ОБНАРУЖЕНИЕ РЕДКОЙ АНОМАЛЬНОЙ АКТИВНОСТИ ПОЛЬЗОВАТЕЛЕЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ МЕТОДАМИ МАШИННОГО ОБУЧЕНИЯ

DOI: 10.36724/2072-8735-2026-20-5-29-38

Manuscript received 28 February 2026;
Accepted 30 April 2026

Шелухин Олег Иванович,
Московский технический университет связи и информатики, Москва, Россия,
sheluhin@mail.ruu

Осин Андрей Владимирович,
Московский технический университет связи и информатики, Москва, Россия,
a.v.osin@mtuci.ru

Ключевые слова: *информационные системы, инсайдер, аномальная активность, поведенческий профиль, кластерное соседство*

Цель исследования: повышение эффективности противодействия инсайдерам в информационных системах за счет построения адекватной модели аномального поведенческого профиля пользователей CRM-систем методами машинного обучения. На основе предложенного подхода идентифицируются потенциальные инсайдеры. Методы исследования: аналитический обзор релевантных научных публикаций в области обнаружения аномальной активности пользователей при работе с информационными системами; поведенческий анализ в сочетании с алгоритмами машинного и глубокого обучения, открывающий новые возможности для раннего выявления инсайдерских угроз. Полученные результаты: Разработан обобщенный алгоритм обнаружения редкого аномального поведения пользователя информационных систем. Показано, как методами машинного обучения может быть сформирована эталонная база кластерных образцов аномальных объектов. Для обнаружения аномальной активности пользователей предложено сравнивать кластерное представление нового объекта с эталонными образцами по метрике сходства, что позволяет оценить, насколько поведение пользователя похоже на ранее зафиксированные аномалии. Научная новизна работы определяется авторским подходом к противодействию редкой аномальной активности пользователей информационных систем на основе сравнения типовых нормальных поведенческих профилей и поведенческих профилей инсайдеров с использованием методов интеллектуального анализа данных.

Информация об авторах:

Шелухин Олег Иванович, *Московский технический университет связи и информатики, д.т.н., профессор, заведующий кафедрой "Информационная безопасность", Москва, Россия. ORCID: https://orcid.org/0000-0001-7564-6744*

Осин Андрей Владимирович, *Московский технический университет связи и информатики, к.т.н., доцент, кафедра "Информационная безопасность", Москва, Россия. ORCID: https://orcid.org/0000-0002-6384-9365*

Для цитирования:
Шелухин О.И., Осин А.В. Обнаружение редкой аномальной активности пользователей информационной системы методами машинного обучения // T-Comm: Телекоммуникации и транспорт. 2026. Том 20. №5. С. 29-38.

For citation:
O. I. Sheluhin, A.V. Osin, "Detecting Rare User Behavior Anomalies in Information Systems with Machine Learning," *T-Comm*, 2026, vol. 20, no.5, pp. 29-38. (*in Russian*)

Введение

Вопрос выявления аномальной активности пользователей в информационных системах приобретает всё большую актуальность в условиях роста внутренних и внешних угроз информационной безопасности [1, 2, 17]. Под аномальной активностью понимаются действия, отклоняющиеся от типичных временных, географических и функциональных характеристик поведения конкретного пользователя [17]. Выявление таких отклонений опирается на построение поведенческого профиля, позволяющего различать нормальную и подозрительную активность [17, 19].

Одной из наиболее опасных форм аномального поведения является инсайдерская угроза, связанная с действиями сотрудников, подрядчиков или иных лиц, имеющих легитимный доступ к ресурсам организации, а также с атаками, имитирующими поведение доверенного пользователя [3-5].

Сложность обнаружения инсайдеров обусловлена тем, что их активность нередко осуществляется в рамках формально допустимых полномочий и с учётом знания внутренних процессов. В работе [5] выделяются невольные нарушители, злонамеренные субъекты, недовольные сотрудники и внешние специалисты, получившие доступ к внутренней инфраструктуре.

К числу признаков аномального поведения относятся входы в нерабочее время, множественные неудачные попытки аутентификации, обращения к нетипичным ресурсам, всплески сетевой активности, использование нестандартных устройств, IP-адресов или географических точек доступа, подозрительные операции с файлами, а также попытки обхода средств защиты [15-17]. Подобные отклонения особенно значимы в системах, содержащих критически важные и чувствительные данные.

Особое место среди таких систем занимают CRM-платформы, предназначенные для хранения клиентских данных, автоматизации взаимодействия с клиентами и поддержки бизнес-процессов. Высокая концентрация коммерчески значимой информации делает CRM-системы привлекательной целью как для внешних злоумышленников, так и для инсайдеров.

Внешние атаки обычно требуют значительных ресурсов, эксплуатации уязвимостей, компрометации учетных данных или применения фишинга и вредоносного ПО, а также сопровождаются заметным риском обнаружения [9].

В отличие от внешнего нарушителя, инсайдер обладает не только санкционированным доступом к CRM-системе, но и пониманием её логики, бизнес-контекста и организационных слабых мест [10, 11]. Это делает его действия менее заметными и существенно осложняет своевременное выявление угроз. В связи с этим традиционные средства периметральной защиты оказываются недостаточными, а приоритет смещается в сторону мониторинга поведения пользователей и обнаружения нетипичных, но формально допустимых действий [6-8, 17, 19]. Следовательно, разработка методов выявления аномальной активности пользователей в CRM-системах представляет собой актуальную задачу обеспечения информационной безопасности.

Целью статьи является построение адекватной модели аномального поведенческого профиля пользователей CRM-систем, на основе которой методами машинного обучения идентифицируются потенциальные инсайдеры [18-20].

1 Аномальное поведение пользователя

1.1 Постановка задачи

Функционал CRM-системы очень широк. Обычно, в рамках ролей, выданных пользователем, известно, каким функционалом пользуется представитель данной роли. Если внезапно этот функционал резко меняется, а пользователь выполняет неожиданные действия, которые ранее он не делал, то, подобные действия также можно идентифицировать как аномальное поведение и делать из этого соответствующие выводы [17].

При работе с пользователем системы нужно обращать внимание на предупреждение системы безопасности, которая в том или ином виде всегда встроена в информационную систему. Например, когда несколько раз фиксируются попытки ввода неправильного пароля, или когда пользователь включил VPN для того, чтобы подключиться к этой информационной системе [6]. В случае использования такого-то вредоносного кода пользователь может попытаться повысить свои привилегии до администратора и получить те права, на которые он изначально не был подписан.

Рассмотренные проявления позволяют отнести поведение пользователя к категории потенциально опасного, поскольку именно такие отклонения нередко предшествуют утечкам информации и реализации инсайдерских сценариев [1, 11]. При анализе внутренних угроз существенное значение имеют не только отдельные подозрительные действия в информационных системах, но и общая динамика изменений в поведении сотрудника [17].

На практике о повышенном риске инсайдерской активности могут свидетельствовать две группы признаков. Первая связана с цифровым следом пользователя и включает появление нетипичных действий в корпоративных информационных системах [17]. Вторая относится к изменению поведенческого состояния сотрудника, которое может проявляться в снижении эффективности работы, ослаблении вовлечённости в рабочие процессы или изменении привычного характера взаимодействия с системой.

