальных наблюдений

Так, на рисунке 14 синим цветом показано нормальное наблюдение x_i, а красным – аномальное x_0. Здесь аномальная точка была разделена в несколько шагов, в то время как для изоляции нормального наблюдения потребовалось больше разделений.

Для принятия решения об аномальности наблюдения алгоритм используют следующую функцию:

$$S(x, n) = 2^{-\frac{\mathsf{E}(h(x))}{c(n)}} \tag{3}$$

где $\mathsf{E}(h(x))$ – средняя длина пути наблюдения x , $c(n)$ – средняя длина пути неудачного поиска в бинарном дереве поиска, а n – количество внешних узлов.

Каждое наблюдение получает индекс аномальности, на основе которого принимается решение:

- Оценка, близкая к 1, указывает на аномальность наблюдения;
- Оценка, близкая к 0, указывает на нормальное поведение наблюдения.

5.3.4 Local Outlier Factor

Локальный уровень выброса (LOF: Local Outlier Factor) [36] обнаруживает выбросы на основе локальной плотности точек, которые являются выбросами по отношению к их локальной окрестности, а не по отношению к глобальному распределению данных. Чем выше значение LOF для наблюдения, тем более аномальным оно является. Точка считается аномальной, если ее локальная плотность значительно отличается от плотности соседних точек. LOF в точке P определяется как:

$$LOF(P) = \sum\nolimits_1 (\text{расстояние до соседей}) \cdot \sum\nolimits_2 (\text{локальная плотность соседей}) \qquad (4)$$

Таким образом, LOF в точке P может принимать:

- большое значение, если точка P находится далеко от своих соседей и ее соседи имеют высокую локальную плотность (т.е. они близки к своим соседям):
  $\uparrow LOF(P) = \sum_1 (\text{большое расстояние}) \cdot \sum_2 (\text{большая плотность})$
- среднее значение, если P находится далеко от своих соседей и ее соседи имеют малую локальную плотность:
  $\sim LOF(P) = \sum_1 (\text{большое расстояние}) \cdot \sum_2 (\text{маленькая плотность})$
- маленькое значение, если P находится близко к своим соседям и ее соседи имеют малую локальную плотность:
  $\downarrow LOF(P) = \sum_1 (\text{маленькое расстояние}) \cdot \sum_2 (\text{маленькая плотность})$

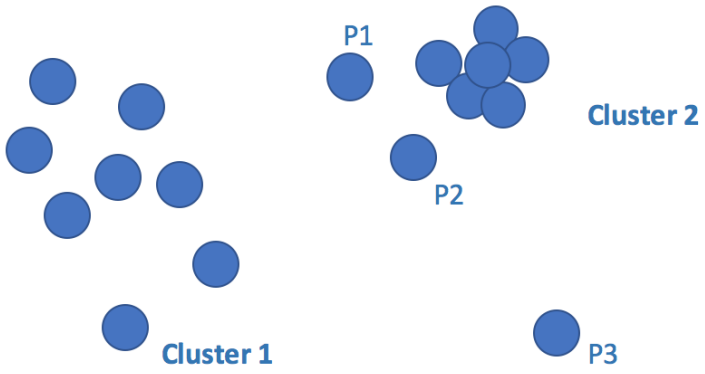


Рисунок 15. Local Outlier Factor: Определение нормальных и аномальных наблюдений

Так в приведенном пространстве признаков на рисунке 15 метод LOF, кроме очевидного аномального наблюдения P3, определит точки P1 и P2 как выбросы, которые являются локальными для второго кластера.

5.3.5 Elliptic Envelope

Эллипсоидальная аппроксимация данных (EE: Elliptic Envelope) [37] – ковариационная оценка, предполагающая, что данные имеют распределение Гаусса. Ограничивающий контур имеют эллиптическую форму.

Процедура EE использует алгоритм FAST-Minimum Covariance Determinate для оценки размера и формы эллипса. Данный алгоритм выбирает неперекрывающиеся подвыборки данных и вычисляет среднее значение μ и ковариационную матрицу C в признаковом пространстве для каждой подвыборки. Расстояние Махаланобиса dMH вычисляется для каждого многомерного вектора данных (признака) x в каждой подвыборке, по формуле:

$$dMH = \sqrt{(x-\mu)^T C (x-\mu)} \tag{5}$$

Затем данные упорядочивают по возрастанию значений dMH. Эта процедура повторяется до сходимости определителя матрицы ковариации. Ковариационная матрица с наименьшим определителем из всех подвыборок образует эллипс, который охватывает часть исходных данных. Данные в пределах поверхности эллипса считаются нормальными, а вне эллипса – аномальными.

5.3.6 Визуализация работы методов

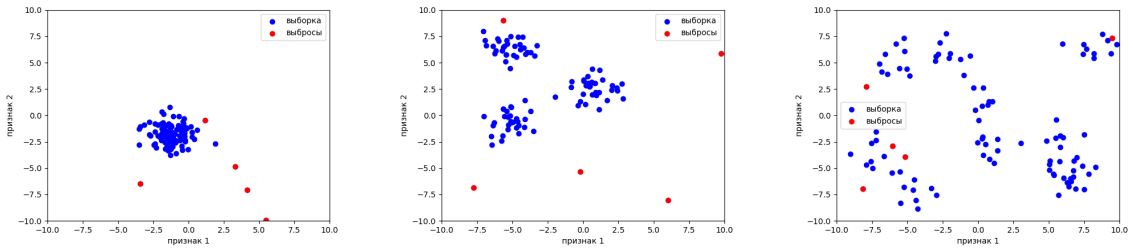
В данном разделе мы визуализируем работу описанных ранее методов, определим достоинства и недостатки каждого с целью выявления наилучшего кандидата для решения поставленной задачи.

