Галушко Мария Юрьевна
студентка, 4 курс компьютерной безопасности
Северо-Кавказского федерального университета
России, г. Ставрополь

РАЗРАБОТКА МЕТОДА ОБНАРУЖЕНИЯ АНОМАЛЬНЫХ ЗАПРОСОВ К СУБД НА ОСНОВЕ КОМБИНИРОВАННОГО ИСПОЛЬЗОВАНИЯ КЛАСТЕРИЗАЦИИ И АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ (АЛГОРИТМ LSTM)

Аннотация. Предложен метод обнаружения аномальных запросов к системам управления базами данных, объединяющий плотностную кластеризацию профилей SQL-активности и анализ последовательностей запросов нейронной сетью LSTM. В отличие от правил сигнатурного контроля и моделей, оценивающих запросы изолированно, метод учитывает одновременно синтаксические, статистические, контекстные и временные признаки. Сначала SQL-текст нормализуется, а запрос преобразуется в вектор характеристик с учетом пользователя, роли, времени выполнения, затронутых объектов и объема результата. DBSCAN выявляет редкие локальные профили, тогда как LSTM оценивает отклонение порядка запросов в пределах сессии. Частные оценки объединяются вероятностным правилом с настраиваемым коэффициентом доверия к кластерному каналу. На сформированном наборе из 6000 последовательностей предложенный вариант достиг F1-меры 88,50% и полноты 85,23%, превысив LSTM-модель без кластерного канала. Метод предназначен для работы как дополнительный уровень аудита СУБД и позволяет объяснить срабатывание через расстояние до опорного кластера и вклад временного отклонения.

Ключевые слова: система управления базами данных, SQL-запрос, обнаружение аномалий, кластеризация, DBSCAN, LSTM, машинное обучение, аудит, информационная безопасность.

Annotation: A method for detecting anomalous queries to database management systems is proposed. It combines density-based clustering of SQL activity profiles with LSTM-based analysis of query sequences. The method uses syntactic, statistical, contextual, and temporal features. SQL text is normalized and transformed

into a feature vector that reflects the user role, execution time, affected objects, and result volume. DBSCAN identifies locally rare profiles, while LSTM estimates deviations in the order of operations within a session. The two scores are combined using a probabilistic fusion rule. In an experiment with 6,000 generated SQL sessions, the combined method achieved an F1 score of 88.50% and recall of 85.23%, outperforming the standalone LSTM model while preserving its false-positive rate. The output includes the distance to the nearest reference cluster and the sequential anomaly score, which supports incident triage and interpretation.

Keywords: database management system, SQL query, anomaly detection, clustering, DBSCAN, LSTM, machine learning, audit, information security.

Введение

Современные информационные системы используют СУБД как основной компонент хранения и обработки критически важных сведений. На уровне базы данных сходятся запросы прикладных сервисов, административные операции, аналитические задания и действия конечных пользователей. Компрометация учетной записи, ошибка конфигурации, SQL-инъекция или злоупотребление легальными полномочиями могут проявляться не отдельной запрещенной командой, а нетипичной последовательностью формально допустимых операций. Поэтому контроль только прав доступа и известных сигнатур не обеспечивает обнаружение всех опасных сценариев.

Аномальным в настоящей работе считается запрос или фрагмент пользовательской сессии, статистические, структурные либо временные характеристики которого существенно отклоняются от профиля штатной активности для соответствующей роли и контекста. Такое определение охватывает как очевидные нарушения — появление редких операторов, обращение к закрытым таблицам, резкий рост объема результата, — так и сложные последовательности разведки, накопления данных и последующей выгрузки. Аномалия при этом не отождествляется с атакой: решение метода является основанием для дополнительной проверки, а не автоматическим доказательством злонамеренности.