Основанием для вывода об аномальном поведении может служить фиксация активности, нехарактерной для конкретного пользователя в обычных условиях [17]. К числу таких проявлений относятся резкие отклонения от типового сценария работы, необычно высокий уровень потребления ресурсов, а также изменение шаблона взаимодействия с системой. Последнее может выражаться в переходе к новым разделам, вкладкам или функциям, которые ранее не были свойственны данному пользователю. В совокупности такие признаки позволяют рассматривать сотрудника как потенциальный объект дополнительного мониторинга и поведенческого анализа [17, 19].

Отдельной категорией аномального поведения являются резкие скачки поведения пользователя, когда фиксируется новые паттерны в его поведении. Например, когда объем информации, связанный с пользователем, составлял ранее несколько мегабайт в час, а в текущем времени зафиксирован скачок, например, до 100 мегабайт в час [15, 16]. Скачки такого рода также являются признаками или типом аномального поведения. Анализ перечисленных аномальных событий позволяет перейти к оценке того, кто является их источником (актором).

Это могут быть либо инсайдеры-сотрудники, которые допущены во внутренний периметр безопасности информационной системы, либо хакеры, которые пытаются вторгнуться извне, используя методики и социальной инженерии, либо атаки, или иные способы (вирусы, черви и так далее). Это могут быть внешние манипуляторы - люди, которые пытаются своими связями, знакомствами проникнуть внутрь периметра безопасности информационной системы и вынудить людей находящихся внутри на выполнение противоправных действий.

В рамках рассмотрения систем, занимающихся аномальным поведением пользователей, известны несколько типов подобных систем [1, 14]. В частности, это UBA (User Behavior Analytics), технология в информационной безопасности, которая анализирует поведение пользователей и выявляет необычные или подозрительные действия (аномалии), которые могут указывать на угрозы [17]. Продвинутой версией UBA является UEBA (User and Entity Behavior Analytics). Данный класс систем кроме поведения пользователей рассматривает и поведение отдельных узлов системы [17].

Ключевой задачей систем выявления аномального поведения является формирование поведенческой модели пользователя, позволяющей оценивать степень отклонения текущей активности от ожидаемой [17, 19]. При обнаружении существенных отклонений важно не только зафиксировать аномалию, но и сопоставить её с ранее выявленными сценариями аномального поведения для более точной идентификации потенциального нарушителя и классификации угрозы [19, 20]. Таким образом, поведенческий профиль должен служить основой как для описания нормальной активности, так и для выделения характерных признаков поведения инсайдера.

2 Обобщённый алгоритм обнаружения аномального поведения пользователя

Обобщенная структура алгоритма обнаружения аномального поведения пользователя [13] может быть представлена в виде последовательности из пяти этапов, представленных на рис. 1.

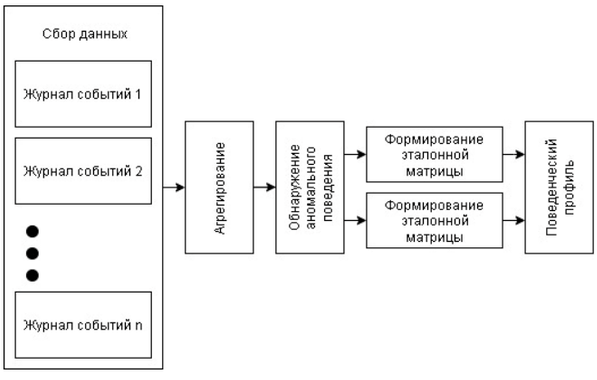

Рис. 1. Структура алгоритма обнаружения аномального поведения пользователя

Этап 1. Сбор данных. Журнал событий

На первом этапе система отслеживает события, происходящие в результате работы пользователя информационной системы. Среди этих событий можно выделить следующие:

- сколько времени пользователь проводит на определенной вкладке информационной системы;
- что пользователь копирует в буфер обмена;
- есть ли в том, что скопировал пользователь в буфер обмена, персональные данные;
- как часто изменяется клавиатурный почерк пользователя («мышиный почерк»).

Эти типы поведения могут быть ределены на основе следующих показателей:

- частота использования клавиатуры;
- как быстро он двигает курсор мыши при работе в различных разделах информационной системе;
- какое ускорение этого курсора, как быстро и нервно он осуществляет скроллинг (от англ. scrolling—«прокрутка») действие, которым пользователь перемещает содержимое на экране своего гаджета, чтобы увидеть скрытую за границами дисплея часть.

В результате анализа подобных событий формируется вектор, содержащий все элементы, характеризующие различные аспекты поведения пользователя при взаимодействии с информационной системой. Несложно предположить, если пользователь выполняет свои рутинные функции, он достаточно спокоен, выполняет их изо дня в день и этот шаблон его поведения повторяется. Однако. как только у пользователя появляется намерение выполнить какое-то негативное действие (например украсть что-то), то, скорее всего, в каком-то одном или в нескольких из этих показателей это проявится.

Это, например, может проявиться в виде какого-то нервного движения мыши, или пользователь перестанет заходить на какую-то вкладку, а будет больше внимания уделять тем разделам системы, где хранятся действительно ценные данные или среди них можно выделить ряд категорий данных, таких как: аккумуляторный почерк; «мышиный почерк»; анализ буфера обмена; пользовательская активность; нахождение пользователя в разделах информационной системы (открытие пользователем определенных вкладок в определенные интервалы времени).

Фиксация действий пользователя в сервисной системе требует использования унифицированного формата представления данных, пригодного для последующей автоматизированной обработки. Одним из наиболее распространённых решений является формат JSON (JavaScript Object Notation), предназначенный для хранения и передачи структурированной информации. Несмотря на связь с синтаксисом JavaScript, данный формат является платформенно независимым и поддерживается широким кругом языков программирования. Представленные в нём записи содержат набор характеристик, описывающих действия пользователя и позволяющих формировать признаки для анализа его поведения. Среди них есть в том числе и параметры курсора мыши при работе.

Из множества показателей можно увидеть, какова история изменения почерка, каково содержимое курсора обмена, какова активность пользователя на определенной вкладке, какой процент времени он там провел и т.д. Это могут быть в том числе данные о том на какие кнопки пользователь нажимал, какие выгружал файлы и т.д. В результате, в реальном времени создается набор JSON-логов, который фиксируется и анализируется с целью формирования информации необходимой для построения профиля функционирования каждого пользователя.

Рассмотрим набор данных, получаемых из CRM-системы через сниффер, который представляет собой записи пяти логов данных пользователя в формате json-файла. Для удобства представления данных json-файлы логов могут быть открыты в браузере. Поле «path» является вкладкой (страницей), на которой находится пользователь. В конце каждого файла присутствует поле «user_id», означающее идентификатор пользователя в CRM-системе.

Записи в каждом из пяти логов содержат поля:

- crm_user_club_id – идентификатор клуба пользователя в системе;
- crm_user_fio – фамилия, имя и отчество пользователя в системе;
- crm_user_role – роль пользователя в системе;
- data – массив данных для каждой записи.

Признаки (атрибуты) действий пользователя с мышью: $\{a_{\text{ДМ.}i}, i = \overline{1,I}, I = 11\}$

- metric_id – идентификатор записи действия пользователя с мышью;
- path - URL-адрес или идентификатор вкладки;
- created_at – время записи действия пользователя с мышью;
- start_at – время начала действия пользователя с мышью;
- end_at – время окончания действия пользователя с мышью;
- left_clicks – количество левых кликов пользователя на вкладке;
- right_clicks – количество правых кликов пользователя на вкладке;
- scroll_up-количество прокруток пользователем вверх;
- scroll_down – количество прокруток пользователем вниз;average_acceleration – среднее ускорение движения курсора (пиксель/миллисекунду);
- average_speed – средняя скорость движения курсора (пиксель/секунду).