Для этого сформируем двумерный набор тестовых данных (toy dataset), содержащий 100 объектов, 5% которых будут выбросами. Первый набор данных (a) состоит из одного кластера, второй (b) – из трех, а последний набор (c) – из десяти, что позволит сильнее разнести объекты в пространстве признаков. Синим цветом помечены нормальные наблюдения, красным – аномальные (выбросы). Алгоритмы машинного обучения взяты из бесплатной библиотеки машинного обучения для языка программирования Python – scikit-learn.

One Class SVM

Гиперпараметры

- kernel (ядро): радиальное (rbf). Доступны также линейное (linear), сигмоидальное (sigmoid) и полиномиальное (poly), но они покзывают низкое качество идентификации в задаче поиска аномалий:
- nu (ν) $\in [0, 1]$ – верхняя граница на % ошибок. В нашем случае nu=0.05.



(a) (b) (c)

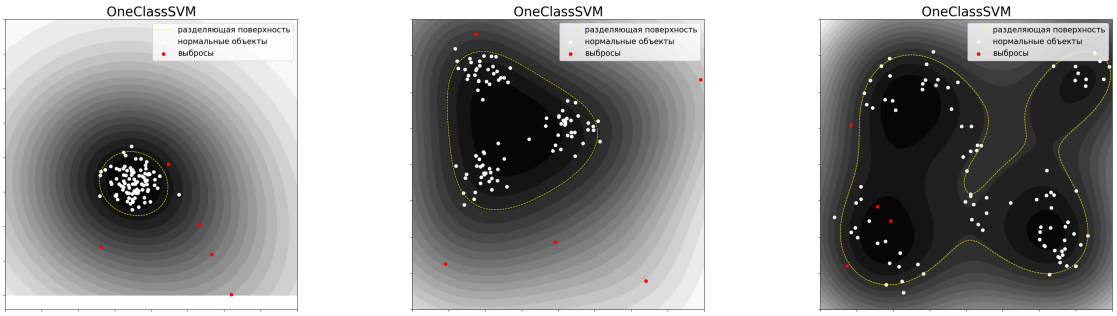
Рисунок 16. Тестовые наборы данных

Достоинства

- благодаря kernel trick, модель способна проводить нелинейные разделяющие границы. Идея заключается в том, что классы, линейно неразделимые в текущем признаковом пространстве, могут стать разделимыми в пространстве более высокой размерности.

Недостатки

- может очень сильно переобучиться и выдавать большое количество ложно отрицательных результатов;
- нужно быть абсолютно уверенным, что тренировочные данные не содержат никаких выбросов, иначе алгоритм будет считать их нормальными наблюдениями.



(a) (b) (c)

Рисунок 17. One Class SVM: визуализация работы метода на тестовых данных

Как видно на рисунке 17 одноклассовый метод опорных векторов хорошо проводит разделяющую поверхность, охватывая большую часть тренировочных данных.

Алгоритм чаще используется для детектирования новизны, т.к. затачивается под обучающую выборку и поэтому лучше подходит для решения нашей задачи.

Isolation Forest

Гиперпараметры

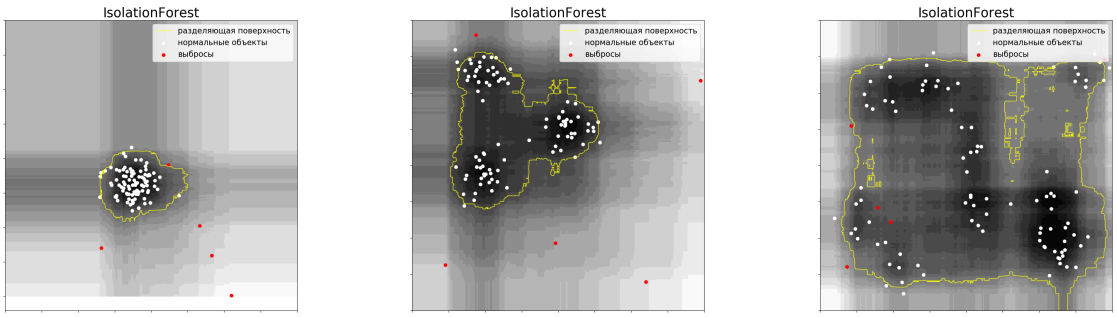
- n_estimators – число деревьев;
- max_samples – объем выборки для построения одного дерева;
- contamination – доля выбросов в выборке.

Достоинства

- алгоритм распознает аномалии различных видов: как изолированные точки с малой локальной плотностью, так и небольшие кластеры аномалий;
- эффективность: сложность алгоритма $O(n \log n)$;
- устойчив к проклятию размерности.

Недостатки

- алгоритм больше подходит для поиска выбросов в данных, а не новизны.



(a) (b) (c)

Рисунок 18. Isolation Forest: визуализация работы метода на тестовых данных

Как видно на рисунке 18 изоляционный лес также хорошо проводит разделяющую поверхность.

Алгоритм хорошо отлавливает именно выбросы.

