ТЕХНИКАЛЫҚ ҒЫЛЫМДАР
ТЕХНИЧЕСКИЕ НАУКИ
TECHNICAL SCIENCE

CENTRAL ASIAN SCIENTIFIC JOURNAL
ISSN 3135-3061
2026. №4 (32), Том 2. С. 451–461
DOI: https://doi.org/10.5281/zenodo.24342602
Сайт журнала: https://cajournal.kz

УДК 378.147:004.8

РАЗРАБОТКА AI-СИСТЕМЫ ОБНАРУЖЕНИЯ АНОМАЛЬНОГО ПОВЕДЕНИЯ ПОЛЬЗОВАТЕЛЕЙ В КОРПОРАТИВНОЙ СЕТИ

З.А. Куванышбек, А.А. Асет
Казахский национальный исследовательский технический университет им. К.И.Сатпаева, Алматы, Казахстан

Куванышбек Заңғар Асқарұлы — студент 4 курса образовательной программы «Информационная безопасность», Казахский национальный исследовательский технический университет им. К.И.Сатпаева, Алматы, Казахстан.

Асет Арсен Арманұлы — студент 4 курса образовательной программы «Информационная безопасность», Казахский национальный исследовательский технический университет им. К.И.Сатпаева, Алматы, Казахстан.

Научный руководитель: Абдуллаевна Асель Сейдуллаевна — Магистр технических наук, Старший преподаватель, Казахский национальный исследовательский технический университет им. К.И.Сатпаева, Алматы, Казахстан.

© З.А. Куванышбек, А.А. Асет, 2026

Аннотация: Рассмотрена задача обнаружения аномального поведения пользователей в корпоративной сети как элемента противодействия внутренним угрозам и компрометации учётных записей. Предложена архитектура AI-системы, включающая сбор журналов событий, формирование суточных профилей «пользователь–день», персонализацию признаков, ансамбль моделей обнаружения аномалий без учителя и модуль объяснения оценок для аналитика. На синтетическом наборе из 18 000 пользователе-дней с тремя сценариями аномалий сопоставлены изолирующий лес, метод главных компонент и автокодировщик. Персонализация признаков повысила среднюю точность изолирующего леса с 0,703 до 0,798, а автокодировщик показал наилучший результат (0,898). Исследованы влияние бюджета оповещений, длины окна обучения и загрязнения обучающих данных. Оценка выполнена по средней точности, ROC-AUC, F1-мере и Precision@k на пяти независимых запусках. Результаты предварительные, получены на синтетических данных и требуют проверки на реальных журналах предприятия.

Ключевые слова: обнаружение аномалий; поведение пользователей; внутренние угрозы; информационная безопасность; машинное обучение; автокодировщик; изолирующий лес.

ВВЕДЕНИЕ

Корпоративная сеть остаётся основной целью целенаправленных атак, причём нередко злоумышленник действует не через эксплуатацию уязвимостей, а от имени легитимной учётной записи: после компрометации пароля, при злоупотреблении привилегиями или в результате умышленных либо ошибочных действий сотрудника. Такие действия не имеют устойчивых сигнатур, поэтому сигнатурные средства обнаружения вторжений их, как правило, пропускают. В руководстве NIST SP 800-94 методы обнаружения разделены на сигнатурные, основанные на аномалиях и на анализе состояний протоколов; подход на основе аномалий предполагает построение профиля нормальной активности и поиск отклонений от него [1].

Для описания действий противника внутри сети широко используется база знаний MITRE ATT&CK, которая систематизирует тактики и техники атак, в том числе использование действительных учётных записей, перемещение внутри сети и вынос данных [2]. Многие из этих техник проявляются в журналах событий как отклонение поведения пользователя от его обычного: нетипичное время входа, обращение к нехарактерным хостам, резкий рост объёма исходящих данных.

Согласно обзору Chandola и соавт., аномалии бывают точечными, контекстными и коллективными [3]. Поведение пользователя — типичный пример контекстной аномалии: один и тот же набор действий нормален для системного администратора и подозрителен для бухгалтера. Это обосновывает необходимость сравнивать активность не с «усреднённым» пользователем, а с историей самого пользователя и его роли.

Применение машинного обучения в обнаружении вторжений имеет принципиальные трудности. Sommer и Paxson указывают на высокую цену ошибок, большое разнообразие нормального трафика и сложность корректной оценки таких систем [4]. На практике это означает, что система должна выдавать аналитику короткий ранжированный список с понятным объяснением, а не поток срабатываний.

В работе Tuor и соавт. на наборе CERT Insider Threat v6.2 неконтролируемые глубокие и рекуррентные нейронные сети превзошли базовые методы на основе PCA, SVM и изолирующего леса, а разложение оценки аномальности по признакам повысило интерпретируемость [5]. Подход к генерации синтетических данных о внутренних угрозах описан в [6]. Для последовательностей системных журналов предложены модели на основе LSTM [7], а обзор методов глубокого обнаружения аномалий приведён в [8].