Признаки (атрибуты) действий пользователя с клавиатурой: $\{a_{\text{ДК.}j}, j = \overline{1,J}, J = 7\}$

- metric_id – идентификатор записи действия пользователя с клавиатурой;
- path – URL-адрес или идентификатор вкладки;
- created_at – время записи действия пользователя с клавиатурой;
- start_at – время начала действий пользователя с клавиатурой в секундах;
- end_at – время окончания действий пользователя с клавиатурой в секундах;
- average_speed – средняя скорость нажатия на клавиши (пиксель/миллисекунду);
- average_press_time – среднее время нажатия на клавиши в миллисекундах.

Признаки (атрибуты) пользовательской активности: $\{a_{\text{Да.}k}, k = \overline{1,K}, K = 7\}$

- metric_id – идентификатор записи пользовательской активности;
- path – URL-адрес или идентификатор вкладки;
- created_at – время записи;
- пользовательской активности в секундах;
- start_at – время начала активности пользователя в секундах;
- end_at – время окончания активности пользователя в секундах;
- active – время активности пользователя на вкладке в миллисекундах;
- inactive – время неактивности пользователя на вкладке в миллисекундах.

Признаки (атрибуты) действий пользователя с буфером обмена: $\{a_{\text{ДВ.}l}, l = \overline{1,L}, L = 5\}$

- metric_id – идентификатор записи действия пользователя с буфером обмена;
- path – URL-адрес или идентификатор вкладки;
- created_at – время записи действия пользователя с буфером обмена в секундах;
- copied_at – время копирования данных пользователем в буфер обмена в секундах;
- data (вложенный словарь) – это может быть, например, сумма продаж за определенный период.

Признаки (атрибуты) событий пользователя на вкладке: $\{a_{\text{ДВ.}m}, m = \overline{1,M}, M = 5\}$

- metric_id – идентификатор записи события пользователя на вкладке;
- path – URL-адрес или идентификатор вкладки;
- created_at – время записи события пользователя на вкладке в секундах;
- event_at – метка времени события в секундах;
- type – тип события пользователя на вкладке.

При анализе можно такое структуры легко выявить случаи появления в буфере обмена персональных данных пользователя. Общий вектор атрибутов имеет вид

$$A = \{a_{\text{ДМ.}i}, a_{\text{ДК.}j}, a_{\text{Да.}k}, a_{\text{ДБ.}l}, a_{\text{ДВ.}m}, i = \overline{1,I}, j = \overline{1,J}, k = \overline{1,K}, l = \overline{1,L}, m = \overline{1,M}\}.$$

Этап 2. Агрегирование по интервалам

На втором этапе решается задача преобразования полученного достаточно разнородного набора данных к некоторым стандартизованным агрегированным интервалам. Агрегация данных является важным этапом в работе системы и выполняется следующим образом.

Для каждой вкладки, перечисленные логи агрегируются в заданном временном интервале. Для определенности будем агрегировать полученные данные (логи событий в системе) по 5-минутным интервалам, как это показано на рис. 2.

Пример представления агрегированных данных метрики агрегированной активности пользователя на вкладке CRM-системы имеет вид.

- active_percentage – процент активности = (сумма времени активности / (сумма времени активности + сумма времени не активости)) – пользователя;
- average_active – средняя активность пользователя в минутах = (процент активности / количество открытий пути пользователем)
- opening_amount – количество открытий пользователем вкладки;
- aggregation_id – номер записи агрегации;
- day_time – временной интервал суток (ночь, день, вечер) на основе времени начала активности (час дня);
- path – URL – адрес или идентификатор вкладки;

• created_at – время записи активности пользователя в секундах;

• start_at – время начала активности пользователя в секундах;

• end_at – время окончания активности пользователя в секундах.

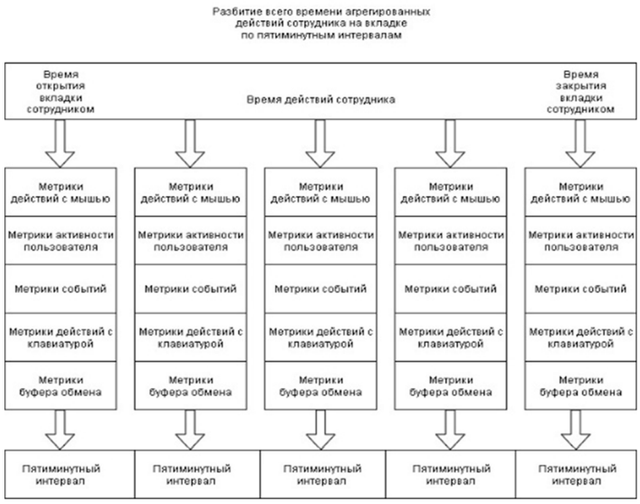

Рис. 2. Схема агрегирования атрибутов по пятиминутным интервалам

Аналогично находятся атрибуты для агрегированных действий с мышью пользователя на вкладке; агрегированных событий пользователя на вкладке; агрегированных данных действий пользователя с клавиатурой на вкладке; агрегированных данных буфера обмена пользователя на вкладке.

При нахождении нескольких типов событий в одном пятиминутном интервале для каждого типа атрибута создается отдельный столбец. Затем, в рамках каждого агрегированного 5-минутного интервала формируется вектор, $A = \{a_1, a_2, \dots a_n, n = \overline{1,N}, N = I + J + K + L + M\}$ содержащий элементы, соответствующие введенным показателям. В результате агрегирования формируется структурированная база данных, которая хранится в электронном виде на компьютере или специальном сервере.

Этап. 3 Обнаружение аномального поведения

На третьем этапе с помощью искусственных нейронных сетей (ИНС) осуществляется обнаружение аномального поведения пользователей. Вариант структуры ИНС в виде автокодировщика (АК) иллюстрируется на рисунке 3.

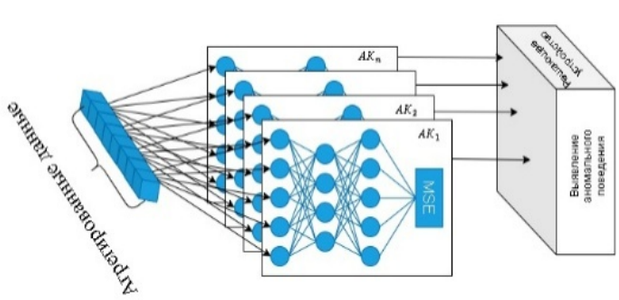

Рис. 3. Структура обработки данных каждой вкладки с помощью АК

Как известно [14] автокодировщик представляет собой композицию двух функций: $\mathrm{AE}(x) = Decoder\big(Encoder(x)\big) = \hat{x}$, где $x \in \mathbb{R}^N$ – выходной нормализованный вектор; $\hat{x} \in \mathbb{R}^N$ – реконструированный выход. Encoder (кодер) сжимает данный до скрытого пространства, а Decoder восстанавливает данные из сжатого представления.