Методы кластеризации хорошо выявляют изолированные группы и точки в многомерном пространстве признаков, однако при агрегировании сессии частично теряют порядок операций. Рекуррентные сети LSTM, напротив, способны моделировать зависимость между удаленными событиями, но качество их решения снижается при появлении редких режимов, слабо представленных в обучающей выборке. Объединение этих каналов позволяет использовать различную природу ошибок: плотностный алгоритм фиксирует пространственную редкость, а LSTM — нарушение ожидаемой последовательности.

Цель исследования состоит в повышении полноты обнаружения аномальных SQL-запросов при контролируемой доле ложных срабатываний за счет комбинированного использования кластеризации и LSTM. Для достижения цели решаются задачи формирования признакового описания, построения опорных кластеров нормальной активности, обучения последовательностной модели, разработки правила объединения оценок и экспериментального сравнения с базовыми подходами.

Научная новизна работы заключается в двухканальном представлении аномальности SQL-сессии и в правиле объединения оценок, при котором кластерная редкость усиливает вероятность, сформированную последовательностной моделью, но не заменяет ее. Практическая значимость определяется возможностью включения метода в подсистему аудита без изменения прикладного кода: источником данных служат журналы СУБД и сведения о контексте соединения.

1. Анализ методов обнаружения аномальной активности в СУБД

Традиционные средства защиты баз данных включают идентификацию, разграничение доступа, аудит, контроль целостности и сигнатурный анализ. Их достоинством являются предсказуемость и возможность связать срабатывание с конкретным правилом. Недостаток проявляется при обнаружении ранее неизвестных сценариев: отдельный запрос может удовлетворять политике доступа, но последовательность таких запросов не соответствовать рабочей

функции пользователя. Кроме того, жесткие пороги плохо переносят изменение нагрузки, появление новых отчетов и сезонные колебания.

Профильные методы строят модель нормальной активности по оператору SQL, множеству таблиц, типам условий, числу соединений, времени выполнения и объему результата. В работах по защите приложений показана полезность отпечатков SQL-запросов и транзакций [1–3]. Однако фиксированный шаблон чувствителен к легитимным изменениям прикладной логики. Если профиль строится только по частотам, то перестановка операций или распределенная во времени атака может остаться незамеченной.

Кластеризация не требует полной разметки атак и поэтому применима к журналам, где нормальные события существенно преобладают. Алгоритм DBSCAN формирует области высокой плотности и помечает шумом объекты, не имеющие достаточного числа соседей [4]. В отличие от k-средних, он не требует заранее задавать количество кластеров и способен описывать группы произвольной формы. Для SQL-профилей это важно, поскольку роли администратора, аналитика и прикладного сервиса образуют неодинаковые по размеру и плотности области. Ограничением DBSCAN остается чувствительность к масштабу признаков и параметрам окрестности.

К методам изоляции выбросов относится Isolation Forest, формирующий ансамбль случайных деревьев [5]. Аномальный объект обычно изолируется меньшим числом разбиений. Метод эффективен как базовый алгоритм для табличных признаков, но порядок запросов приходится заменять агрегатами — средними, максимумами и диапазонами, что уменьшает чувствительность к временным зависимостям.

LSTM была предложена для обучения длительным зависимостям при сохранении управляемого потока градиента [6]. В задачах анализа системных журналов последовательностные модели показали способность прогнозировать следующее событие и выявлять отклонение реального события от множества ожидаемых [7]. Перенос этой идеи на SQL-аудит требует учитывать не только токены текста, но также роль, объект данных и параметры выполнения.

В противном случае одинаковая команда SELECT будет интерпретироваться одинаково для администратора и пользователя с ограниченными функциями.

Исследования систем обнаружения вторжений в базы данных отмечают необходимость сочетания контекстных и поведенческих признаков [8–10]. Существуют подходы к обнаружению SQL-инъекций на основе нейронных сетей, однако они преимущественно решают задачу бинарной классификации строки запроса и не всегда выявляют злоупотребление легальными запросами. Перспективным является переход от единичной строки к сессии и объединение обучаемой модели с неконтролируемым выявлением локальной редкости.

