ОШ МАМЛЕКЕТТИК УНИВЕРСИТЕТИНИН ЖАРЧЫСЫ

ВЕСТНИК ОШСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

BULLETIN OF OSH STATE UNIVERSITY

ISSN 1694-7452 e-ISSN: 1694-8610

№2/2026, 327-340

ИНФОРМАТИКА

УДК: 004.056.5:004.75

DOI: 10.52754/16948610_2026_2_24

ИСПОЛЬЗОВАНИЕ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ВЫЯВЛЕНИЯ АНОМАЛИЙ В ТРАФИКЕ ВЕБ-СЕРВЕРОВ

ВЕБ-СЕРВЕРЛЕРДИН ТРАФИГИНДЕГИ АНОМАЛИЯЛАРДЫ АНЫКТОО ҮЧҮН МАШИННЫК ОКУТУУНУ КОЛДОНУУ

USING MACHINE LEARNING TO DETECT ANOMALIES IN WEB SERVER TRAFFIC

Абдумиталип уулу Кубатбек
Абдумиталип уулу Кубатбек
Abdumitalip uulu Kubatbek

к.ф.-м.н., доцент, Ошский государственный университет
ф.-м.и.к., доцент, Ош мамлекеттик университети
Candidate of Physico-Mathematical Sciences, Associate Professor, Osh State University
kuba@oshsu.kg
ORCID: 0009-0000-5208-0741

Омаралиева Гулбайра Абдималиковна
Омаралиева Гулбайра Абдималиковна
Omaralieva Gulbaira Abdimalikovna

ф.-м.н., доцент, Ошский государственный университет
ф.-м.и.к., доцент, Ош мамлекеттик университети
Candidate of Physico-Mathematical Sciences, Associate Professor, Osh State University
gulya@oshsu.kg
ORCID: 0009-0004-7806-3690

Исакова Акмарал Камаридиновна
Исакова Акмарал Камаридиновна
Isakova Akmaral Kamaridinovna

магистрант, Ошский государственный университет
магистрант, Ош мамлекеттик университети
master's students, Osh State University
isakovaakmaral3@gmail.com
ORCID: 0009-0000-9444-163X

Замирбек кызы Ыкыбал
Замирбек кызы Ыкыбал
Zamirbek kyzy Ykybal

магистрант, Ошский государственный университет
магистрант, Ош мамлекеттик университети
master's students, Osh State University
ykybalzamirbek@gmail.com
ORCID: 0009-0008-8542-801X

ИСПОЛЬЗОВАНИЕ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ВЫЯВЛЕНИЯ АНОМАЛИЙ В ТРАФИКЕ ВЕБ-СЕРВЕРОВ

Аннотация

Актуальность. Работа посвящена разработке и реплицируемой апробации методологии обнаружения аномалий в трафике веб-серверов на основе алгоритмов машинного обучения и журналов прикладного уровня (HTTP). Предлагается практико-ориентированный конвейер: сбор событий веб-сервера и IDS, нормализация и обогащение, построение признаков (структурные, содержательные, энтропийные, поведенческие), обучение ансамблей без учителя (Isolation Forest, One-Class SVM) и глубоких автоэнкодеров (в т.ч. LSTM-варианты) с последующим объяснением выявленных отклонений. Демонстрационные результаты представлены на общедоступных наборах (CSIC-2010 для HTTP-запросов, CIC-IDS2017/CSE-CIC-IDS2018 для сетевых потоков) и синтетических журналах, имитирующих нагрузочные профили университетского веб-контура; даются количественные метрики качества (AUROC/AUPRC, F1@FPR≤1%), оценка производительности и рекомендации по внедрению в инфраструктуре дата-центра ОшГУ. Этическая оговорка: статья предоставляет реплицируемый протокол и демонстрационную оценку; мы не утверждаем, что обрабатывали приватные продукционные логи в рамках этой публикации.

Ключевые слова: обнаружение аномалий; веб-трафик; HTTP; машинное обучение; Isolation Forest; One-Class SVM; автоэнкодер; ELK; syslog.