АК обучается на основании минимизации ошибки реконструкции и для некоторых пятиминутных интервалов считает их аномальными соответствующими аномальным поведениям пользователей. Количество АК соответствует числу вкладок действий пользователя CRM-системы. Для выявления аномального поведения пользователя для каждого пятиминутного интервала для каждой вкладки вычисляется ошибка реконструкции. Архитектура кодера описывается функцией отображения $f_{enc}: \mathbb{R}^N \to \mathbb{R}^d$, где $d < N$ и может быть описана как последовательность линейных преобразований и нелинейностей вида

$$z = f_{enc}(x) = \sigma^{(L)}\left(W^{(L)} \cdot \dots \sigma^{(1)}\left(W^{(1)}x + b^{(1)}\right) + \dots + b^{(L)}\right) \quad (1)$$

где $W^{(l)}, b^{(l)}$ – веса и смещения на уровне l; $\sigma^{(l)}$ – функция активации (обычно ReLU, than, sigmoid); $z \in \mathbb{R}^d$ – сжатое (бутылочное) представление.

Архитектура декодера описывается функцией восстановления $f_{dec}: \mathbb{R}^d \to \mathbb{R}^N$ и строится аналогично (1) в виде последовательности преобразований:

$$\hat{x} = f_{\text{dec}}(z) = \phi^{(M)}\left(V^{(M)} \cdot \dots \phi^{(1)}\left(V^{(1)}z + c^{(1)}\right) + \dots + c^{(M)}\right) \quad (2)$$

где $V^{(m)}, c^{(m)}$ – веса и смещения на уровне слоя m соответственно; $\phi^{(m)}$ – функция активации (обычно симметрична $\sigma^{(l)}$); $\hat{x} \in \mathbb{R}^N$ – восстановленный вектор. Данные, сжатые в «узком слое», восстанавливаются в выходном слое.

На заключительном этапе вычисляется функция потерь, которая оценивается как среднеквадратичная ошибка (MSE-Mean Squared Error.) реконструкции для тренировочных данных для обновления весов модели. Для тестовых данных ошибка реконструкции вычисляется путем усреднения по всему подмножеству данных $\mathcal{L}(x, \hat{x}) = \frac{1}{N}\sum_{j=1}^{N}\left(x_j - \hat{x}_j\right)^2$.

Без анализа реальных событий сложно судить о том на сколько эти аномалии относятся или не относятся к инсайдерским действиям. Для этого нужна эталонная матрица, задающая срез поведения пользователя, когда АК, обученный на заданном интервале, признал эти участки нормальными. В эталонной матрице содержатся примеры нормального и аномального поведения, которые формирует АК.

Этап.4. Формирование эталонной матрицы

Предлагается предварительно разделить поведенческие временные интервалы на «типовые», «промежуточные» и «резко отличающиеся» от тех, которые в рамках эталонной матрицы признаны аномальными (нетипичными). Для такого разделения необходимо автоматическое присвоение меток на основе анализа ошибок реконструкции АК. Процесс обучения начинается с момента появления у пользователя нового

типа активности, например, при первом открытии новой вкладки. С этого момента начинается накопление пятиминутных интервалов до достижения необходимого объёма данных, пригодного для дальнейшего обучения.

Пусть входные данные описываются матрицей признаков $D \in \mathbb{R}^{T\times N}$, где: T – количество объектов, N – количество признаков. Требуется автоматически присвоить каждому объекту одну из трёх меток: 0, 1 или 2, отражающих уровень «сходство с эталоном».

С этой целью на первом шаге для стабилизации диапазонов изменения признаков и подготовки данных к обучению АК для каждого признака $j \in \{1,\ldots,N\}$ и каждого объекта данные $i \in \{1,\ldots,T\}$ нормализуются **с** помощью преобразования вида

$$x_{i,j}^{\text{norm}} = \frac{x_{i,j} - \min\left(x_{\cdot,j}\right)}{\max\left(x_{\cdot,j}\right) - \min\left(x_{\cdot,j}\right)},$$

где $x_{\cdot,j}$ – столбец всех значений признака j; $x_{i,j}^{norm} \in [0,1]$.

На втором шаге, в режиме обучения АК, восстанавливаются нормальные/нетипичные объекты, так чтобы аномальные давали высокую ошибку реконструкции путем вычисления функция потерь $\mathcal{L}(x,\hat{x}) = \frac{1}{N}\sum_{j=1}^{N}\left(x_j^{\text{norm}} - \hat{x}_j^{\text{norm}}\right)^2$. В качестве обучающей выборки в режиме обучения используется сбалансированный набор, содержащий равные доли объектов разных типов (если это возможно).

На третьем шаге осуществляется **п**рогон объектов через АК и для каждого объекта x_i вычисляется текущая ошибка реконструкции $\varepsilon_i = \frac{1}{N}\sum_{j=1}^{N}\left(x_{i,j}^{\text{norm}} - \hat{x}_{i,j}^{\text{norm}}\right)^2$.

На четвертом шаге с целью адаптации границ присвоения меток на основе распределения ошибок реконструкции $\vec{\varepsilon} = \{\varepsilon_1,\ldots,\varepsilon_T\}$ определяются пороговые уровни по процентилям: $\varepsilon_{20} = Percentile_{20}(\vec{\varepsilon})$ и и $\varepsilon_{50} = Percentile_{50}(\vec{\varepsilon})$. Здесь $\varepsilon_{20} = 20$й перцентиль $\{\varepsilon_i\}, \varepsilon_{50} = 50$й перцентиль $\{\varepsilon_i\}$.

По величине ошибки реконструкции в соответствии с правилом $y_i = \begin{cases} 0, & \varepsilon \le \varepsilon_{20} \\ 1, & \varepsilon_{20} < \varepsilon \le \varepsilon_{50} \\ 2, & \varepsilon_i > \varepsilon_{50} \end{cases}$ осуществляется присвоение меток.

Метки присваиваются следующим образом:

- Если ошибка реконструкции $\varepsilon \le 20\%$ →присваивается метка 0;
- Если ошибка реконструкции $\varepsilon \in (20\%, 50\%]$ → присваивается метка 1;
- Если ошибка реконструкции $\varepsilon > 50\%$ → присваивается метка 2.

На пятом заключительном шаге с целью создания сбалансированной эталонной выборки происходит формирование эталонной матрицы D_{ref}. С этой целью выбирается одинаковое количество объектов с каждой меткой (0, 1, 2) и объединяются в эталонную матрицу $D_{\text{ref}} = D_0 \cup D_1 \cup D_2$, где $D_m = \{x_i | y_i = m\}, m \in \{0,1,2\}$.

В результате каждый объект получает метку, отражающую его уровень «нормальности» относительно обучающего автокодировщика. При дальнейшем анализе с помощью адаптивных порогов (процентильный подход) формируется эталонная матрица. С этой целью признаки предварительно нормализуются, что делает метод универсальным для разных шкал. При работе АК было выбрано три уровня поведения пользователя: полностью нормальное, переходное и аномальное, как это показано на рис. 4.

В качестве примера на Рис. 5 представлена гистограмма ошибок и границы классификации для случая, когда вместо фиксированных порогов используются перцентили: – 20-й перцентиль равный 0.152, и 50-й перцентиль равный 0.280.

Рисунок 5 иллюстрирует процесс разделения групп ошибок реконструкции для автокодировщика. Показаны метки порогов 20% и 50%. Ошибки, не превышающие порог в 20%, соответствуют нормальному поведению пользователя. Область 20-50% характеризует переходное поведение. Ошибки выше 50% соответствуют аномальным выбросам. В случае, когда формируются указанные кластерные соседства интересует тот кластер, в котором оказалась пятиминутная выборка, сгенерированная пользователем в результате его работы. В соответствии с введенной терминологией интересует сколько и какие события оказались в соответствующем кластере: нормальные, переходные, аномальные.