Обзор Chandola и соавт. выделяет основные трудности обнаружения аномалий: сложность описания «нормальной» области, изменение нормального поведения со временем, наличие шума в данных и отсутствие размеченных примеров [3]. Все эти трудности характерны для поведения пользователей: рабочие роли и проекты меняются, журналы содержат случайные выбросы, а подтверждённых инцидентов для обучения недостаточно. NIST также отмечает, что методы на основе аномалий склонны к ложным срабатываниям, поскольку безобидная активность может существенно отличаться от построенного профиля [1].

Современные обзоры систематизируют подходы глубокого обнаружения аномалий по тому, как используется нейронная сеть: для извлечения признаков, для обучения представлений нормального поведения или для непосредственного вычисления оценки аномальности [8]. Автокодировщики относятся ко второй группе: сеть сжимает вход в компактное представление и восстанавливает его, а ошибка восстановления служит мерой аномальности [8, 10]. Изолирующий лес [9], напротив, не использует понятий расстояния и плотности и отличается малой вычислительной сложностью, что делает его удобным базовым методом.

Sommer и Paxson подчёркивают, что обнаружение аномалий отличается от типичных задач машинного обучения, где модель распознаёт заранее известные классы: система должна выделять неизвестное, а цена пропуска и ложной тревоги в безопасности высока [4]. Поэтому в данной работе оценка качества ориентирована на нагрузку аналитика: помимо традиционных метрик анализируются точность и полнота при фиксированном бюджете оповещений.

Методы обнаружения аномалий принято делить по наличию разметки на контролируемые, частично контролируемые и неконтролируемые, а по принципу работы — на статистические, основанные на расстоянии и плотности, на реконструкции и на классификации [3, 8]. В задачах безопасности размеченных примеров вредоносного поведения мало, а сами атаки постоянно меняются, поэтому наибольший практический интерес представляют методы без учителя, которые обучаются на преобладающей нормальной активности. К ним относятся изолирующий лес, который изолирует редкие объекты случайными разбиениями пространства признаков, и модели реконструкции (метод

главных компонент, автокодировщики), для которых аномальным считается объект, плохо восстанавливаемый моделью нормального поведения [8].

Практическая значимость работы состоит в том, что предложенная схема опирается на данные, которые уже собираются в большинстве организаций (журналы аутентификации, сетевых шлюзов и файловых серверов), и не требует размеченных инцидентов. Это позволяет использовать её как дополнительный слой к сигнатурным средствам защиты [1], а не как их замену.

Цель работы — разработать архитектуру AI-системы обнаружения аномального поведения пользователей в корпоративной сети и экспериментально сопоставить три метода обнаружения аномалий без учителя с учётом персонализации признаков. Задачи: описать архитектуру системы; сформировать набор данных и признаки; реализовать и сравнить модели; оценить результаты и ограничения.

РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ

1. Модель угроз и требования к системе.

Рассматриваются три класса нарушителей: внешний атакующий, получивший доступ к учётной записи сотрудника (например, после фишинга); внутренний нарушитель, умышленно выносящий данные или расширяющий свои права; небрежный сотрудник, чьи действия создают риск (массовое копирование данных на личные носители). Общим для всех трёх классов является то, что действия выполняются от имени легитимной учётной записи и формально разрешены политиками доступа. Принято, что журналы событий не подделываются нарушителем до попадания в систему сбора, а пороги срабатывания ему неизвестны.

Из модели угроз и особенностей задачи [3, 4] следует набор требований, приведённый в таблице 1. Часть из них (обучение без учителя, персонализация, ранжирование) реализована и проверена в эксперименте, другие (объяснимость, адаптивность, защита данных) заложены в архитектуру, но в данной работе экспериментально не оценивались.

Таблица 1 – Требования к системе и их реализация

Требование	Содержание	Реализация
Обучение без учителя	Размеченных инцидентов мало или нет	Изолирующий лес, PCA, автокодировщик
Персонализация	Сравнение с историей самого пользователя	z-оценки по формуле (1)
Ограниченная нагрузка	Аналитик разбирает конечное число оповещений	Ранжирование и бюджет оповещений
Объяснимость	Понятные причины срабатывания	Вклад признаков в ошибку (2); не оценивалось
Адаптивность	Поведение и роли меняются со временем	Скользящее окно, дообучение; не оценивалось
Защита данных	Обрабатываются сведения о действиях людей	Разграничение доступа, журнал обращений

2. Архитектура системы.