Таким образом, выбранным аналогом служит LSTM-классификатор последовательности нормализованных запросов. Его основные недостатки: зависимость от представительности размеченной выборки, слабая интерпретируемость и риск пропуска новой аномалии, если ее временная структура близка к обученным сценариям. Предлагаемый метод дополняет аналог кластерным каналом, обучаемым преимущественно на нормальной активности, и формирует объясняющие признаки срабатывания.

Таблица 1 – Сравнение подходов к обнаружению аномальных SQL-запросов

Подход	Разметка	Учет порядка	Новые аномалии	Основное ограничение
Правила и сигнатуры	не требуется	нет	низкий	не выявляют неизвестный сценарий
DBSCAN	не требуется	только через агрегаты	высокий	зависимость от масштаба и ε
Isolation Forest	не требуется	только через агрегаты	средний	ограниченное описание контекста
LSTM	частичная/полная	да	средний	зависимость от обучающих последовательностей
Комбинированный метод	частичная	да	высокий	необходима настройка двух каналов

2. Постановка задачи и формирование признаков

Пусть журнал аудита представлен упорядоченным множеством событий $Q = \{q_1, q_2, \ldots, q_n\}$. Каждое событие содержит нормализованный SQL-текст, идентификатор сессии, роль, время, длительность выполнения, число затронутых строк, объем результата и перечень объектов данных. Для исключения

утечки чувствительной информации литералы заменяются маркерами типов, комментарии удаляются, регистр ключевых слов унифицируется, а списки однотипных значений сворачиваются.

После нормализации запрос q_i преобразуется в d-мерный вектор. В экспериментальной реализации использованы десять групп признаков: тип оператора, структурная сложность, число соединений, вложенность, отклонение роли, редкость таблиц, несоответствие приложению, длительность, объем результата и временное положение в сессии. Категориальные поля кодируются частотами либо компактными индексами, числовые признаки робастно масштабируются.

$$x_i = [x_i^1, x_i^2, \dots, x_i^d]^T, \quad x_i \in \mathbb{R}^d \tag{1}$$

Для числового признака используется медиана и межквартильный размах, что уменьшает влияние отдельных экстремальных значений в обучающем журнале:

$$\tilde{x}_i^j = (x_i^j - \mathrm{med}(x^j)) / (\mathrm{IQR}(x^j) + \delta) \tag{2}$$

Запросы группируются по идентификатору соединения и разбиваются на окна длины T. Вектор сессии X_s образуется как последовательность признаков, сохраняющая порядок операций:

$$X_s = [x_{s,1}, x_{s,2}, \dots, x_{s,}{}^T] \in \mathbb{R}^{Txd} \tag{3}$$

Для кластерного канала дополнительно строится агрегированное описание z_s, содержащее среднее, стандартное отклонение и диапазон каждого признака. Такое представление устойчиво к единичному шуму, но намеренно не заменяет последовательностную модель:

$$z_s = \mathrm{concat}(\mathrm{mean}(X_s), \mathrm{std}(X_s), \max(X_s) - \min(X_s)) \tag{4}$$

Требуется построить функцию $A(X_s) \in [0; 1]$, максимальную для аномальной сессии. Решение принимается при превышении порога τ. Оптимизация проводится по F1-мере на валидационной части при дополнительном

ограничении на долю ложных срабатываний. В рабочей системе порог может задаваться отдельно для ролей с различной критичностью.