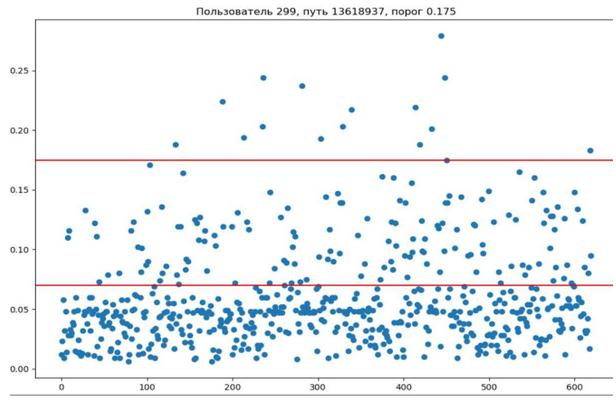

Рис. 4. Трехуровневая кластеризация аномального поведения на выходе АК

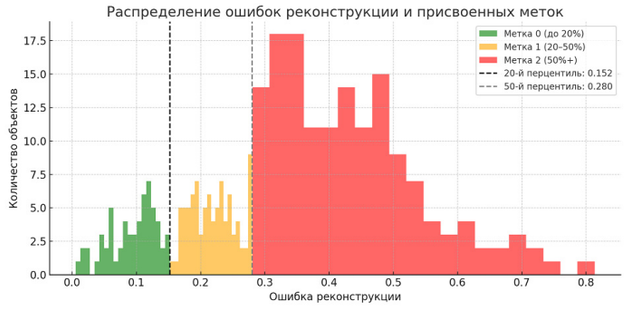

Рис. 5. Гистограмма ошибок и границы классификации

Этап 5. Поведенческий профиль

Для того, чтобы принимать решение при построении поведенческого профиля, используя для этих целей алгоритмы

кластерного соседства, введенные уровни аномальности поведения пользователя должны быть представлены в эталонной матрице. Под «Кластерным соседством» понимается оценка того какие события соседствуют друг другу, т.е. располагаются в одном кластере для заданного пространства атриубутов. Обычно эта процедура осуществляется с помощью кластеризации. Учитывая большую размерность пространства данных, которая определяется количеством атрибутов предлагается осуществлять кластеризацию в следующей последовательности. Процедура анализа кластерного соседства осуществляется в следующей последовательности.

На шаге 1 формируются все возможные комбинации N анализируемых атрибутов в различных комбинациях. Пусть $x \in \mathbb{R}^N$ - нормализованный вектор признаков пятиминутного интервала. Обозначим через К – минимальное количество атрибутов в комбинации (например, от 3 и до 10). Комбинации таких атрибутов легко и быстро вычисляются. В результате формируются бутстрепы (bootstrap) с различными пространствами, формированными из таких комбинаций атрибутов. Суть бутстрэпа заключается в том, что на основе одной исходной выборки формируют несколько новых обучающих выборок, случайным образом выбирая элементы и копируя их в новую выборку (в прежней они остаются, не удаляются).

Затем эта операция повторяется по заданному размеру новой выборки. В результате формируется новая обучающая выборка, состоящая из элементов исходной с некоторыми повторениями. Множество всех возможных комбинаций атрибутов из A, длины от K до $K+M$ может быть описано соотношением $C = \bigcup_{r=K}^{K+M} C_r = \bigcup_{r=K}^{K+M} \{C \subseteq A \| C| = r\}$, где $C_r = \{C \subseteq A \| C| = r\}$ - множество всех комбинаций длины r.

Каждая комбинация С из r атрибутов $(K \le r \le K+M)$ может быть представлена либо в виде множества индексов (например, комбинация $\{a_1, a_3, a_4\}$ представляется как $\{1,3,4\}$), либо как битовая маска длины N – вектор $b \in \{0,1\}^N$, где $b_i = 1$, если атрибут a_i включен в комбинацию, в противном случае (при в= 0) атрибут в комбинацию не включен.

На шаге 2 выполняется кластеризация по каждой комбинации атрибутов. С этой целью для каждого множества C всех комбинаций атрибутов из этапа 1 в полученном пространстве $D \in \mathbb{R}^{T \times N}$производится кластерный анализ. Здесь $D \in \mathbb{R}^{T \times N}$ – исходная таблица данных, состоящая из T строк (объектов) и N столбцов (атрибутов).

Кластерный анализ позволяет упорядочить результаты прогнозирования, выделяя в исходном массиве данных группы объектов со схожими характеристиками. Такая группировка снижает сложность дальнейшего анализа и позволяет работать не с отдельными наблюдениями, а с компактным набором однородных сегментов. В отличие от задач классификации, кластеризация не опирается на заранее заданную целевую переменную: структура данных выявляется на основе выбранного пространства признаков, а число кластеров заранее может быть неизвестно и подбирается в зависимости от цели исследования и качества разбиения. В рассматриваемой постановке набор признаков определяется исходной таблицей, содержащей T строк (объектов) и N столбцов (атрибутов).

Для выбранного алгоритма кластеризации (например, для алгоритма k-means) ставится в соответствие $k \in \{2,3,4,5\}$ кластеров. С этой целью для каждой комбинации атрибутов $C \in \mathcal{C}$, где $C = \{a_{i1}, a_{i2}, \ldots, a_{ir}\}$ определяется подматрица данных $D_C = D[:, C] \in \mathbb{R}^{T \times r}$. Поскольку для алгоритма k-means количество кластеров k задано заранее, то необходимо разделить данные на кластеры таким образом, чтобы минимизировать полную сумму квадратов расстояния от данной точки до центра кластера, к которому она принадлежит.

Для решения задачи кластеризации необходимо построить множество $C = \{c_1, c_2, \ldots, c_g\}$, где c_k – кластер, содержащий похожие друг на друга атрибуты $c_k = \{i_j, i_p \mid i_j \in I, i_p \in I \text{ и } d(i_j, i_p) < \sigma\}$, где $d(i_j, i_p)$ – мера близости между атрибутами ; σ - заданное расстояние между атрибутами для включения их в один кластер. Для каждого подмножества C и параметра кластеризации $k \in \{2,3,4,5\}$ используют итеративный алгоритм $\mathrm{KMeans}_k(D_C) \to \mathrm{ClusterLabels}_{C,k} \in \{1, \ldots, k\}^T$. Объект x_i попадает в кластер $K_{C,x}(x_i)$ и определяется его кластерное окружение $G_{C,k}(x_i) = \left\{ x_j \middle| K_{C,k}(x_j) = K_{C,k}(x_i) \right\}$.