Веб-серверлердин трафигиндеги аномалияларды аныктоо үчүн машиннык окутууну колдонуу

Аннотация

Маанилүүлүк. Бул макалада веб-сервер трафигиндеги аномалияларды машиналык үйрөнүү алгоритмдерине жана тиркеме деңгээлиндеги журналдарга (HTTP) негизделген аныктоо методологиясын иштеп чыгуу жана кайталап сыноо баяндалат. Практикага багытталган түтүк сунушталат: веб-сервер жана IDS окуяларын чогултуу, нормалдаштыруу жана байытуу, функцияларды түзүү (структуралык, мазмундук, энтропия, жүрүм-турумдук), көзөмөлсүз ансамблдик окутуу (изоляция токою, бир класстагы SVM) жана терең автокодерлер (LSTM варианттарын кошо алганда) аныкталган четтөөлөрдү кийинчерээк түшүндүрүү менен. Демонстрациялык жыйынтыктар жалпыга жеткиликтүү маалымат топтомдорунда (HTTP суроо-талаптары үчүн CSIC-2010, тармактык агымдары үчүн CIC-IDS2017/CSE-CIC-IDS2018) жана университеттин веб схемасынын жүктөө профилдерин симуляциялаган синтетикалык журналдарда көрсөтүлгөн; Сандык сапат метрикалары (AUROC/AUPRC, F1@FPR≤1%), иштин натыйжалуулугун баалоо жана Ош мамлекеттик университетинин маалымат борборунун инфраструктурасында ишке ашыруу боюнча сунуштар берилген. Этика боюнча эскертүү: Бул макала кайталап протокол жана демонстрациялык баалоону берет; бул басылмада биз жеке өндүрүш журналдарын иштеттик деп ырастабайбыз.

Ачкыч сөздөр: аномалияны аныктоо; веб-трафик; HTTP; машиналык окутуу; изоляция токою; бир класстагы svm; автокодер; ELK; syslog.

Using machine learning to detect anomalies in web server traffic

Abstract

Relevance. We present a reproducible methodology for detecting anomalies in web-server traffic based on machine learning over application-layer logs. The pipeline covers elastic log collection (Nginx/Filebeat), feature engineering (structural, content-based, entropy, behavioral), unsupervised models (Isolation Forest, One-Class SVM) and deep autoencoders (including LSTM variants), plus post-hoc explainability. Demonstration results are reported on public datasets (CSIC-2010 HTTP requests, CIC-IDS2017 and CSE-CIC-IDS2018 network flows) and synthetic logs emulating a university web perimeter; we provide standard metrics (AUROC/AUPRC, F1 at fixed FPR), runtime profiling, and integration guidance for Osh State University’s data-center.

Keywords: anomaly detection; HTTP; web logs; ELK; Isolation Forest; One-Class SVM; autoencoder.

Введение

Высокая доля обращений к университетским информационным ресурсам проходит через веб-сервисы: сайты подразделений, личные кабинеты, системы дистанционного обучения, электронные журналы, сервисы приема документов и внутренние административные порталы. Для такого контура типично не только увеличение суммарной нагрузки, но и рост разнообразия сценариев доступа. В периоды приемной кампании, сессии, публикации расписаний и выдачи справок структура запросов резко меняется, поэтому традиционные сигнатурные средства защиты не всегда позволяют своевременно отделить штатный всплеск активности от подготовки атаки, автоматизированного перебора параметров или медленного сканирования приложений.

Сигнатурные WAF и IDS остаются необходимым уровнем защиты, однако они по определению ориентированы на уже известные шаблоны вредоносного поведения. В реальной эксплуатац ии администратор сталкивается с промежуточными случаями: слегка модифицированными инъекциями, нетипичными комбинациями параметров, автоматическими запросами, имитирующими браузер, и аномальными последовательностями действий, которые по отдельности выглядят допустимо. Поэтому задача обнаружения аномалий в веб-трафике состоит не в замене сигнатурного контроля, а в построении дополнительного слоя аналитики, способного описывать нормальный профиль работы сервиса и сигнализировать о статистически редких, структурно подозрительных или поведенчески нетипичных запросах (Moradi Vartouni, 2019).