3. Предлагаемый комбинированный метод

3.1. Кластерный канал

Опорное множество для кластеризации формируется из сессий, прошедших первичный контроль и не имеющих подтвержденных инцидентов. Желательно строить отдельные модели по крупным ролям или классам приложений; при малом числе событий контекст включается в общий вектор. После стандартизации агрегированных признаков запускается DBSCAN. Точка относится к ядру кластера, если в ε-окрестности находится не менее MinPts объектов.

$$N\varepsilon(z) = \{z_j : \rho(z, z_j) \leq \varepsilon\} \quad (5)$$

$$\text{core}(z) \Leftrightarrow |N\varepsilon(z)| \geq \text{MinPts} \quad (6)$$

При обработке новой сессии рассчитывается среднее расстояние до k ближайших нормальных соседей. Оно переводится в интервал [0; 1] относительно квантилей расстояний валидационного набора:

$$\text{Scl}(z) = \text{clip}((d_k(z) - q_{0.02}) / (q_{0.995} - q_{0.02}), 0, 1) \quad (7)$$

Значение Scl, близкое к единице, означает, что профиль находится в разреженной области. Вместе с оценкой сохраняются ближайший кластер, расстояние до него и признаки с наибольшим стандартизованным отклонением. Эти сведения используются аналитиком как объяснение кластерной части решения.

3.2. Последовательностный канал LSTM

Входом LSTM является окно X_s. На каждом шаге вычисляются входной, забывающий и выходной вентили, а также кандидат состояния. Обозначим через h_t скрытое состояние, через c_t — состояние памяти, через σ — логистическую функцию:

$$i_t = \sigma(W_i[x_t; h_{t-1}] + b_i) \quad (8)$$

$$f_t = \sigma(Wf[x_t; h_{t-1}] + bf) \quad (9)$$

$$o_t = \sigma(Wo[x_t; h_{t-1}] + bo) \quad (10)$$

$$g_t = \tanh(Wg[x_t; h_{t-1}] + bg) \quad (11)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (12)$$

$$h_t = o_t \odot \tanh(c_t) \quad (13)$$

Последнее скрытое состояние поступает в полносвязный выходной слой. Полученная вероятность интерпретируется как последовательностная оценка аномальности:

$$\mathrm{SLSTM}(X_s) = \sigma(w^T h^T + b) \quad (14)$$

Обучение выполняется по бинарной кросс-энтропии. Для уменьшения влияния дисбаланса вес аномального класса задается обратно пропорционально его доле. В производственной эксплуатации модель может дополнительно обучаться как предиктор следующего типа запроса на большом объеме неразмеченных нормальных данных, после чего дообучаться на подтвержденных инцидентах.

3.3. Объединение оценок и алгоритм работы

Каналы объединяются по вероятностному правилу «мягкого ИЛИ». В отличие от простого среднего, высокая оценка одного канала не подавляется низким значением другого:

$$S(X_s) = 1 - (1 - \mathrm{SLSTM}(X_s)) \cdot (1 - \alpha \cdot \mathrm{Scl}(z_s)) \quad (15)$$

Коэффициент $\alpha \in [0; 1]$ определяет максимальный вклад кластерной редкости. В эксперименте значение $\alpha = 0{,}60$ выбрано по валидационной выборке. Итоговое решение задается пороговым правилом:

$$\hat{y}_s = 1, \text{если } S(X_s) \geq \tau; \quad \hat{y}_s = 0, \text{если } S(X_s) < \tau \quad (16)$$

Общая структура обработки показана на рисунке 1. В журнал событий возвращаются итоговая оценка, оценки каналов, порог, ближайший кластер и

основные отклонения. Это позволяет отличить срабатывание по редкому составу запросов от нарушения последовательности.