Предлагаемая система состоит из восьми последовательных блоков (рис. 1). Источниками данных служат журналы службы каталогов, VPN, прокси-серверов, файловых серверов и средств контроля конечных точек. События нормализуются и передаются в SIEM, где для каждой пары «пользователь–день» агрегируются признаки: время входа, число входов, число обращений к файлам, число уникальных хостов, число подключений съёмных носителей и объём исходящих данных. Далее признаки персонализируются относительно истории пользователя и подаются в ансамбль моделей, который выдаёт оценку аномальности. Пользователе-дни ранжируются по оценке, и аналитик центра мониторинга (SOC) получает список наиболее подозрительных с указанием признаков, внесших наибольший вклад. Результаты расследований используются для дообучения. Выбран режим

обучения без учителя, поскольку размеченные инциденты в реальных организациях редки и неполны.



Рис. 1. Архитектура системы обнаружения аномального поведения

3. Данные и признаки.

Доступ к реальным журналам предприятия для исследования отсутствовал, поэтому использован собственный упрощённый генератор синтетических данных (общая идея синтетических данных о внутренних угрозах рассмотрена в [6]). Сгенерированы журналы 300 пользователей четырёх ролей за 60 дней с учётом снижения активности в выходные. Набор содержит 18 000 пользователе-дней. Первые 40 дней использованы для обучения (считаются свободными от инцидентов), последние 20 дней (6 000 пользователе-дней) — для проверки. В проверочную часть внедрено 2 % аномалий (120 пользователе-дней) трёх сценариев: а) вынос данных (объём исходящих данных увеличен в 5–12 раз, дополнительно 3–7 подключений USB, число обращений к файлам в 3–6 раз выше); б) внеурочное перемещение по сети (вход в 1–3 часа ночи или в 23:30, на 8–19 дополнительных хостов, число входов в 2–4 раза выше); в) «тихая» аномалия (умеренное одновременное увеличение числа обращений к файлам в 1,8–2,5 раза, объёма данных в 2–3 раза и числа хостов на 2–4).

Признаки выбраны так, чтобы они были доступны в типовых журналах и отражали стадии атаки, описанные в [2] (таблица 2). Источники журналов указаны как типичные для реальной сети; в эксперименте признаки порождались генератором.

Таблица 2 – Признаки пары «пользователь–день»

Признак	**Типичный источник**	**Связь с атакой [2]**
Час входа	Журнал аутентификации	Использование учётной записи вне обычного времени
Число входов	Журнал аутентификации	Подбор или массовое использование учётной записи
Число обращений к файлам	Файловые серверы	Сбор данных перед выносом
Число уникальных хостов	Журналы каталога, сетевые журналы	Перемещение внутри сети
Число подключений USB	Средства защиты конечных точек	Вынос данных на носитель
Объём исходящих данных, МБ	Прокси, межсетевой экран	Вынос данных по сети

Персонализация выполнялась по формуле:

$z_{u,j} = (x_{u,j} - \mu_{u,j}) / \sigma_{u,j}$ (1)

где x — значение признака j для пользователя u в данный день; μ и σ — среднее и стандартное отклонение этого признака по обучающим дням того же пользователя. На вход моделей подавался 12-мерный вектор: шесть стандартизированных исходных признаков и шесть персонализированных z-оценок.

4. Модели и метрики.

Сопоставлены четыре конфигурации. Изолирующий лес [9] (200 деревьев) обучен на исходных признаках (IF, исходные) и на персонализированном векторе (IF, персонализ.). Метод главных компонент (PCA) с четырьмя компонентами использован как модель реконструкции. Автокодировщик [10] с архитектурой 12–16–4–16–12 (функция активации tanh, ранняя остановка, не более 300 эпох) обучен восстанавливать вход. Для PCA и автокодировщика оценка аномальности равна суммарной квадратичной ошибке реконструкции:

$s(x) = \Sigma_j (x_j - \hat{x}_j)^2$ (2)

Реализация выполнена на Python с использованием библиотеки scikit-learn [11]. Поскольку доля аномалий мала, основной метрикой выбрана средняя точность (AP, площадь под кривой «точность–полнота»), которая информативнее ROC-AUC при сильном дисбалансе классов [12]. Дополнительно рассчитаны ROC-AUC, максимальная F1-мера по всем порогам и Precision@k, где k равно числу аномалий (120). Эксперимент повторён на пяти независимо сгенерированных наборах; приведены среднее значение и стандартное отклонение.

Для исключения утечки информации из проверочной части в обучение использовано разбиение по времени: стандартизация признаков, средние и стандартные отклонения для персонализации, а также параметры всех моделей определялись только по первым 40 дням, а оценка проводилась на последних 20 днях. Такой протокол соответствует реальной эксплуатации, когда модель обучается на прошлом и оценивает новые события. Каждая из пяти генераций набора использовала собственное начальное значение генератора случайных чисел (от 0 до 4), что обеспечивает воспроизводимость эксперимента; модели каждый раз обучались заново с тем же начальным значением.