Цель кластеризации состоит в таком разбиении множества объектов, при котором минимизируется суммарная величина квадратов расстояний от каждой точки до центра того кластера, к которому она отнесена. Для этого, как правило, используется итеративный алгоритм:

1. Выбор k произвольных исходных центров $C = \{c_i\}$, где $c_i = \frac{\sum_{j=1}^{d} u_{ij} m_j}{\sum_{j=1}^{d} u_{ij}}$ $C_i = \frac{\sum_{j=1}^{d} u_{ij} m_j}{\sum_{j=1}^{d} u_{ij}}$, а $M = \{m_j\}$ – множество данных;

2. Разбиение всех объектов на k групп, наиболее близких к одному из центров $U = \{u_{ij}\}$, где

$$u_{ij} = \begin{cases} 1 \text{ при } d(m_j, c_i) = \min\limits_{1 \le k \le c} d(m_j, c_k) \\ 0 \text{ иначе} \end{cases}$$

$$u_{ij} = \begin{cases} 1, \text{при } d(m_j, c_i) = \min\limits_{1 \le k \le c} d(m_j, c_k) \\ 0, \text{иначе} \end{cases}.$$

3. Вычисление новых центров кластеров, пока центры не перестанут меняться, то есть пока решение не будет удовлетворять целевой функции $J(M, U, C) = \sum_{i=1}^{c} \sum_{j=1}^{d} u_{ij} d_A^2(m_j, c_i)$

На шаге 3, *в* рамках полученных кластеров формируются данные о соседстве аномалий и нормальных события, или соседстве переходных событий. При поступлении новой информации в виде нового вектора $x \in \mathbb{R}^N$ – объект с N признаками

из очередного пяти минутного интервала формируется текущее кластерное пространство комбинаций атрибутов $\mathcal{C}$ из Этапа 1. Используя кластерные модели (центроиды) из Этапа 2 для каждой пары (C,k) $\mu_1,\dots\mu_k \in \mathbb{R}^r$ – центроиды кластеров, полученные на подпространстве C определим для каждой комбинации атрибутов $C \in \mathcal{C}$ и каждого $k \in \{2,3,4,5\}$ номер кластера, в которые попадает объект x. Кластер для объекта x определяется как $\hat{y}_{C,k} = \arg\min_{j\in\{1,\dots k\}} \left\| x_C - \mu_j \right\|^2$, где $x_c \in \mathbb{R}^r$ – вектор, соответствующий подмножеству атрибутов $C \subseteq \{1,\dots,N\}$. В результате можно узнать с какими типами событий (кластерами) соседствует новое наблюдение.

На шаге 4 осуществляется анализ состава кластера и вычисление распределения меток среди соседей. С этой целью для каждого объекта x_i, комбинации C, и значения k определяется кластер G, в который он попал и рассчитывается доля объектов с каждой меткой $y_i \in \{0,1,2\}, i = 1,\dots,T$ в этом кластере. В результате кластеризации для каждой комбинации C и каждого k, каждому объекту сопоставлен номер кластера $\hat{y}_i^{C,k}$ и вычисляется матрица процентного состава соседей (p_0, p_1, p_2) среди объектов в том же кластере, что и x_i. Тогда доля объектов с меткой $m \in \{0,1,2\}$ в кластере G для объекта x_i оценивается соотношением $p_{C,k}^{(m)}(x_i) = \frac{1}{\left|G_{C,k}(x_i)\right|} \sum_{x_j \in G_{c,k}(x_i)} \mathbb{I}[y_i = m]$, где $\mathbb{I}$ – индикаторная функция; $G_{C,k}(x_i) = \left\{ x_j \middle| \hat{y}_j^{C,k} = \hat{y}_i^{C,k} \right\}$ – множество объектов в том же кластере, что и x_i; $y_j \in \{0,1,2\}$ – истинная метка объекта x_j.

Поскольку таких пространств будет десятки тысяч, можно получить пространственную область поведенческого профиля, каждая точка которой будет отражать процент соседства с событиями определённого типа. Итоговая картина представляет собой поведенческий профиль поскольку будет характеризовать поведенческие особенности события.

На шаге 5 производится визуализация кластерных пространств. С этой целью для каждой пары (C, k) выполняется 2D-проекция пространства с помощью ПО (PCA / t-SNE / UMAP), где каждая точка $x_i^C \in \mathbb{R}^r$ характеризует объект в подпространстве признаков C а её 2D-проекция: $z_i = Proj_2(x_i^C), z_i \in \mathbb{R}^2$ её координаты. Результат проекции характеризуются цветом в пространстве $\mathrm{RGB}_i = (p_0, p_1, p_2)$. Для каждой комбинации признаков C и значения k необходимо:

- Построить двухмерную проекцию подпространства признаков $D_c \in \mathbb{R}^{T\times r}$ с помощью PCA, t-SNE или UMAP;
- Визуализировать на полученной 2D-плоскости каждый объект в виде точки, так что для каждой пары (C,k) формируется серия 2D -проекций (рис. 6).

Каждая точка проекции характеризует объект. Проекции помогают понять геометрию кластеров , а также проверить, насколько логично разделяются объекты при разных k. Цвет точки $= (p_0, p_1, p_2)$, где p_0 – красный; p_1 – зелёный; p_2 – синий цвет точки характеризует номер кластера (от 0 до k – 1), в который объект попал при разбиении с данным значением k в k-means. Например, для $k = 3$, кластер 0 – красный, кластер 1 – зелёный, кластер 2 – жёлтый.

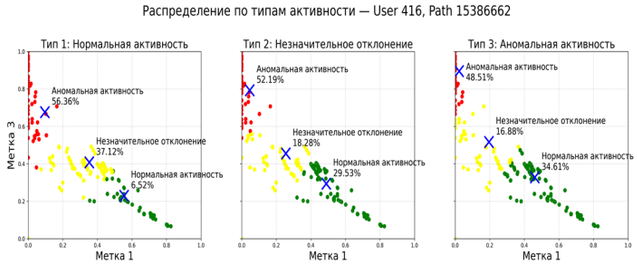

Рис. 6. Разбиение эталонной матрицы по типам на мета-кластеры

В результате кластерного анализа для каждого объекта определяется множество кластерных меток по всем (*C, k*) и проценты соседей по меткам (бутстреп-признаки).

Графическое представление кластерных пространств

Для каждого «нового объекта» после кластеризации (по комбинациям атрибутов) формируется множество признаков, отражающих его окружение (например, вектор частот меток в кластерах). В результате кластерного анализа для каждого объекта x формируется множество комбинаций признаков $\mathcal{C}$. Для каждой комбинации признаков C и каждого k, формируется вектор $f_{C,k}(x) = (p_0^{C,k}, p_1^{C,k}, p_2^{C,k})$. Совокупность подобных векторов формирует «отпечаток» (signature) объекта: $F(x) = \left\{ f_{C,k}(x) \middle| C \in \mathcal{C}, k \in \{2,3,4,5\} \right\}$. Поскольку аналогичные множества признаков имеются и у эталонных объектов/классов то сравнивая оба множества принимается решение «насколько "похож" новый объект на эталон по структуре кластерных соседств? Для сравнения множеств между используются метрики расстояний между множествами векторов. Каждое множество $F(x)$ нтерпретируется как распределение точек в 3D-пространстве (по p_0, p_1, p_2) и можно измерить расстояние до эталонного множества F_{ref}, используя например среднее попарное расстояние между точками $\mathrm{Dist}(x,x') = \frac{1}{\left|F(x)\right| \cdot \left|F(x')\right|} \sum_{u\in F(x)} \sum_{v\in F(x')} \left\| u - v \right\|_2$.

Для оценки степени сходства текущего поведения с аномальными шаблонами может быть использована метрика расстояния между многомерными векторами, например евклидова $S(x_i) = \min_{x_r \in \mathcal{F}_{anom}} \sqrt{\sum_{C,k} \sum_{m=0}^{2} \left(p_{C,k}^{(m)}(x_i) - p_{C,k}^{(m)}(x_r) \right)^2}$ или косинусное сходство $\mathrm{Sim}_{\cos}(x_i, x_r) = \frac{\langle \mathcal{F}(x_i), \mathcal{F}(x_r) \rangle}{\left\| \mathcal{F}(x_i) \right\| \cdot \left\| \mathcal{F}(x_r) \right\|}$.