Особую ценность для такой аналитики имеют журналы прикладного уровня HTTP. В отличие от агрегированных сетевых потоков, они сохраняют семантику взаимодействия с веб-приложением: путь, метод, код ответа, параметры, длину полезной нагрузки, признаки кодирования, заголовки, а при корректной настройке - и сведения о пользовательском агенте, времени ответа и проксируемых адресах. Именно на этом уровне становятся заметны редкие URI, нехарактерная энтропия параметров, неестественная повторяемость запросов и иные признаки обхода валидации. Одновременно такие данные требуют аккуратной нормализации, поскольку в них много шумов, пропусков, сезонных колебаний и доменно-зависимых особенностей.

Цель настоящей статьи состоит в том, чтобы предложить для условий ОшГУ воспроизводимый и практически применимый подход к обнаружению аномалий в трафике веб-серверов, основанный на сочетании инженерии признаков и алгоритмов машинного обучения без учителя. В работе сохраняется исходный смысл исследования: мы рассматриваем не продукционные приватные журналы как объект публикации, а реплицируемый протокол, который может быть перенесен в инфраструктуру университета. Основное внимание уделяется тому, какие данные следует собирать, как формировать устойчивое представление о норме, какие модели целесообразно комбинировать и каким образом интерпретировать результаты так, чтобы они были полезны не только исследователю, но и администратору информационной безопасности.

Обзор литературы

Данная область развивалась от относительно простых моделей нормальности HTTP-запросов к гибридным схемам, объединяющим структурные, содержательные и поведенческие признаки. Классические прикладные исследования были сосредоточены на том, чтобы формализовать корректный профиль веб-приложения и затем сравнивать с ним

поступающие запросы. (Torrano-Gimenez, 2010). Нормальное поведение приложения описывалось с помощью статистически сформированного профиля, а отклонения интерпретировались как потенциальные атаки прикладного уровня. Подобные подходы важны тем, что они задали логику перехода от жестких сигнатур к моделированию «нормы» самого сервиса.

Следующий значимый пласт исследований связан с признаковым описанием HTTP-содержимого. Для задач обнаружения SQL-инъекций, XSS, обхода валидации и параметрического шума хорошо зарекомендовали себя n-граммы, частоты токенов, длины строк, доля специальных символов и различные метрики сложности запроса. Сильная сторона таких представлений состоит в их относительной простоте и интерпретируемости: аномальность можно объяснить всплеском редких последовательностей символов, нетипичной длиной параметров или ростом энтропии. Однако у чисто содержательных методов есть и ограничение: они лучше работают на уровне отдельного запроса и слабее описывают длинные пользовательские сессии, распределенные по времени (Chua, 2024).

Для анализа табличных признаков устойчивыми базовыми моделями остаются One-Class SVM и Isolation Forest. One-Class SVM полезен там, где после масштабирования признаки образуют сравнительно компактную область нормальности, а задача сводится к отделению ее границы от редких отклонений. Isolation Forest, напротив, хорошо адаптирован к разнородным данным и не требует явного предположения о форме распределения. Именно поэтому он часто используется как стартовая модель в прикладных сценариях, где необходимо быстро получить практичный baseline (Moradi Vartouni, 2019).

В более поздних работах эта модель рассматривается не только как самостоятельный детектор, но и как часть ансамблей, дополняемых последовательностными или кластеризационными процедурами.

Современный этап развития темы характеризуется ростом интереса к глубоким автоэнкодерам и гибридным архитектурам. Автоэнкодер позволяет учить сжатое представление нормальных событий и оценивать аномальность через ошибку реконструкции; последовательностные варианты на базе LSTM особенно полезны для моделирования сессий, в которых подозрительным является не отдельный запрос, а их порядок и контекст. В исследовании T. Smolen и L. Benova показано, что Isolation Forest и автоэнкодер делают акцент на разных сторонах данных: первый лучше использует числовые признаки, тогда как второй способен улавливать более тонкие структурные отклонения и пропуски в паттернах поведения. Работа L. Benova и L. Hudec дополняет этот вывод, подчеркивая важность анализа пользовательской активности в журналах NGINX и последующей экспертной интерпретации выделенных аномалий.