Оценка аномальности в изолирующем лесу определяется средней длиной пути h(x) до изоляции объекта в деревьях ансамбля: s(x, n) = 2^(−E[h(x)]/c(n)), где c(n) = 2H(n − 1) − 2(n − 1)/n — нормирующая константа, H(i) — гармоническое число [9]. Значения, близкие к единице, соответствуют аномалиям, так как редкие объекты изолируются за малое число разбиений. В методе главных компонент объект восстанавливается по k = 4 первым компонентам: $\hat{x} = \mu + W \cdot W^T (x - \mu)$, где W — матрица компонент, μ — среднее обучающей выборки. Автокодировщик задаёт отображение $\hat{x} = g(f(x))$, где кодировщик f сжимает 12 признаков до 4 значений, а декодер g восстанавливает вход; обучение минимизирует среднюю квадратичную ошибку на обучающих днях [10]. Узкий слой не позволяет сети «запомнить» вход целиком, поэтому типичное поведение восстанавливается хорошо, а редкие сочетания признаков — плохо.

5. Основное сравнение моделей.

Результаты сравнения представлены в таблице 3 и на рис. 2.

Таблица 3 – Сравнение моделей (среднее ± СО по 5 запускам)

Модель	ROC-AUC	AP	F1 (макс.)	Precision@k
IF, исходные признаки	0,941 ± 0,013	0,703 ± 0,039	0,676 ± 0,028	0,647 ± 0,039
IF, персонализация	0,960 ± 0,013	0,798 ± 0,046	0,778 ± 0,042	0,745 ± 0,046
PCA, персонализация	0,965 ± 0,012	0,796 ± 0,038	0,782 ± 0,039	0,735 ± 0,034
Автокодировщик, персонализация	**0,974 ± 0,012**	**0,898 ± 0,025**	**0,872 ± 0,028**	**0,847 ± 0,034**

Персонализация признаков улучшила все показатели изолирующего леса: AP выросла с 0,703 до 0,798, максимальная F1-мера — с 0,676 до 0,778. Это согласуется с контекстной природой аномалий поведения [3]: отклонение от собственной нормы пользователя информативнее отклонения от общей. PCA на персонализированных признаках показал результат, близкий к изолирующему лесу (AP 0,796 против 0,798; разница меньше стандартного отклонения).

Наилучший результат продемонстрировал автокодировщик: AP 0,898, Precision@k 0,847, то есть около 85 % из 120 пользователе-дней с наибольшей оценкой оказались внедрёнными аномалиями. Его преимущество по AP над изолирующим лесом и PCA на персонализированных признаках составило около 0,10, что превышает разброс между запусками, однако статистическая значимость при пяти запусках не оценивалась. Вероятная причина преимущества — способность автокодировщика учитывать нелинейные зависимости между признаками, что важно для «тихих» аномалий сценария в), в которых ни один признак по отдельности не выходит за допустимые границы.

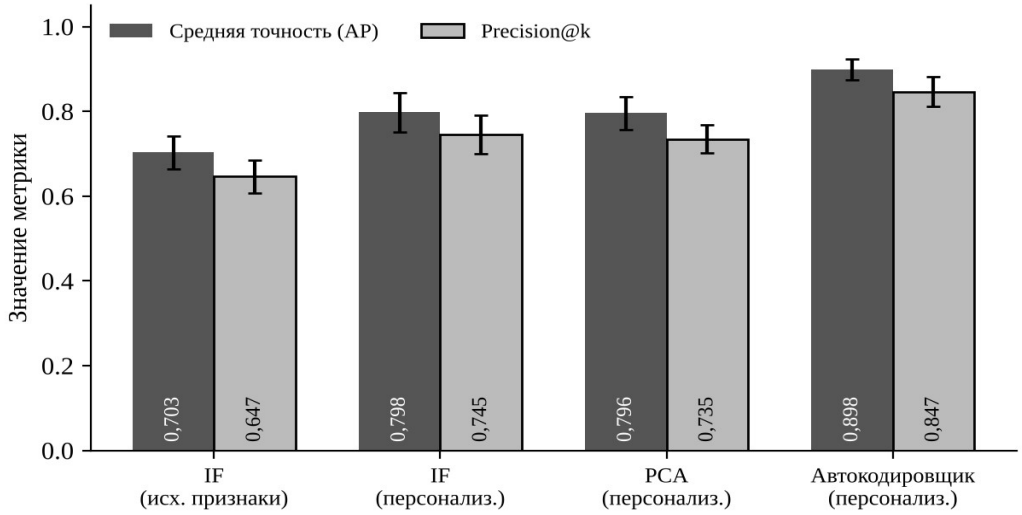


Рис. 2. Средняя точность (AP) и Precision@k для четырёх конфигураций

Ошибка реконструкции (2) раскладывается на вклад отдельных признаков, что позволяет показывать аналитику, какие именно параметры поведения отличают подозрительный день, по аналогии с подходом [5]. В данной работе этот модуль на практике не оценивался.