Local Outlier Factor

Гиперпараметры

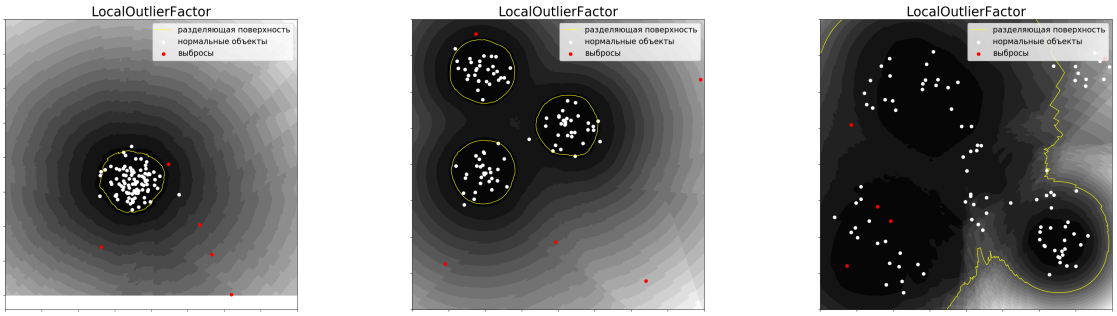
- n_neighbors – количество соседей;
- contamination – доля выбросов в выборке.

Достоинства

- метод способен выявить локальные выбросы в наборе данных.

Недостатки

- т.к. метод является метрическим, то он хорошо определяет выбросы только если относительное положение разных точек отражает различие в поведении;
- получающиеся значения аномальности труднее интерпретировать.



(a) (b) (c)

Рисунок 19. Local Outlier Factor: визуализация работы метода на тестовых данных

Как видно на рисунке 19, алгоритм отлично справляется с поставленной задачей, когда кластеры данных явно выражены. Однако в случае разрозненности наблюдений, как на рисунке (c), алгоритм может ошибочно объявить небольшие кластеры наблюдений аномальными.

Мы будем использовать этот метод в качестве предобработки тренировочного набора данных от выбросов.

Elliptic Envelope

Гиперпараметры

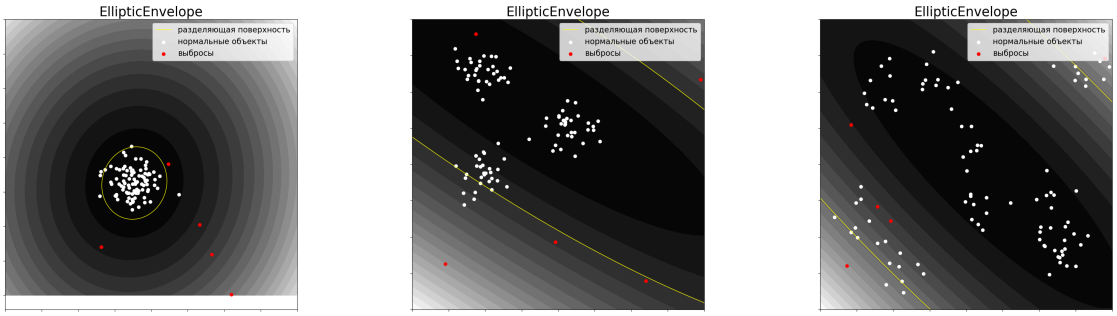
- contamination – доля выбросов в выборке.

Достоинства

- нет необходимости подбирать гиперпараметры модели.

Недостатки

- метод успешно работает только на нормально распределенных одномодальных данных.



(a) (b) (c)

Рисунок 20. Elliptic Envelope: визуализация работы метода на тестовых данных

На рисунке 20 видно, что эллипсоидальная аппроксимация данных работает плохо, если все данные не образуют один локальный кластер. Так на (b) и (c) разделяющая поверхность слишком обширна и охватывает огромную часть пространства, оставив за пределами существенную часть данных.

6 Описание практической части

6.1 Программная реализация

В практической части работы реализован автоматизированный компонентный экспериментальный стенд, архитектура которого представлена на рисунке 21, на языке программирования Python 3 с использованием ряда open source библиотек, основными из которых являются следующие:

- Pandas v1.0.0: для анализа и обработки данных;
- NumPy v1.18.1: для эффективной работы с массивами;
- Scikit-learn v0.22.1: для решения задач классического машинного обучения;
- Matplotlib v3.1.3: для визуализации данных двумерной графикой;
- Tensorflow v2.1.0: для решения задач построения и обучения нейронных сетей.