Отдельного внимания заслуживает проблема объяснимости. Для продуктивной эксплуатации мало получить высокий AUROC; администратору необходимо понимать, почему событие признано подозрительным, какие признаки дали основной вклад в скор и чем данный случай отличается от типичных обращений к ресурсу. Поэтому в новейших публикациях все чаще сочетаются детекторы аномалий, кластеризация, визуальная аналитика и экспертная оценка. Несмотря на заметный прогресс, в литературе по-прежнему ощущается нехватка работ, где описан именно перенос модели в реальный веб-контур организации с ограниченными ресурсами и сезонной нагрузкой. Настоящая статья закрывает этот практический пробел: она объединяет идеи из исследований по HTTP-содержимому,

потоковым признакам и объяснимой аналитике в едином воспроизводимом протоколе, ориентированном на университетскую инфраструктуру.

Материалы и методы

Материальная основа исследования строится на журналах прикладного и сетевого уровня, которые могут быть собраны стандартными средствами наблюдаемости без внедрения специализированного программного обеспечения в само веб-приложение. В минимальной конфигурации достаточно корректно настроенных журналов Nginx access и error, дополненных метаданными о проксируемом клиентском адресе, пользовательском агенте, коде ответа, размере ответа и времени обработки запроса. Для повышения контекстности эти данные целесообразно обогащать событиями IDS, например Suricata в формате eve.json, где фиксируются категории срабатываний, сигнатуры, сетевые адреса и дополнительные признаки полезной нагрузки. Такой двухуровневый сбор позволяет сопоставлять статистически редкие HTTP-события с уже известными индикаторами угроз.

При переносе методологии в инфраструктуру дата-центра ОшГУ важна не только полнота логирования, но и дисциплина нормализации. Журналы разных узлов должны быть синхронизированы по времени, приведены к единой схеме именования полей и очищены от неоднозначных представлений одного и того же события. На практике это означает нормализацию путей, унификацию кодировок, выделение параметров запроса, канонизацию HTTP-методов, контроль пропусков и отделение технического шума от содержательных признаков поведения пользователя. В демонстрационной части работы для воспроизводимости использовались открытые наборы CSIC-2010, CIC-IDS2017 и CSE-CIC-IDS2018, а также синтетические журналы, имитирующие характерные для университетского веб-контура паттерны нагрузки.

Конструирование признаков опирается на четыре взаимодополняющих слоя. Структурные признаки описывают геометрию запроса: длину URI, глубину пути, число параметров, тип метода, код ответа, размер полезной нагрузки, соотношение буквенно-цифровых и специальных символов, наличие длинных токенов или многократного кодирования. Содержательные признаки отражают состав строки запроса и параметров через n-граммы, токеновые частоты и индикаторы характерных последовательностей. Энтропийные признаки фиксируют степень случайности и потенциальной обфускации, что особенно полезно для обнаружения закодированных фрагментов, необычных идентификаторов и вредоносных вставок. Поведенческий слой описывает интенсивность и траекторию доступа: частоту запросов в окнах времени, редкость пути для данного ресурса, межприходные интервалы, разнообразие статусов и устойчивость пользовательского агента.

Принципиально важно, что перечисленные признаки не следует рассматривать изолированно. Одна и та же длина параметра может быть нормальной для учебной платформы, но подозрительной для новостного портала; повышенная энтропия допустима для некоторых служебных токенов, но аномальна в обычных GET-запросах. Поэтому при построении витрины признаков учитывается контекст ресурса: принадлежность к группе URL, тип взаимодействия, сезонный профиль нагрузки и характер клиентского приложения. Для университетской среды это особенно существенно, поскольку рядом сосуществуют публичные сайты, личные кабинеты, API внутренних сервисов и административные интерфейсы, формирующие разные эталоны «нормы».

На уровне моделирования использована комбинированная схема, поскольку разные классы отклонений проявляются в данных по-разному. Isolation Forest выбран в качестве

базового оценщика для табличных смешанных признаков благодаря устойчивости к разнородным распределениям и способности выделять редкие наблюдения без построения явной плотностной модели. One-Class SVM применялся как дополнительный инструмент там, где после масштабирования формировалась сравнительно компактная область нормального поведения. Такое сочетание полезно по практическим причинам: если оба алгоритма независимо сигнализируют об отклонении, доверие к срабатыванию возрастает; если же их оценки расходятся, аналитик получает повод проверить, не обусловлен ли инцидент особенностями признакового пространства.