6. Результаты по сценариям и бюджету оповещений.

Дополнительные эксперименты выполнены на повторно реализованном (векторизованном) генераторе с теми же параметрами, но иным порядком розыгрыша случайных чисел при внедрении аномалий, поэтому значения несколько отличаются от таблицы 3 (например, AP автокодировщика 0,903 против 0,898). Используются те же пять запусков и то же разбиение по времени.

В таблице 4 приведена доля найденных аномалий каждого сценария среди 120 пользователе-дней с наибольшей оценкой (бюджет оповещений 2 %). Вынос данных и внеурочное перемещение по сети обнаруживаются хорошо всеми персонализированными моделями; автокодировщик находит все аномалии второго сценария во всех запусках. Наиболее трудна «тихая» аномалия, в которой ни один признак не выходит за границы нормы: здесь автокодировщик находит 0,56 аномалий против 0,33 у изолирующего леса и 0,23 у изолирующего леса без персонализации. PCA хуже других персонализированных моделей находит вынос данных (0,83), что, вероятно, связано с тем, что линейная проекция менее чувствительна к резким выбросам объёма и числа USB, чем нелинейная модель; эта гипотеза отдельно не проверялась.

Таблица 4 – Доля найденных аномалий по сценариям при бюджете оповещений 2 % (среднее ± СО)

Модель	Вынос данных	Внеурочное перемещение	«Тихая» аномалия
IF, исходные признаки	0,918 ± 0,011	0,829 ± 0,050	0,227 ± 0,093
IF, персонализация	0,966 ± 0,025	0,937 ± 0,036	0,333 ± 0,066
PCA, персонализация	0,831 ± 0,040	0,994 ± 0,012	0,419 ± 0,089
Автокодировщик	0,980 ± 0,020	1,000 ± 0,000	0,555 ± 0,087

Для оценки нагрузки на аналитика рассчитаны точность и полнота при фиксированном числе оповещений (таблица 5, рис. 3). Бюджет 0,5 %, 1 %, 2 % и 3 % соответствует 30, 60, 120 и 180 оповещениям на 6 000 пользователе-дней проверочной выборки. При бюджете 2 % число оповещений равно числу аномалий, поэтому точность и полнота совпадают.

Таблица 5 – Точность / полнота при фиксированном бюджете оповещений

Модель	0,5 % (30)	1 % (60)	2 % (120)	3 % (180)
IF, исходные признаки	0,99 / 0,25	0,93 / 0,46	0,66 / 0,66	0,49 / 0,73
IF, персонализация	1,00 / 0,25	0,98 / 0,49	0,74 / 0,74	0,54 / 0,80
PCA, персонализация	1,00 / 0,25	1,00 / 0,50	0,75 / 0,75	0,53 / 0,80
Автокодировщик	1,00 / 0,25	1,00 / 0,50	0,84 / 0,84	0,60 / 0,90