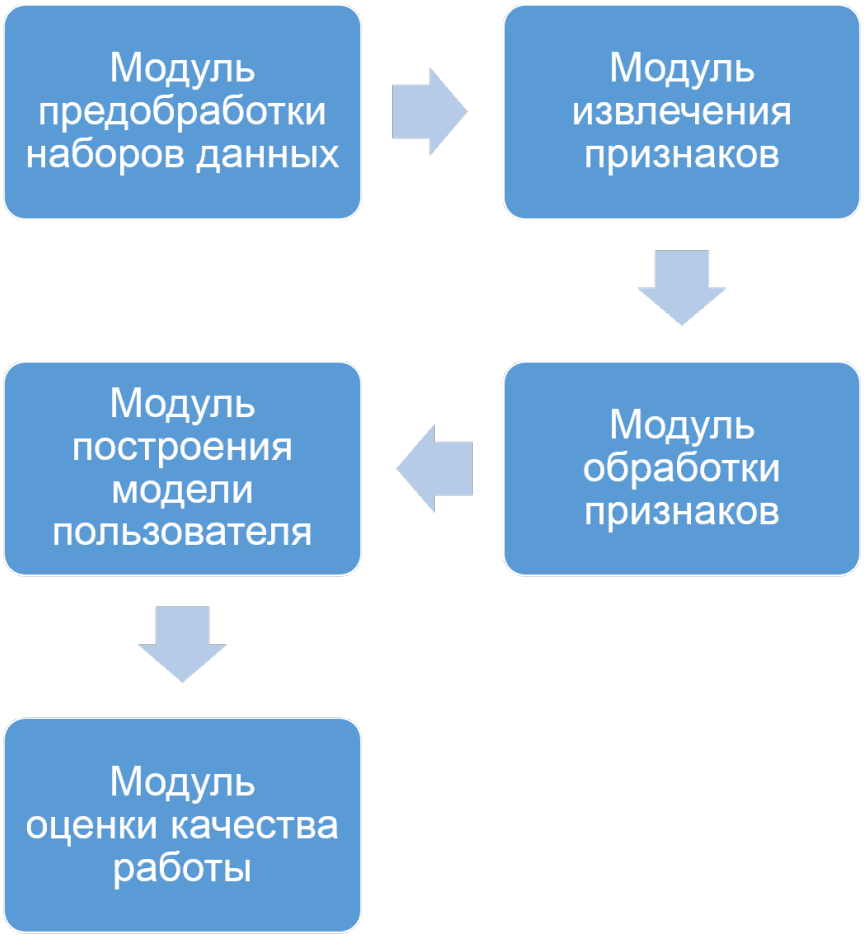


Рисунок 21. Архитектура экспериментального стенда

Модуль предобработки данных

В рамках данного модуля происходит унифицирование наборов данных, описанных в разделе 5.1.1, в соответвии со структурой, заданной в разделе 5.1.2. Удаляются все особенности, выявленные в разделе 5.1.3. Преобразованные наборы данных сохраняются для следующего этапа конвейера.

Модуль извлечения признаков

На данном этапе из предобработанных наборов данных происходит выделение признаков, описанных в разделе 4.3 обзора (таблица 1 и таблица 2), и сохранение полученных сессий пользователей.

Модуль обработки признаков

Далее происходит объединение всех сессий и формирование полного исходного признакового пространства. Также в этом модуле доступны методы предобработки признаковых пространств, описанные в разделе 5.2, результатом работы которых являются модифицированные пространства.

Модуль построения модели пользователя

На следующем этапе экспериментального стенда в полученных признаковых пространствах происходит обучение модели пользователя с использованием классических методов машинного обучения, рассмотренных в разделе 5.3. Подбираются гиперпараметры, при которых модель демонстрирует наилучшее качество работы. Полученные модели сохраняются для возможности дальнейшего использования.

Модуль оценки качества работы

В заключительной части происходит оценка качества работы обученных моделей на тестовых наборах данных. Проводится перекрестная проверка one-vs-all, в условиях которой в систему защиты текущего пользователя пытаются проникнуть все остальные пользователи данного набора данных. Усредненное по всем пользователям значение метрики ROC-AUC, описанной в разделе 4.2, считалось итоговой оценкой качества работы метода. Результатом данного модуля является визуализация оценки качества и сохранение полученных графиков. Пример визуализации качества работы метода во время такой «атаки» продемонстрирован на рисунке 22.

Также в течение работы каждого из описанных модулей ведется подробная запись в log-файлы для сбора статистики и дальнейших исследований. Весь код проекта доступен по ссылке в репозитории: https://github.com/Berezniker/HiddenMouse. Общий объем кода составил порядка 1000 строк.

6.2 Экспериментальные исследования

В результате работы модуля предобработки были удалены особенности из наборов данных, что составило для каждого пользователя порядка:

- $22 \pm 3\%$ записей в датасете TWOS;
- $21 \pm 7\%$ записей в датасете BALABIT;
- $4 \pm 1\%$ записей в датасете DATAIIT.

Заметим, что количество неотфильтрованных записей в открытых наборах данных значительно превосходит тот же показатель для датасета, собранного нашей кафедрой в рамках исследования задачи динамической аутентификации пользователя.