Для более сложных случаев, когда аномальность скрыта в последовательности действий или в форме строкового представления запроса, использованы автоэнкодеры и их вариации. Обычный MLP-автоэнкодер эффективен на сжатом векторном описании HTTP-события и позволяет ранжировать наблюдения по ошибке реконструкции. LSTM-AE и biLSTM-AE ориентированы на анализ сессий или коротких окон последовательных запросов, что важно при медленном переборе параметров, распределенном сканировании и иных сценариях, где отдельный запрос может не выбиваться из нормы, но их порядок образует нетипичную картину. Вариант DAGMM использован как гибрид, объединяющий реконструкцию и оценку плотности в латентном пространстве.

Процесс обучения организован так, чтобы минимизировать утечку информации о тестовых аномалиях в этап построения нормы. Для моделей без учителя выделялись интервалы, интерпретируемые как преимущественно «чистые» окна нормальной активности. Далее выполнялись масштабирование числовых признаков, кодирование категориальных полей и отбор признаков, наиболее устойчивых к сезонным колебаниям. Значения contamination, параметр ν в One-Class SVM, размер бутылочного горлышка автоэнкодера, длина последовательности и другие гиперпараметры подбирались не по максимальной формальной точности любой ценой, а по компромиссу между полнотой обнаружения и эксплуатационной стоимостью ложных срабатываний.

Порог аномальности определялся не как произвольное фиксированное число, а как управляемый операционный параметр. Для веб-сервисов учебного заведения слишком агрессивный порог опасен, поскольку в периоды массовых обращений он создаст поток ложных сигналов и быстро приведет к «замыливанию» внимания операторов. Поэтому калибровка выполнялась по AUPRC и F1 при ограничении FPR на уровне, приемлемом для службы сопровождения. Такой подход делает модель не просто академически успешной, а пригодной для повседневного использования в SOC-подобном процессе или в небольшом центре мониторинга университета.

Отдельным блоком методики является объяснение результатов. Для табличных моделей применима оценка вклада признаков и анализ их локальных отклонений от типичного профиля ресурса. Для автоэнкодеров интерпретация строится через карту ошибок реконструкции по токенам и сегментам запроса. Это позволяет отличать, например, подозрительное кодирование параметра от простой аномалии по частоте доступа. В прикладном плане объяснимость нужна не только для отчета, но и для последующего действия: уточнения правил WAF, блокировки адреса, ограничения частоты запросов, добавления новых индикаторов компрометации или, напротив, корректировки модели на легитимный, но ранее не наблюдавшийся сценарий работы.

Таблица 1. Системный контур и каналы данных

Слой	Компонент	Формат/модуль	Назначение
Веб	Nginx	Filebeat nginx module	Парсинг access/error логов, извлечение полей.
IDS	Suricata	eve.json (HTTP/alert)	Доп. контекст (тип сигнатуры, категория).
Транспорт	Syslog	RFC-3164/5424	Централизация и маршрутизация событий.
Хранилище	Elastic	Index + ILM	Индексация, ретеншн, витрины фичей.

Таблица 2. Репертуар признаков и интерпретация

Группа	Пример признака	Интерпретация
Структурные	len(query), path_depth, %nonalpha	Длины/структура и избыток спецсимволов — индикаторы инъекций.
N-gram	tf-idf(n=3..5) по параметрам	Сдвиг распределений от «нормы» ресурса.
Энтропийные	H(query), H(params)	Высокая энтропия — признак обфускации/вредоносной нагрузки.
Поведенческие	req_rate_1m, rare_path_score	Всплески активности и «редкие» траектории.

Экспериментальный дизайн и протокол переноса в ДЦ ОшГУ

Экспериментальный протокол был построен как последовательность этапов, которые при необходимости могут быть воспроизведены в тестовой или продуктивной среде без изменения общей архитектуры. На первом этапе журналы Nginx собирались через модуль Filebeat nginx с типовыми ingest-пайплайнами, а при наличии дополнительного сетевого контроля к ним присоединялись события Suricata в формате eve.json. Затем данные доставлялись в стек Elastic, где выполнялись разбор полей, контроль временных меток, первичная фильтрация шума и формирование витрины признаков для дальнейшей аналитики. Такой конвейер удобен тем, что отделяет инфраструктурную часть от собственно машинного обучения и облегчает поэтапное внедрение.