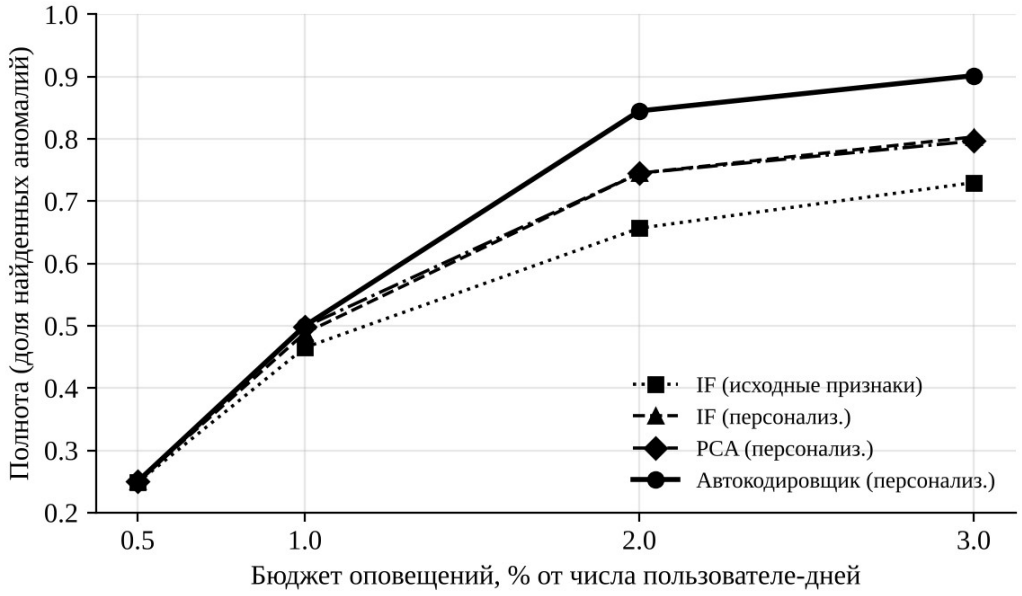


Рис. 3. Полнота в зависимости от бюджета оповещений

При бюджете 1 % автокодировщик выдаёт 60 оповещений без единого ложного, но находит лишь половину аномалий (полнота 0,50 — максимум для 60 оповещений при 120 аномалиях). Для достижения полноты около 0,90 требуется бюджет 3 %, при этом точность снижается до 0,60, то есть примерно 40 % оповещений будут ложными. Для изолирующего леса без персонализации при том же бюджете полнота составляет лишь 0,73. Таким образом, выбор бюджета — это компромисс между пропущенными аномалиями и нагрузкой на аналитика, и модель с более высоким качеством ранжирования позволяет достичь той же полноты при меньшем числе оповещений.

7. Абляция и чувствительность к параметрам.

Чтобы выяснить, что даёт прирост качества, автокодировщик обучен на трёх наборах входов: только стандартизированные исходные признаки, только персонализированные z-оценки и их объединение (таблица 6). Переход от исходных признаков к z-оценкам

повышает AP с 0,788 до 0,899, тогда как добавление исходных признаков к z-оценкам почти ничего не меняет (0,903, разница меньше стандартного отклонения). Следовательно, основной вклад вносит персонализация, а не архитектура сети.

Таблица 6 – Абляция входных признаков автокодировщика

Входные признаки	**AP (среднее ± СО)**	**ROC-AUC**
Исходные (6)	0,788 ± 0,028	0,964
Персонализированные z-оценки (6)	0,899 ± 0,021	0,977
Исходные + z-оценки (12)	0,903 ± 0,021	0,982

Чувствительность модели к параметрам исследована для автокодировщика (таблица 7). Длина обучающего окна существенно влияет на результат: при семи днях AP равна 0,825, при 40 днях — 0,903. Вероятная причина в том, что короткая история содержит мало наблюдений для устойчивой оценки среднего и разброса каждого пользователя, а также смешивает рабочие и выходные дни. Ширина узкого слоя, равная 4, оказалась лучше значений 2 и 8; значение 4 выбрано до этого анализа и по его результатам не подбиралось. Узкий слой из двух нейронов, по-видимому, слишком ограничивает восстановление нормального поведения, а из восьми — позволяет частично восстанавливать и аномальные объекты.

Наиболее значимой для практики оказалась чувствительность к загрязнению обучающих данных. Если в обучающие дни попадают неразмеченные аномалии, качество падает: при доле 1 % AP равна 0,833, при 5 % — 0,679 (при чистых данных 0,903). Это согласуется с замечанием о шуме в данных как одной из ключевых трудностей обнаружения аномалий [3] и означает, что в реальной системе обучающие данные необходимо очищать или применять робастные способы оценки профиля.

Таблица 7 – Чувствительность автокодировщика к параметрам (AP, среднее ± СО)

Параметр	**Значение**	**AP**
Длина окна обучения, дней	7	0,825 ± 0,028
	14	0,873 ± 0,028
	21	0,881 ± 0,028
	40	0,903 ± 0,021
Доля аномалий в обучающих данных	0 %	0,903 ± 0,021
	1 %	0,833 ± 0,032
	2 %	0,796 ± 0,047
	5 %	0,679 ± 0,049
Нейронов в узком слое	2	0,868 ± 0,025
	4	0,903 ± 0,021
	8	0,829 ± 0,033

8. Соответствие сценариев тактикам ATT&CK.

Для практической интерпретации результатов сценарии аномалий сопоставлены с тактиками и техниками базы знаний ATT&CK [2] (таблица 8). Сопоставление выполнено авторами на уровне общих тактик и носит иллюстративный характер: оно показывает, какие наблюдаемые признаки журналов могут указывать на соответствующую стадию атаки, но не означает, что любое срабатывание модели соответствует конкретной технике.

Таблица 8 – Сопоставление сценариев аномалий и тактик ATT&CK

Сценарий	Ключевые признаки	Тактика ATT&CK [2]
Вынос данных	объём исходящих данных, USB, обращения к файлам	Exfiltration (TA0010)
Внеурочное перемещение по сети	время входа, число хостов, число входов	Lateral Movement (TA0008); Valid Accounts (T1078)
«Тихая» аномалия	умеренные сдвиги нескольких признаков одновременно	Valid Accounts (T1078); Collection, Exfiltration

9. Обсуждение.