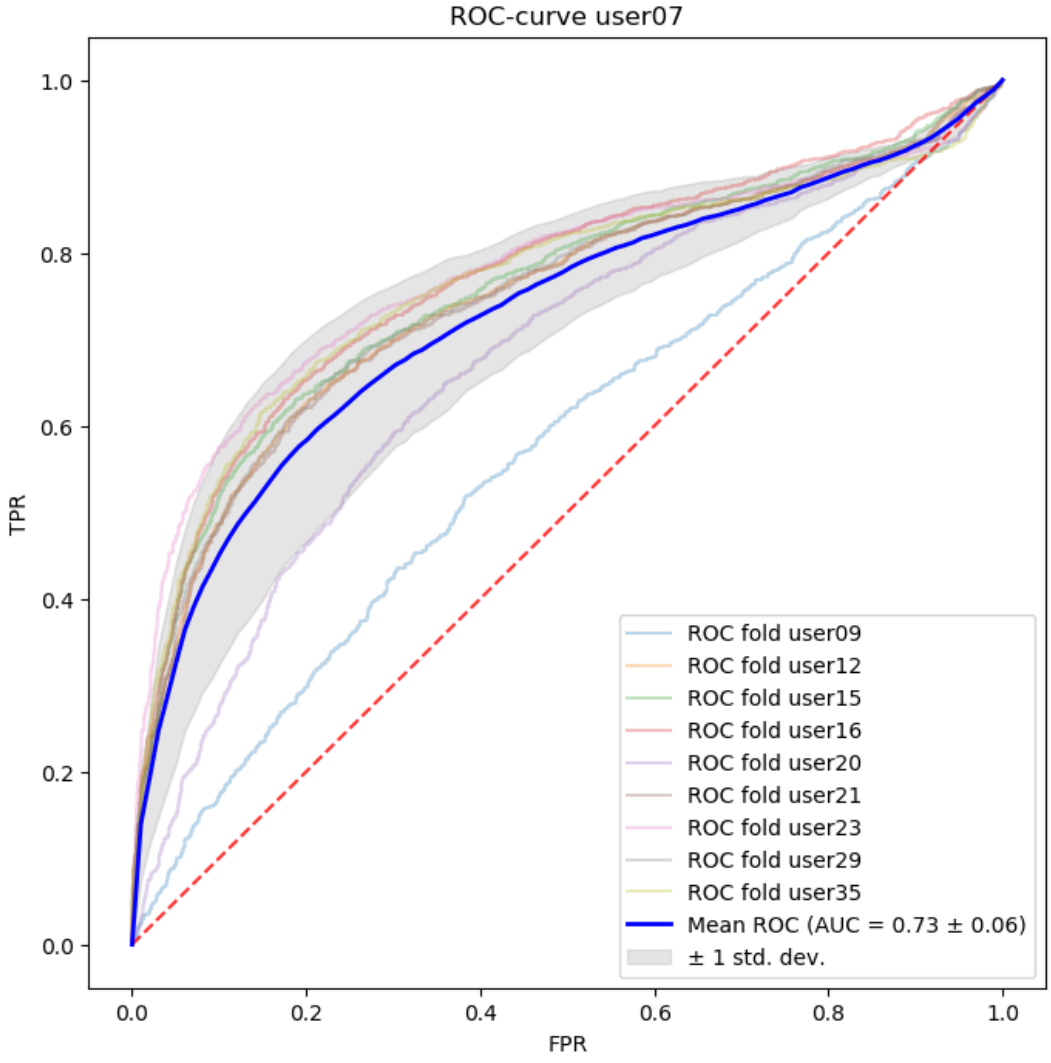


Рисунок 22. Пример оценки работы метода

В модуле извлечения признаков были поставлены эксперименты, результаты которых можно наблюдать в таблице 4, с разбиением по временному порогу в T-Time Split разбиении, заданному в разделе 5.1.2. Были сделаны следующие выводы:

- $T = 1 - 2$ секунды: долго выделять признаки и обучать модель, мало действий для уникальности одного признака;
- $T = 3 - 5$ секунд: *оптимальнo*;
- $T = 10 - 15$ секунд: очень быстро, но может оказаться мало векторов для обучения, много действий за один этап;
- $T > 15$ секунд: редкая проверка подлинности, пропадает эффект динамики системы, что может стать уязвимостью для внедрения.

Таблица 4. Разбиение сессий по временному порогу

Dataset	Time, h	T-Time Split, sec			
		2	5	10	15
BALABIT	43 ± 17	9600 ± 4100	4700 ± 2000	2800 ± 1200	2000 ± 800
DATAIIT	21 ± 5	7000 ± 2500	3400 ± 1000	2000 ± 600	1400 ± 400
TWOS	10 ± 1	3000 ± 500	1550 ± 200	800 ± 100	600 ± 70

С использованием модуля обработки признаков из предобработанных датасетов были выделены признаки, заданные в таблице 1 и таблице 2, описанные в разделе 4.3 обзора. Получившееся признаковое пространство имело размерность 78 признаков на вектор. Также были сформированны модифицированные признаковые пространства в соответвии с методами предобработки, описанными в разделе 5.2. В результате чего:

- One-Hot кодировка (раздел 5.2.2) была применена только к корзинным (bin) признакам, что расширило признаковое пространство до 131 признака;
- Квантильная дискретизация (раздел 5.2.1) была применена ко всем признакам. Построено разбиение на 4 и 10 квантилей.

Перед обучением к исходному тренировочному набору данных было применено преобразование (6), стандартизирующее значение всех признаков путем вычитания среднего значения μ и масштабирования до единичной дисперсии σ. Преобразование с теми же параметрами было применено в дальнейшем и к тестовому набору.

$$z = \frac{x - \mu}{\sqrt{\sigma}} \tag{6}$$

После чего, на получившихся признаковых пространствах была обучена модель с применением градиентного бустинга (раздел 5.2.3) для выявления наиболее важных признаков.

Результаты экспериментов, представленные в таблице 5, продемонстрировали, что применение One-Hot кодировки и квантильной дискретизации к исходному признаковому пространству не повышает качества аутентификации. Более того, градиентый бустинг показывал низкую значимость почти всех признаков после применения One-Hot кодировки. С учетом того, что эти преобразования увеличивали размерность признакового пространства, что влияло на скорость обучения моделей, было принято решение отказаться от данных модификаций. В свою очередь стандартизация значений (6) поспособствовала ускорению и увеличению качества работы методов в среднем на 0.05 по метрике ROC-AUC.

Кроме того, перед обучением модели для динамического обнаружения неконтролируемых локальных выбросов мы применяли технологию Local Outlier Factor, подробно рассмотренную в разделе 5.3.4, что позволило увеличить качество работы методов в среднем на 0.02.

Таблица 5. Методы предобработки признакового пространства

Модели	Методы предобработки признакового пространства				ROC-AUC
	One-Hot	Quantile	Normalize	Boosting	
OC-SVM					0.65
	×				0.53 ↓
		×			0.57 ↓
			×		0.70 ↑
				×	0.68 ↑
IF					0.61
	×				0.50 ↓
		×			0.59 ↓
			×		0.67 ↑
				×	0.63 ↑
LOF					0.56
	×				0.47 ↓
		×			0.52 ↓
			×		0.60 ↑
				×	0.56 ↑
EE					0.55
	×				0.51 ↓
		×			0.52 ↓
			×		0.62 ↑
				×	0.60 ↑

Был проведен ряд экспериментов для выявления наилучшего метода из рассмотренных в разделе 5.3 методов классического машинного обучения. Для каждой модели были подобраны гиперпараметры, отвечающие наилучшему результату работы метода.