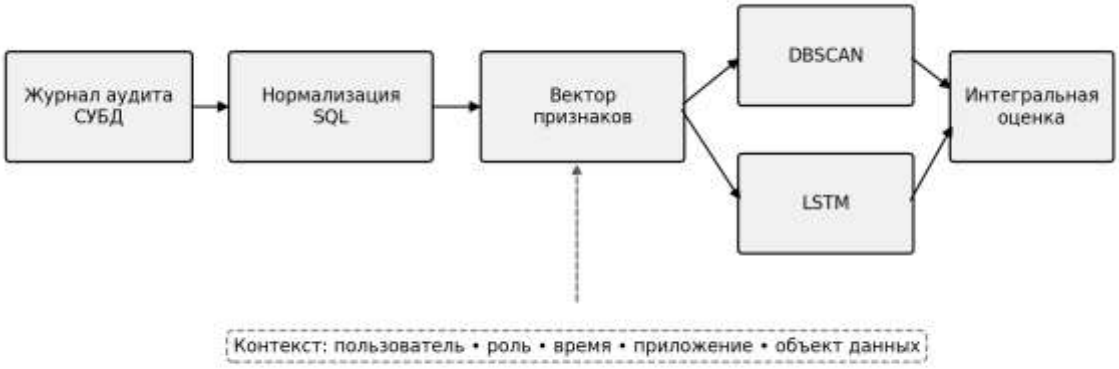

Рисунок 1 – Структура комбинированного метода обнаружения аномальных запросов

Таблица 2 – Последовательность этапов предлагаемого метода

Этап	Вход	Операция	Результат
1	запись аудита	очистка, параметризация, выделение контекста	нормализованный запрос
2	запрос и контекст	формирование десяти групп признаков	вектор x_i
3	окно сессии	агрегирование и поиск соседей	оценка Scl
4	последовательность X_s	прямой проход LSTM	оценка SLSTM
5	две оценки	объединение по формуле (15)	интегральная оценка S
6	S и τ	пороговое решение и объяснение	событие контроля/пропуск

4. Экспериментальное исследование

4.1. Набор данных и параметры эксперимента

Для проверки работоспособности реализован программный прототип на Python с использованием NumPy и scikit-learn. Вычисления выполнялись на процессоре общего назначения без аппаратного ускорения. Набор данных формировался генератором ролевых SQL-сессий, что позволило контролировать тип и момент появления отклонения и не использовать конфиденциальные журналы организации.

Сформировано 6000 сессий длиной 12 запросов: 5000 штатных и 1000 аномальных. Нормальные последовательности моделировали четыре роли с различными частотами чтения, изменения данных, соединений и объемов результата. В часть штатных сессий добавлены редкие легитимные операции обслуживания, повышающие сложность разделения. Аномальный класс включал SQL-инъекционно-подобные структурные всплески, несоответствие полномочий, эксфильтрацию и нарушение порядка формально допустимых операций.

Данные случайно разделены на обучающую, валидационную и тестовую части в отношении 70:15:15. Масштабирующие параметры, кластеры и нейронная сеть оценивались только по обучающей части; пороги и коэффициент объединения выбирались по валидационной. Тестовая часть использовалась однократно для итогового сравнения. Такая схема исключает прямую подстройку порога под приведенные результаты.

Таблица 3 – Основные параметры вычислительного эксперимента

Параметр	Значение
Количество сессий	6000
Нормальные / аномальные	5000 / 1000
Длина окна T	12 запросов
Размерность запроса d	10 групп признаков
Разделение	70% / 15% / 15%
Скрытые элементы LSTM	18
Число эпох	28
Размер мини-пакета	96
α / τ	0,60 / 0,70

4.2. Метрики качества

Из-за преобладания нормальных сессий обычная доля правильных ответов недостаточно информативна. Использованы точность P, полнота R, F1-мера, доля ложноположительных решений FPR и площадь под ROC-кривой AUC. Через TP, FP, TN и FN обозначены элементы матрицы ошибок:

$$P = TP / (TP + FP) \quad (17)$$

$$R = TP / (TP + FN) \quad (18)$$

$$F1 = 2PR / (P + R) \quad (19)$$

$$FPR = FP / (FP + TN) \quad (20)$$

Точность показывает долю подтвержденных аномалий среди срабатываний; полнота — долю обнаруженных аномалий среди всех аномальных сессий. Для подсистемы мониторинга критична полнота, но ее повышение не должно достигаться неконтролируемым ростом ложных тревог, поэтому основным сводным показателем выбрана F1-мера.