Совокупность результатов позволяет сделать несколько наблюдений. Во-первых, наибольший прирост даёт не выбор более сложной модели, а правильное представление данных: сравнение пользователя с его собственной историей повышает качество всех методов (таблицы 3 и 6), что согласуется с контекстной природой аномалий поведения [3]. Во-вторых, преимущество автокодировщика проявляется прежде всего на «тихих» аномалиях (таблица 4), где отклонение распределено по нескольким признакам; это сходится по направлению с выводом [5] о выигрыше глубоких неконтролируемых моделей над классическими, однако прямое сравнение с [5] некорректно, поскольку данные и протоколы оценки различаются.

В-третьих, практическая применимость определяется не только метриками ранжирования, но и бюджетом оповещений (таблица 5): модель с более высоким качеством ранжирования уменьшает долю ложных оповещений при заданной полноте, что напрямую снижает нагрузку на SOC. В-четвёртых, слабым местом подхода является зависимость от чистоты обучающих данных (таблица 7): уже 5 % неразмеченных аномалий в обучении заметно ухудшают результат. Поэтому первичное обучение следует проводить на периоде, проверенном аналитиками, и использовать профили, устойчивые к выбросам.

Следует учитывать, что все выводы получены на синтетических данных с известной структурой нормального поведения. Реальные профили пользователей менее регулярны, поэтому абсолютные значения метрик, вероятно, будут ниже, а соотношение между методами может отличаться. Роль настоящего эксперимента — показать работоспособность архитектуры, выявить факторы, определяющие качество, и задать план проверки на реальных данных.

10. Ограничения.

Результаты получены на синтетических данных, в которых аномалии заданы авторами, поэтому они, вероятно, оптимистичнее, чем будут на реальных журналах с дрейфом поведения, общими учётными записями, отпусками и сезонностью. Максимальная F1-мера определена по порогу, подобранному на проверочной выборке, и является верхней оценкой. Обучение автокодировщика ограничено 300 эпохами; гиперпараметры подробно не оптимизировались. Не учитывались последовательности событий внутри дня, для которых перспективны рекуррентные модели [7]. Наконец, как отмечено в [4], в реальной эксплуатации решающим становится объём ложных срабатываний, который на синтетическом наборе корректно оценить нельзя.

11. Рекомендации по внедрению.

Следующие рекомендации вытекают из архитектуры и ограничений работы и экспериментально не проверялись. Во-первых, порог срабатывания целесообразно задавать не значением оценки, а «бюджетом оповещений»: числом пользователе-дней в сутки, которое аналитики способны разобрать. Это напрямую связывает качество модели с нагрузкой на SOC и учитывает замечание о цене ложных срабатываний [4]. Во-вторых, профили пользователей необходимо периодически пересчитывать скользящим окном, чтобы модель адаптировалась к смене должности, проектов и сезонным колебаниям активности; при этом окно обучения не должно включать подтверждённые инциденты.

В-третьих, для новых сотрудников, у которых нет истории, персонализацию следует заменять сравнением с профилем роли или подразделения (холодный старт). В-четвёртых,

поскольку система обрабатывает данные о действиях конкретных людей, в промышленной эксплуатации необходимы разграничение доступа к оценкам, ведение журнала обращений аналитиков и соблюдение требований законодательства о персональных данных. Наконец, итоговое решение о реагировании должен принимать человек: оценка аномальности указывает на необходимость проверки, а не на факт нарушения.

12. Воспроизводимость эксперимента.

Для воспроизведения результатов достаточно параметров, приведённых в таблице 9. Генератор данных и все модели реализованы на Python с использованием библиотек NumPy, pandas и scikit-learn [11]; начальные значения генератора случайных чисел для пяти запусков равны 0–4. Данные не содержат сведений о реальных лицах и могут быть опубликованы вместе с программным кодом.

Таблица 9 – Параметры эксперимента

Параметр	**Значение**
Число пользователей / ролей	300 / 4
Длина журнала	60 дней (18 000 пользователе-дней)
Разбиение по времени	обучение — дни 1–40; проверка — дни 41–60
Доля аномалий в проверочной части	2 % (120 пользователе-дней), три сценария
Число признаков	6 исходных, 12 на входе моделей (с z-оценками)
Изолирующий лес	200 деревьев
PCA	4 главные компоненты, ошибка реконструкции
Автокодировщик	12–16–4–16–12, tanh, ранняя остановка, не более 300 эпох
Число независимых запусков	5 (значения среднее ± стандартное отклонение)

13. Направления дальнейшей работы.

Первоочередной задачей является проверка на публичных наборах данных о внутренних угрозах, в частности на наборах CERT, использованных в [5], и сопоставление результатов с опубликованными. Это позволит оценить, сохраняется ли выигрыш от персонализации и автокодировщика на данных с более сложной структурой, и выявить признаки, которые не удалось смоделировать генератором.