Помимо расстояния между точками можно оценить плотность объектов в этих множествах по каждой из трёх меток $\{0,1,2\}$ в проекциях. Для этого можно использовать различные способы оценки плотности. Например, рассмотрев каждую точку $f_{C,k}(x) = (p_0, p_1, p_2)$ как 3D-точку можно построить оценку плотности по каждой из проекций p_0, p_1, p_2.

Метрика расстояния позволяет оценить, насколько текущее поведение пользователя близко по структуре к ранее выявленным аномальным сценариям. Если её значение оказывается высоким, это может свидетельствовать о возрастании вероятности угрозы, связанной с действиями пользователя информационной системы [7]. Существенным преимуществом такого показателя является возможность фиксировать потенциально опасные состояния даже в тех случаях, когда ошибка реконструкции автокодировщика не указывает на явную аномалию. Кроме того, использование метрики расстояния позволяет ранжировать наблюдаемые события по степени риска — от общего подозрительного поведения до действий, имеющих выраженное сходство с инсайдерскими или вредоносными сценариями.

Введённые метрики отклонения от эталонного профиля могут использоваться при принятии решений в различных формах: как плотностные характеристики по отдельным классам, как набор признаков или как расстояния, применяемые для отбора и фильтрации аномалий. При этом наблюдение за пользователем строится на последовательной фиксации событий его активности в виде журналов. Из логов извлекаются атрибуты, преобразуемые в векторное представление, после чего данные агрегируются в пятиминутные интервалы, внутри которых определяются кластерные соседства. Это обеспечивает возможность непрерывного формирования поведенческих профилей пользователей и последующего отслеживания их изменений в процессе функционирования системы.

4 Выводы

Разработана обобщенная структура алгоритма обнаружения редкого аномального поведения пользователя информационных систем, состоящая из пяти этапов. Для обнаружения аномальной активности пользователей кластерное представление нового объекта сравнивается с эталонными объектами по метрике сходства, что позволяет оценить, на сколько поведение пользователя похоже на ранее зафиксированные аномалии.

Введенные сравнительные метрики оценки отклонения от эталона можно использовать для принятия решения. Это могут быть плотностные профили по каждому классу, признаки или метрики расстояний для отбора или фильтрации аномалий. Фиксируя текущий профиль нормального или аномального функционирования, можно наблюдать за сотрудником (пользователем информационной системы) и каждый раз, когда он воспроизводит какие-то события фиксировать их в виде набора логов. Из логов извлекаются атрибуты, формируется вектор, из которого после агрегирования формируются 5-минутные интервалы, в рамках которых формируются кластерные соседства. В результате в процессе работы системы можно непрерывно получать поведенческие профили для каждого пользователя и отслеживать их характеристики.

Литература

1. *Rida Nasir* и др. Behavioral Based Insider Threat Detection Using Deep Learning // IEEE Access. 2021.
2. *Miguel Villarreal-Vasquez* и др. Hunting for Insider Threats Using LSTM-Based Anomaly Detection // IEEE TDSC. 2023.
3. *Simon Bertrand* и др. Unsupervised User-Based Insider Threat Detection Using BGMM // PST. 2023.
4. *Taher Al-Shehari* и др. Insider Threat Detection Using Isolation Forest // IEEE Access. 2023.
5. *Шелухин О. И.* Анализ методов обнаружения редкой аномальной активности пользователей информационных систем // Инженерный вестник Дона. 2026. № 1.
6. *Xiao H., Zhang W., Lu H.* Unveiling Shadows: A Comprehensive Framework for Insider Threat Detection Based on Statistical and Sequential Analysis // J. Inf. Secur. Appl. 2023. Vol. 75. DOI: 10.1016/j.jisa.2023.103582.
7. *Буйневич М. В., Власов Д. С., Моисеенко Г. Ю.* Комбинирование способов выявления инсайдеров больших информационных систем // Вопросы кибербезопасности. 2024. № 3(61). С. 2-13.
8. *Буйневич М. В., Власов Д. С., Моисеенко Г. Ю.* Повышение «устойчивости» регламентов деятельности как способ противодействия неумышленному инсайдингу // Вопросы кибербезопасности. 2024. № 6(64). С. 108-116.
9. *Sridevi D.* и др. ML and DL Techniques for Insider Threats // ICCSAI. 2023.
10. *Gayathri R. G.* и др. Hybrid DL Model with SPCAGAN // arXiv. 2022.
11. *Zewdie M.* и др. DNN for Insider Threats and Social Engineering // ICECET. 2024.
12. *Erramilli V.* How Salesforce Helps Protect You from Insider Threats // Salesforce Engineering. 2023. Режим доступа: https://engineering.salesforce.com/how-salesforce-helps-protect-you-from-insider-threats-5f9ae8b0e55d
13. *Шелухин О. И., Осин А. В., Хализев К. А.* Обнаружение аномальной активности пользователей информационной системы на основе анализа кластерных соседств // Материалы 34-й научно-технической конференции «Методы и технические средства обеспечения безопасности информации». Санкт-Петербург, 2025. № 34. С. 11-14.
14. *Шелухин О. И., Зегжда Д. П., Раковский Д. И., Самарин Н. Н., Александрова Е. Б.* Интеллектуальные технологии информационной безопасности. Москва: Горячая линия – Телеком, 2023. 384 с.
15. *Шелухин О. И., Гармашев А. В.* Обнаружение аномальных выбросов телекоммуникационного трафика методами дискретного вейвлет-анализа // Электромагнитные волны и электронные системы. 2012. Т. 17, № 2. С. 15-26. EDN OYMAQX.
16. *Шелухин О. И., Антонян А. А.* Анализ изменений фрактальных свойств телекоммуникационного трафика вызванных аномальными вторжениями // T-Comm: Телекоммуникации и транспорт. 2014. Т. 8, № 6. С. 61-64. EDN SGWEGX.
17. *Шелухин О. И., Осин А. В., Костин Д. В.* Мониторинг и диагностика аномальных состояний компьютерной сети на основе изучения "исторических данных" // T-Comm: Телекоммуникации и транспорт. 2020. Т. 14, № 4. С. 23-30. DOI 10.36724/2072-8735-2020-14-4-23-30. EDN AQKPYX.
18. *Шелухин О. И., Симонян А. Г., Ванюшина А. В.* Формирование исходных данных и анализ программного обеспечения для классификации приложений трафика методом машинного обучения // T-Comm: Телекоммуникации и транспорт. 2017. Т. 11, № 1. С. 67-72. EDN YGIYBV.
19. *Шелухин О. И., Раковский Д. И.* Бинарная классификация многоатрибутных размеченных аномальных событий компьютерных систем с помощью алгоритма SVDD // Наукоемкие технологии в космических исследованиях Земли. 2021. Т. 13, № 2. С. 74-84. DOI 10.36724/2409-5419-2021-13-2-74-84. EDN JBAPWV.
20. *Хализев К. А., Осин А. В.* Прогнозирование редких событий на основе анализа графлетов взаимодействия в социальных сетях // Инженерный вестник Дона. 2025. № 4(124). С. 336-348. EDN QDKFMF.