4.3. Результаты сравнения

Таблица 4 – Результаты обнаружения аномальных SQL-сессий, %

Метод	P	R	F1	FPR	AUC
Правила	67.65	15.44	25.14	1.46	0.651
DBSCAN	95.05	64.43	76.80	0.67	0.866
Isolation Forest	91.26	63.09	74.60	1.20	0.879
LSTM	91.85	83.22	87.32	1.46	0.991
Предлагаемый метод	92.03	85.23	88.50	1.46	0.992

Правила показали высокую интерпретируемость, но обнаружили только 15,44% аномалий: временное нарушение из допустимых операций не содержит обязательного сигнатурного признака. DBSCAN и Isolation Forest повысили полноту до 64,43 и 63,09% соответственно. При этом DBSCAN обеспечил наиболее высокую точность среди неконтролируемых методов — 95,05%, что подтверждает пригодность плотностной модели для выделения редких профилей.

Одиночная LSTM увеличила полноту до 83,22% и F1-меру до 87,32%, поскольку учитывала направление и порядок изменения признаков. Комбинированный метод достиг полноты 85,23% и F1-меры 88,50%. Прирост F1 относительно LSTM составил 1,18 процентного пункта, а число пропусков уменьшилось без увеличения FPR: у обеих моделей значение равно 1,46%. AUC выросла с 0,991 до 0,992.

Полученный прирост нельзя считать универсальной оценкой для любой СУБД: он относится к описанному генератору и фиксированной схеме разделения. Однако эксперимент подтверждает основную гипотезу — кластерная

оценка содержит дополнительную информацию и способна скорректировать часть неопределенных решений LSTM. При переносе в организацию требуется повторное обучение на локальных журналах и проверка на временно отделенной выборке.