Вторым направлением является учёт временной структуры: последовательности событий внутри суток и между сутками можно моделировать рекуррентными сетями по аналогии с [7]. Третьим — проверка модуля объяснения: необходимо оценить, насколько разложение ошибки реконструкции по признакам помогает аналитикам быстрее принимать решение, для чего потребуется эксперимент с участием специалистов SOC. Наконец, требует исследования устойчивость к загрязнению обучающих данных (таблица 7): робастные оценки среднего и разброса, взвешивание обучающих примеров и итеративная очистка профилей могут снизить зависимость качества от чистоты истории.

ВЫВОДЫ

1. Предложена архитектура AI-системы обнаружения аномального поведения пользователей, включающая сбор и нормализацию журналов, формирование профилей «пользователь–день», персонализацию признаков, ансамбль моделей без учителя, ранжирование оценок и объяснение для аналитика с обратной связью.

2. На синтетическом наборе из 18 000 пользователе-дней показано, что персонализация признаков относительно истории пользователя повышает среднюю точность изолирующего леса с 0,703 до 0,798.

3. Абляция показала, что основной вклад в качество вносит персонализация: AP автокодировщика выросла с 0,788 на исходных признаках до 0,899 на z-оценках. Автокодировщик лучше других находит «тихие» аномалии (доля найденных 0,56 против 0,33 у изолирующего леса при бюджете 2 %).

© Central Asian Scientific Journal, 2026

4. Качество существенно зависит от длины обучающего окна и от чистоты обучающих данных: при 5 % неразмеченных аномалий AP снижается с 0,903 до 0,679, поэтому первичное обучение необходимо проводить на проверенном периоде.

5. Автокодировщик показал лучший результат среди сравниваемых моделей (AP 0,898; Precision@k 0,847), однако результат предварительный. Следующим шагом является проверка на публичных наборах данных о внутренних угрозах, в том числе CERT [5, 6], добавление последовательных моделей и оценка числа ложных срабатываний на реальных журналах.

СПИСОК ЛИТЕРАТУРЫ:

1. Scarfone K., Mell P. Guide to Intrusion Detection and Prevention Systems (IDPS): NIST Special Publication 800-94. Gaithersburg: National Institute of Standards and Technology, 2007. 127 p.

2. Strom B. E. et al. MITRE ATT&CK: Design and Philosophy. McLean: The MITRE Corporation, 2020.

3. Chandola V., Banerjee A., Kumar V. Anomaly detection: A survey // ACM Computing Surveys. 2009. Vol. 41, No. 3. Art. 15. 58 p. DOI: 10.1145/1541880.1541882.

4. Sommer R., Paxson V. Outside the closed world: On using machine learning for network intrusion detection // Proceedings of the IEEE Symposium on Security and Privacy. 2010. P. 305–316. DOI: 10.1109/SP.2010.25.

5. Tuor A., Kaplan S., Hutchinson B., Nichols N., Robinson S. Deep learning for unsupervised insider threat detection in structured cybersecurity data streams // Proceedings of the AAAI-17 Workshop on Artificial Intelligence for Cyber Security. 2017. P. 224–231. arXiv:1710.00811.

6. Glasser J., Lindauer B. Bridging the gap: A pragmatic approach to generating insider threat data // Proceedings of the IEEE Security and Privacy Workshops. 2013. P. 98–104. DOI: 10.1109/SPW.2013.37.

7. Du M., Li F., Zheng G., Srikumar V. DeepLog: Anomaly detection and diagnosis from system logs through deep learning // Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS). 2017. P. 1285–1298. DOI: 10.1145/3133956.3134015.

8. Pang G., Shen C., Cao L., van den Hengel A. Deep learning for anomaly detection: A review // ACM Computing Surveys. 2021. Vol. 54, No. 2. Art. 38. DOI: 10.1145/3439950.

9. Liu F. T., Ting K. M., Zhou Z.-H. Isolation forest // Proceedings of the 8th IEEE International Conference on Data Mining (ICDM). 2008. P. 413–422. DOI: 10.1109/ICDM.2008.17.

10. Hinton G. E., Salakhutdinov R. R. Reducing the dimensionality of data with neural networks // Science. 2006. Vol. 313, No. 5786. P. 504–507. DOI: 10.1126/science.1127647.

11. Pedregosa F. et al. Scikit-learn: Machine learning in Python // Journal of Machine Learning Research. 2011. Vol. 12. P. 2825–2830.

12. Saito T., Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets // PLoS ONE. 2015. Vol. 10, No. 3. Art. e0118432. DOI: 10.1371/journal.pone.0118432.

© Central Asian Scientific Journal, 2026