Таблица 6. Гиперпараметры моделей

Method ML	Hyperparameters
OneClassSVM	kernel = 'rbf' gamma = $(n\_features \cdot X.var())^{-1}$ nu = 0.10
IsolationForest	n_estimators = 10 max_samples = 0.7 contamination = 0.1
LocalOutlierFactor	n_neighbors = 10 algorithm = 'brute' contaminations = 0.1
EllipticEnvelope	support_fraction = 0.9 contaminations = 0.1

Окончательные результаты экспериментов представлены в сводной таблице 7. Лучший результат продемонстрировала предложеннная комбинация следующих методов:

- стандартная нормализация (6) признакового пространства;
- GradientBoostingClassifier (раздел 5.2.3) для выявления значимых признаков;
- LocalOutlierFactor (раздел 5.3.4) для удаления локальных выбросов;
- OneClassSVM (раздел 5.3.2) в качестве классификатора.

Таблица 7. Результаты экспериментальных исследований

Method ML	Dataset	FRR	FAR	ROC-AUC
IsolationForest	BALABIT	0.14	0.18	0.68
	DATAIIT	0.22	0.23	0.55
	TWOS	0.15	0.21	0.64
EllipticEnvelope	BALABIT	0.23	0.15	0.62
	DATAIIT	0.15	0.32	0.53
	TWOS	0.17	0.18	0.65
OneClassSVM	BALABIT	0.14	0.15	0.71
	DATAIIT	0.09	0.11	**0.80**
	TWOS	0.11	0.14	0.75
LocalOutlierFactor	BALABIT	0.23	0.17	0.60
	DATAIIT	0.24	0.20	0.56
	TWOS	0.18	0.23	0.59

7 Планы на будущее

Нейронные сети активно применяются в сфере безопасности. Они работают быстрее и показывают результат лучше классических методов машинного обучения. Поэтому дальнейшие исследования будут направлены на поиск решения поставленной задачи с применением нейронных сетей.

Решение о легитимности пользователя предложено принимать на основе сравнения с пороговым значением результата работы функции оценки предсказания модели по заданному биометрическому набору данных с векторами признаков из тренировочного набора данных.

Исследования в этом направление уже ведутся.

7.1 Автокодировщик

Автокодировщик (autoencoder) [38] – это нейронная сеть, которая восстанавливает входной сигнал на выходе. Автокодировщик состоит из двух частей: кодировщика, который кодирует данные в свое внутреннее представление, и декодировщика, который восстанавливает исходный вектор. Обычно автокодировщики ограничивают в размерности скрытых слоев (они меньше, чем размерность сигнала на входе) и используют l1-, l2-регуляризаторы для штрафов за активации в скрытых слоях.

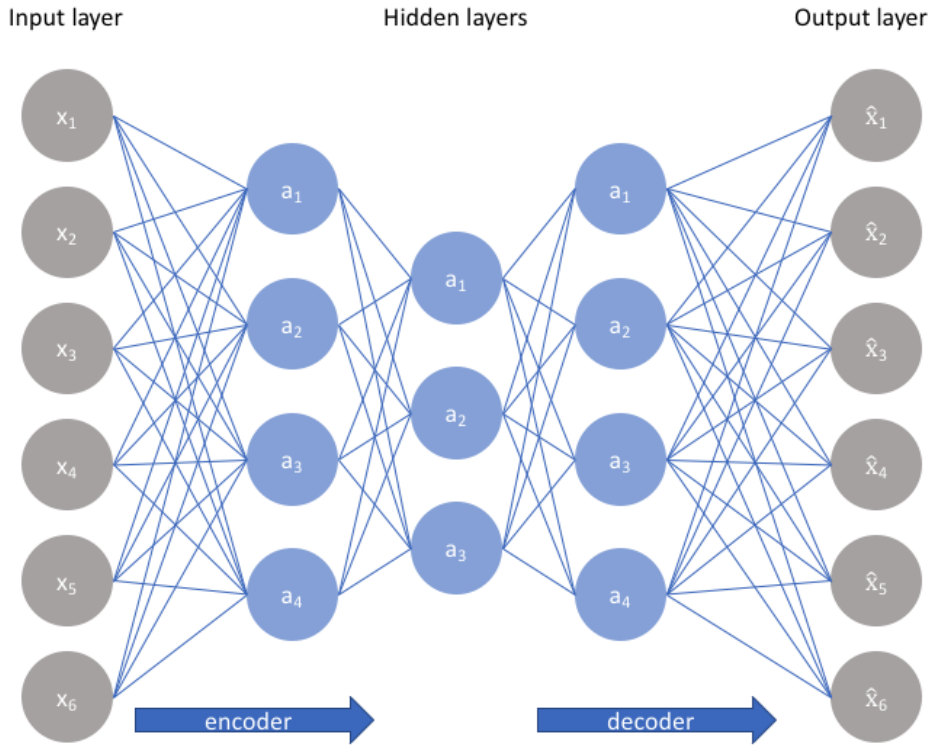


Рисунок 23. Архитектура сети автокодировщика

7.2 Рекуррентные нейронные сети

Одним из преимуществ рекуррентных нейросетей (RNN: Recurrent Neural Networks) [39] является тот факт, что они потенциально умеют связывать информацию с предыдущих входных значений с текущим вектором признаков. Рекуррентные нейронные сети содержат обратные связи.