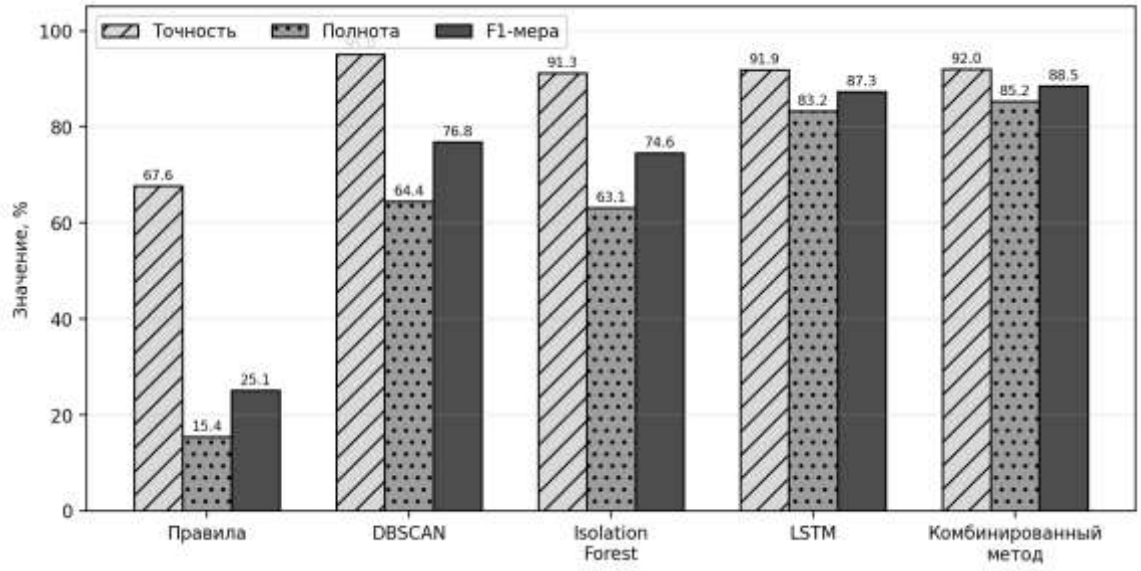

Рисунок 2 – Сравнение точности, полноты и F1-меры исследованных методов

4.4. Анализ типов аномалий

Полнота комбинированного метода по отдельным сценариям составила 96,77% для SQL-инъекционно-подобных запросов, 81,40% для несоответствия полномочий, 96,88% для эксфильтрации и 72,09% для нарушения порядка. Первые и третьи сценарии одновременно изменяют структуру запроса и агрегаты сессии, поэтому надежно фиксируются обоими каналами. Нарушение порядка сохраняет близкие средние и диапазоны признаков и в основном распознается LSTM; более низкое значение указывает на необходимость увеличения длины окна и числа подобных примеров.

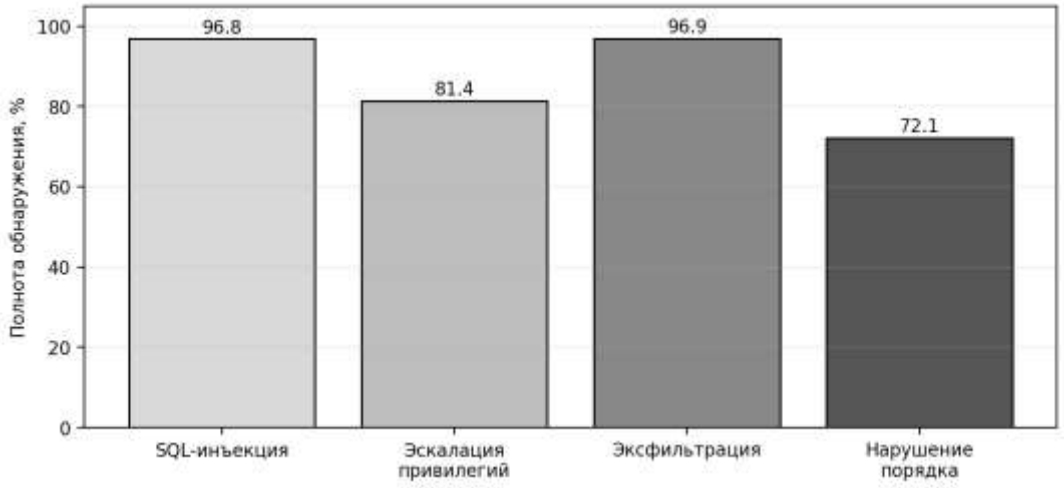

Рисунок 3 – Полнота предлагаемого метода для различных типов аномалий

5. Обсуждение и практическое применение

Предлагаемый метод целесообразно размещать между подсистемой аудита и центром мониторинга информационной безопасности. Он не должен блокировать запросы непосредственно в транзакционном контуре без дополнительной политики, поскольку статистическая аномалия может соответствовать срочной легитимной операции. Для низкого и среднего риска достаточно регистрации и накопления контекста; для высокого риска возможны запрос повторной аутентификации, ограничение объема результата или передача события аналитику.

Первоначальное обучение следует проводить на периоде, включающем рабочие и выходные дни, регулярную отчетность и плановые задания. Журналы с известными инцидентами исключаются из опорного множества DBSCAN либо помечаются. Переобучение выполняется по скользящему окну после подтверждения новых нормальных режимов. Резкое автоматическое включение всех последних событий опасно, поскольку атакующий способен постепенно сместить профиль.

Интерпретируемость обеспечивается разложением решения на Scl и SLSTM. Для кластерного канала выводятся расстояние до ближайшего профиля, роль и отклоняющиеся признаки. Для LSTM можно применять последовательное исключение событий из окна: изменение оценки после маскирования показывает, какие запросы сильнее всего повлияли на решение. Такое

объяснение не заменяет формальную экспертизу, но сокращает время первичного анализа.

Вычислительная сложность обучения DBSCAN в худшем случае квадратична по числу объектов, поэтому для больших архивов следует использовать выборку, индекс ближайших соседей или мини-пакетное обновление опорного множества. Онлайн-обработка одного окна включает формирование признаков, поиск k соседей и прямой проход LSTM. Эти операции допускают параллельное выполнение, а итоговое правило имеет постоянную сложность.