DETECTING RARE USER BEHAVIOR ANOMALIES IN INFORMATION SYSTEMS WITH MACHINE LEARNING

Oleg. I. Sheluhinv, *Moscow Technical University of Communications and Informatics, Moscow, Russia,* ***sheluhin@mail.ru***
Andrey V. Osin, *Moscow Technical University of Communications and Informatics, Moscow, Russia,* ***a.v.osin@mtuci.ru***

Abstract
Aim of the study: to improve the effectiveness of countering insider threats in information systems by developing an adequate model of anomalous user behavioral profiles in CRM systems using machine-learning methods, enabling the identification of potential insiders. Research methods: an analytical review of relevant scientific publications on detecting anomalous user activity in information systems; behavioral analysis combined with machine-learning and deep-learning algorithms, which opens new opportunities for the early detection of insider threats. Results: a generalized algorithm for detecting rare anomalous user behavior in information systems has been developed. It is shown how machine-learning techniques can be used to build a reference repository of clustered patterns representing anomalous objects. To detect anomalous user activity, it is proposed to compare the cluster-based representation of a new object with reference patterns using a similarity metric, which makes it possible to assess the extent to which a user's behavior resembles previously recorded anomalies. Scientific novelty: the novelty of the work lies in an original approach to countering rare anomalous user activity in information systems based on comparing typical normal behavioral profiles with insider behavioral profiles using data mining and intelligent data analysis methods.

Keywords: *information systems, insider, anomalous activity, behavioral profile, cluster neighborhoody.*

References

[1] R. Nasir, M. Afzal, R. Latif, and W. Iqbal, "Behavioral Based Insider Threat Detection Using Deep Learning," *IEEE Access*, 2021, vol. 9, pp. 143266-143274, doi: 10.1109/ACCESS.2021.3118297.
[2] M. Villarreal-Vasquez, G. Modelo-Howard, S. Dube, and B. Bhargava, "Hunting for Insider Threats Using LSTM-Based Anomaly Detection," *IEEE Trans. Dependable Secure Comput.*, 2023, vol. 20, no. 1, pp. 451-462, doi: 10.1109/TDSC.2021.3135639.
[3] S. Bertrand, J. Desharnais, and N. Tawbi, "Unsupervised User-Based Insider Threat Detection Using Bayesian Gaussian Mixture Models," in *Proc. Privacy, Security and Trust Conf. (PST)*, 2023, pp. 1-10, doi: 10.1109/PST58708.2023.10320169.
[4] T. Al-Shehari, M. S. Al-Razgan, T. Alfakih, R. A. Alsowail, and S. Pandiaraj, "Insider Threat Detection Model Using Anomaly-Based Isolation Forest Algorithm," *IEEE Access*, 2023, vol. 11, pp. 118170-118185, doi: 10.1109/ACCESS.2023.3326750.
[5] O. I. Sheluhin, "Analysis of Methods for Detecting Rare Anomalous Activity of Information-System Users," *Engineering Bulletin of the Don*, 2026, no. 1, Art. 10690. [Online]. Available: https://www.ivdon.ru/ru/magazine/archive/n1y2026/10690. Accessed: Feb. 1, 2026.
[6] H. Xiao, W. Zhang, and H. Lu, "Unveiling Shadows: A Comprehensive Framework for Insider Threat Detection Based on Statistical and Sequential Analysis," *Computers & Security*, 2024, vol. 138, Art. 103665, doi: 10.1016/j.cose.2023.103665.
[7] M. V. Buinevich, D. S. Vlasov, and G. Yu. Moiseenko, "Combining Methods for Identifying Insiders in Large Information Systems," *Voprosy Kiberbezopasnosti*, 2024, no. 3(61), pp. 2-13.
[8] M. V. Buinevich, D. S. Vlasov, and G. Yu. Moiseenko, "Increasing the 'Resilience' of Activity Regulations as a Method to Counter Unintentional Insider Threats," *Voprosy Kiberbezopasnosti*, 2024, no. 6(64), pp. 108-116.
[9] D. Sridevi, L. Kannagi, G. Vivekanandan, and S. Revathi, "Detecting Insider Threats in Cybersecurity Using Machine Learning and Deep Learning Techniques," in *Proc. Int. Conf. on Communication, Security and Artificial Intelligence (ICCSAI)*, Greater Noida, India, 2023, pp. 871-875, doi: 10.1109/ICCSAI59366.2023.10397460.
[10] R. G. Gayathri, A. Sajjanhar, and Y. Xiang, "Hybrid Deep Learning Model using SPCAGAN Augmentation for Insider Threat Analysis," arXiv:2203.02855, 2022, doi: 10.48550/arXiv.2203.02855.
[11] M. Zewdie, S. Yigezu, and A. Limenih, "Deep Neural Networks for Detecting Insider Threats and Social Engineering Attacks," in *Proc. 2nd Int. Conf. on Electrical, Computer and Energy Technologies (ICECET)*, 2024, pp. 1-8, doi: 10.1109/ICECET61485.2024.10698519.
[12] V. Erramilli, "How Salesforce Helps Protect You From Insider Threats," Salesforce Engineering Blog, Nov. 14, 2019. [Online]. Available: https://engineering.salesforce.com/how-salesforce-helps-protect-you-from-insider-threats-5f9ae8b0e55d/. Accessed: Feb. 1, 2026.
[13] O. I. Sheluhin, A. V. Osin, and K. A. Khalizev, "Detection of Anomalous User Activity in an Information System Based on Cluster Neighborhood Analysis," in *Proc. 34th Sci.-Tech. Conf. "Methods and Technical Means of Information Security"*, St. Petersburg, Russia, 2025, no. 34, pp. 11-14.
[14] O. I. Sheluhin, D. P. Zegzhda, D. I. Rakovskiy, N. N. Samarin, and E. B. Aleksandrova, Intelligent Information Security Technologies. Moscow, Russia: Goryachaya Liniya-Telecom, 2023, 384 p.
[15] O. I. Sheluhin and A. V. Garmashev, "Detection of Anomalous Outliers in Telecommunication Traffic Using Discrete Wavelet Analysis Methods," *Elektromagnitnye Volny i Elektronnye Sistemy*, vol. 17, no. 2, pp. 15-26, 2012.
[16] O. I. Sheluhin and A. A. Antonyan, "Analysis of Changes in the Fractal Properties of Telecommunication Traffic Caused by Anomalous Intrusions," *T-Comm*, vol. 8, no. 6, pp. 61-64, 2014.
[17] O. I. Sheluhin, A. V. Osin, and D. V. Kostin, "Monitoring and Diagnosis of Abnormal States of a Computer Network Based on the Study of 'Historical Data'," *T-Comm*, vol. 14, no. 4, pp. 23-30, 2020, doi: 10.36724/2072-8735-2020-14-4-23-30.
[18] O. I. Sheluhin, A. G. Simonyan, and A. V. Vanyushina, "Formation of Source Data and Analysis of Software for Traffic Application Classification Using Machine Learning," *T-Comm*, vol. 11, no. 1, pp. 67-72, 2017.
[19] O. I. Sheluhin and D. I. Rakovskiy, "Binary Classification of Multi-Attribute Labeled Anomalous Events in Computer Systems Using the SVDD Algorithm," *High Technologies in Earth Space Research*, vol. 13, no. 2, pp. 74-84, 2021, doi: 10.36724/2409-5419-2021-13-2-74-84.
[20] K. A. Khalizev and A. V. Osin, "Prediction of Rare Events Based on the Analysis of Interaction Graphlets in Social Networks," *Inzhenernyi Vestnik Dona*, no. 4(124), pp. 336-348, 2025.