Долгая краткосрочная память (LSTM: Long Short-Term Memory) [40] – особый вид архитектуры рекуррентных нейронных сетей, способный к обучению долговременных связей.

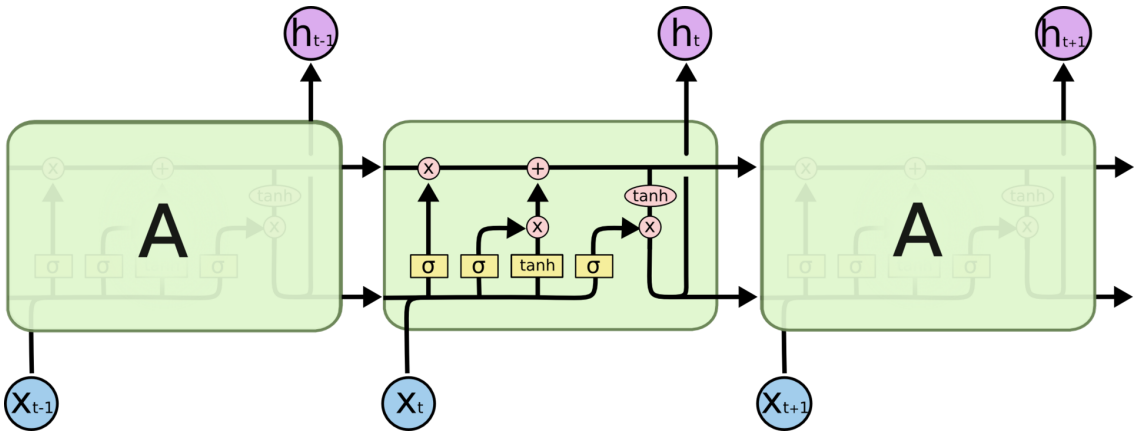


Рисунок 24. Структура нейрона с LSTM

7.3 Полносверточные нейронные сети

Основной особенностью полносверточных нейросетей [41] (FCNN: Fully Convolutional Neural Network) является относительно небольшое число параметров за счет использования сверточных слоев, что позволяет конструировать и обучать более сложные архитектуры для повышения качества решения поставленной задачи.

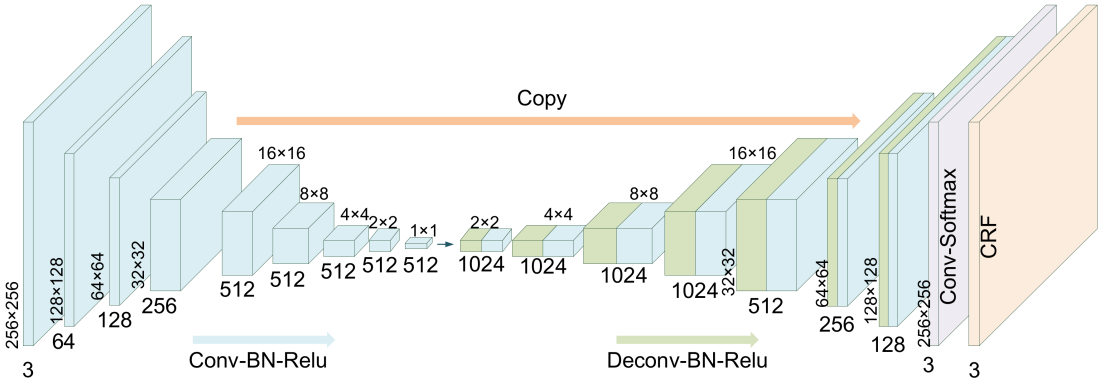


Рисунок 25. Архитектура полносверточного автокодировщика

8 Заключение

В рамках данной работы выполнены все поставленные задачи:

- рассмотрены существующие методы решения задачи динамической аутентификации пользователя на основе анализа его работы с компьютерной мышью с использованием классических методов машинного обучения;
- рассмотрены способы построения признаковых пространств, описывающих динамику работы пользователя с компьютерной мышью, и методы предобработки получившихся вектор-признаков;
- предложены модификации алгоритмов для повышения качества работы методов;
- разработан компонентный экспериментальный стенд и проведены исследования для подтверждения гипотезы об улучшении качества работы методов с предложенными модификациями.

Список литературы

[1] Jain A. K., Ross A., Prabhakar S. An introduction to biometric recognition //IEEE Transactions on circuits and systems for video technology. – 2004. – Т. 14. – №. 1. – С. 4-20.

[2] Wayman J. et al. An introduction to biometric authentication systems //Biometric Systems. – Springer, London, 2005. – С. 1-20.

[3] John D. Cook Biometric security and hypothesis testing (https://www.johndcook.com/blog/2018/10/31/biometric-security-error/)

[4] Fawcett T. An introduction to ROC analysis //Pattern recognition letters. – 2006. – Т. 27. – №. 8. – С. 861-874.

[5] Mondal S., Bours P. A study on continuous authentication using a combination of keystroke and mouse biometrics //Neurocomputing. – 2017. – Т. 230. – С. 1-22.

[6] Mondal S., Bours P. Combining keystroke and mouse dynamics for continuous user authentication and identification //2016 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA). – IEEE, 2016. – С. 1-8

[7] Mondal S., Bours P. A computational approach to the continuous authentication biometric system //Information Sciences. – 2015. – Т. 304. – С. 28-53.

[8] Stokes R. et al. Comparison of Biometric Authentication Software Techniques: GEFE vs. Angle Based Metrics //MAICS. – 2016. – С. 75-89.

[9] Antal M., Egyed-Zsigmond E. Intrusion detection using mouse dynamics //IET Biometrics. – 2019.