К ограничениям исследования относятся синтетический характер данных, фиксированная длина окна и использование бинарной разметки. Реальные журналы содержат пропуски, разные диалекты SQL, служебные запросы драйверов и изменения схемы. Кроме того, подтверждение внутренних злоупотреблений встречается редко, поэтому оценка должна дополняться сценариями красной команды и длительным теневым запуском. Дальнейшая работа предусматривает исследование адаптивного коэффициента α, обучение без разметки, учет графа зависимостей таблиц и проверку устойчивости к постепенному дрейфу поведения.

Заключение

Разработан метод обнаружения аномальных запросов к СУБД, объединяющий плотностную кластеризацию агрегированных профилей и LSTM-анализ последовательностей. Сформировано признаковое представление SQL-события, определены правила построения сессионного окна, кластерной и последовательностной оценок, предложено вероятностное объединение результатов и пороговое решение.

Вычислительный эксперимент на 6000 сессиях показал, что комбинированный вариант достиг F1-меры 88,50% и полноты 85,23% при FPR 1,46%. По сравнению с одиночной LSTM F1 увеличилась на 1,18 процентного пункта, а полнота — на 2,01 пункта. Результаты подтверждают целесообразность совместного учета пространственной редкости и временного контекста. Метод может использоваться как дополнительный уровень аудита и ранжирования

событий, а перед промышленным внедрением должен быть переобучен и проверен на локальных журналах конкретной СУБД.

Использованные источники:

1. Rietta F. S. Application layer intrusion detection for SQL injection // Proceedings of the 44th Annual Southeast Regional Conference. 2006. P. 531–536. DOI: 10.1145/1185448.1185564.

2. Sallam A., Bertino E. DetAnom: Detecting anomalous database transactions by insiders // Proceedings of the 5th ACM Conference on Data and Application Security and Privacy. 2015. P. 25–35. DOI: 10.1145/2699026.2699111.

3. Darwish S. M. Machine learning approach to detect intruders in database based on hexplet data structure // Journal of King Saud University — Computer and Information Sciences. 2016. Vol. 28, No. 4. P. 465–477.

4. Ester M., Kriegel H.-P., Sander J., Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise // Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. 1996. P. 226–231.

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

6. Hochreiter S., Schmidhuber J. Long short-term memory // Neural Computation. 1997. Vol. 9, No. 8. P. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

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

8. Santos R. J., Bernardino J., Vieira M. Approaches and challenges in database intrusion detection // ACM SIGMOD Record. 2014. Vol. 43, No. 3. P. 36–47. DOI: 10.1145/2694428.2694435.

9. Bertino E., Kamra A., Terzi E., Vakali A. Intrusion detection in RBAC-administered databases // Proceedings of the 21st Annual Computer Security Applications Conference. 2005. P. 170–182. DOI: 10.1109/CSAC.2005.22.

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

11. Breunig M. M., Kriegel H.-P., Ng R. T., Sander J. LOF: Identifying density-based local outliers // Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. 2000. P. 93–104. DOI: 10.1145/342009.335388.

12. Schubert E., Sander J., Ester M., Kriegel H.-P., Xu X. DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN // ACM Transactions on Database Systems. 2017. Vol. 42, No. 3. Article 19. DOI: 10.1145/3068335.

13. Aggarwal C. C. Outlier Analysis. 2nd ed. Cham: Springer, 2017. 466 p. DOI: 10.1007/978-3-319-47578-3.

14. Naserinia V. Anomaly Detection in a SQL Database: Master's thesis. Stockholm: KTH Royal Institute of Technology, 2022. 84 p.

15. Shlezinger M., Akirav S., Zhou L. et al. Leveraging large language models for SQL behavior-based database intrusion detection. 2025. arXiv:2508.05690.