[10] Fridman L. et al. Multi-modal decision fusion for continuous authentication //Computers Electrical Engineering. – 2015. – Т. 41. – С. 142-156.

[11] Khalifa A. A. et al. Comparison between mixed binary classification and voting technique for active user authentication using mouse dynamics //2015 International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE). – IEEE, 2015. – С. 281-286.

[12] El Masri A. et al. Active authentication using scrolling behaviors //2015 6th International Conference on Information and Communication Systems (ICICS). – IEEE, 2015. – С. 257-262.

[13] Борисов Р. В. и др. Оценка идентификационных возможностей особенностей работы пользователя с компьютерной мышью //Вестник Сибирской государственной автомобильно-дорожной академии. – 2015. – №. 5 (45).

[14] Kasprowski P., Harezlak K. Biometric Identification Using Gaze and Mouse Dynamics During Game Playing //International Conference: Beyond Databases, Architectures and Structures. – Springer, Cham, 2018. – С. 494-504.

[15] Tan Y. X. M., Binder A., Roy A. Insights from curve fitting models in mouse dynamics authentication systems //2017 IEEE Conference on Application, Information and Network Security (AINS). – IEEE, 2017. – С. 42-47.

[16] Shen C. et al. Pattern-growth based mining mouse-interaction behavior for an active user authentication system //IEEE Transactions on Dependable and Secure Computing. – 2017.

[17] Pilankar P. S., Padiya P. Multi-phase mouse dynamics authentication system using behavioural biometrics //2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES). – IEEE, 2016. – С. 1947-1950.

[18] Chong P., Elovici Y., Binder A. User Authentication Based on Mouse Dynamics Using Deep Neural Networks: A Comprehensive Study //IEEE Transactions on Information Forensics and Security. – 2019.

[19] Chong P. et al. Mouse Authentication without the Temporal Aspect–What does a 2D-CNN learn? //2018 IEEE Security and Privacy Workshops (SPW). – IEEE, 2018. – С. 15-21.

[20] Monaro M., Gamberini L., Sartori G. The detection of faked identity using unexpected questions and mouse dynamics //PloS one. – 2017. – Т. 12. – №. 5. – С. e0177851.

[21] Feher C. et al. User identity verification via mouse dynamics //Information Sciences. – 2012. – Т. 201. – С. 19-36.

[22] Hall M. et al. The WEKA data mining software: an update //ACM SIGKDD explorations newsletter. – 2009. – Т. 11. – №. 1. – С. 10-18.

[23] Breiman L. Random forests //Machine learning. – 2001. – Т. 45. – №. 1. – С. 5-32.

[24] Breiman L. Bagging predictors //Machine learning. – 1996. – Т. 24. – №. 2. – С. 123-140.

[25] Szegedy C. et al. Going deeper with convolutions //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2015. – С. 1-9.

[26] Fülöp, Á., Kovács, L., Kurics, T., Windhager-Pokol, E. (2016). Balabit Mouse Dynamics Challenge data set. Available at: https://github.com/balabit/Mouse-Dynamics-Challenge

[27] Harilal A. et al. Twos: A dataset of malicious insider threat behavior based on a gamified competition //Proceedings of the 2017 International Workshop on Managing Insider Security Threats. – 2017. – С. 45-56.

[28] Friedman J. H. Greedy function approximation: a gradient boosting machine //Annals of statistics. – 2001. – С. 1189-1232.

[29] Binning [HTML] (https://wiki.loginom.ru/articles/binning.html)

[30] Kazachuk M. et al. One-class models for continuous authentication based on keystroke dynamics //International Conference on Intelligent Data Engineering and Automated Learning. – Springer, Cham, 2016. – С. 416-425.

[31] Дьяконов А.Г. Поиск аномалий (Anomaly Detection) (https://dyakonov.org/2017/04/19/поиск-аномалий-anomaly-detection/)

[32] Pimentel M. A. F. et al. A review of novelty detection //Signal Processing. – 2014. – Т. 99. – С. 215-249.

[33] Cortes C., Vapnik V. Support-vector networks //Machine learning. – 1995. – Т. 20. – №. 3. – С. 273-297.

[34] Manevitz L. M., Yousef M. One-class SVMs for document classification //Journal of machine Learning research. – 2001. – Т. 2. – №. Dec. – С. 139-154.

[35] Liu F. T., Ting K. M., Zhou Z. H. Isolation forest //2008 Eighth IEEE International Conference on Data Mining. – IEEE, 2008. – С. 413-422.

[36] Breunig M. M. et al. LOF: identifying density-based local outliers //Proceedings of the 2000 ACM SIGMOD international conference on Management of data. – 2000. – С. 93-104.

[37] Hoyle B. et al. Anomaly detection for machine learning redshifts applied to SDSS galaxies //Monthly Notices of the Royal Astronomical Society. – 2015. – Т. 452. – №. 4. – С. 4183-4194.

[38] Ng A. et al. Sparse autoencoder //CS294A Lecture notes. – 2011. – Т. 72. – №. 2011. – С. 1-19.

[39] Rumelhart D. E., Hinton G. E., Williams R. J. Learning internal representations by error propagation. – California Univ San Diego La Jolla Inst for Cognitive Science, 1985. – №. ICS-8506.

[40] Hochreiter S., Schmidhuber J. Long short-term memory //Neural computation. – 1997. – Т. 9. – №. 8. – С. 1735-1780.

[41] Long J., Shelhamer E., Darrell T. Fully convolutional networks for semantic segmentation //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2015. – С. 3431-3440.