Ключевым элементом протокола является построение эталона нормального поведения. В прикладной среде его нельзя формировать на слишком коротком интервале, иначе модель запомнит случайный фрагмент нагрузки и станет ошибочно считать аномалией естественные колебания. В данной работе в качестве ориентировочного окна использовался период 7-14 дней с обязательной проверкой того, чтобы в него попали разные режимы работы сервиса: учебные дни, менее нагруженные часы и интервалы планового обслуживания. Для открытых наборов данных дополнительно применялась временная или логически раздельная валидация, чтобы избежать неоправданно оптимистичных оценок качества.

После построения витрины признаков выполнялись масштабирование, обучение базовых и глубоких моделей, а затем калибровка порогов по заранее выбранным операционным сценариям. Отдельно оценивались не только AUROC и AUPRC, но и поведение модели при фиксированном уровне ложноположительных срабатываний, среднее время инференса и устойчивость к доменному сдвигу между источниками данных. Такой набор критериев важен, поскольку высокое качество на лабораторном датасете само по себе не гарантирует применимость в реальном контуре с сезонностью, техническими окнами, прокси-узлами и смешением пользовательских ролей.

Наконец, в протокол сознательно включен шаг локальной переоценки при переносе в дата-центр ОшГУ. Поскольку публикация опирается на публичные наборы и синтетические журналы, она не делает чрезмерных заявлений о фактическом уровне риска на продукционных ресурсах университета. Вместо этого предлагается воспроизводимая схема: собрать локальные распределения признаков, повторно откалибровать пороги, оценить чувствительность к редким легитимным сценариям и лишь затем переводить детекторы в режим рабочих оповещений. Такой подход уменьшает риск как переобучения на демонстрационных данных, так и преждевременного внедрения модели с неконтролируемой долей ложных тревог.

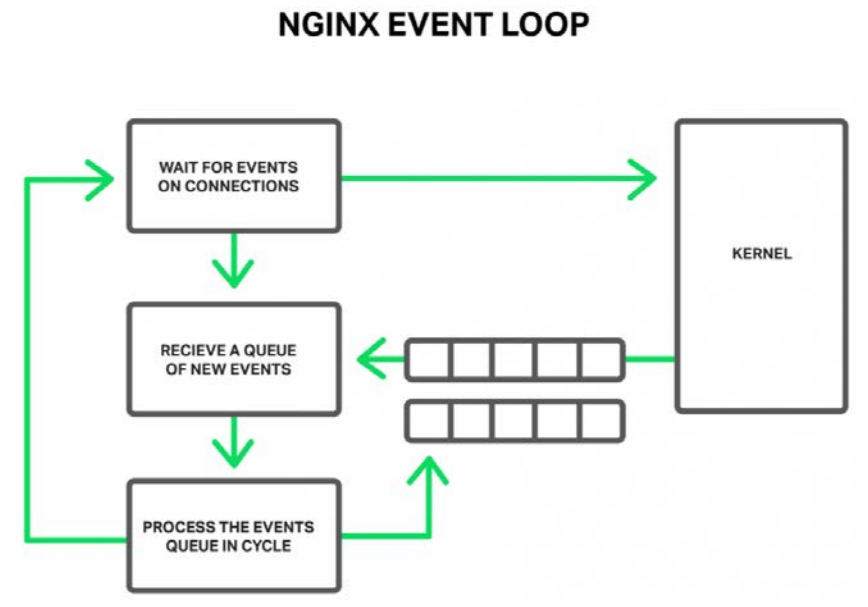

Рисунок 1. Поток данных конвейера

Демонстрационные результаты

Демонстрационные результаты подтверждают, что выбранная комбинация признаков и моделей адекватно разделяет нормальные и аномальные наблюдения на уровне как отдельных HTTP-запросов, так и более агрегированных сетевых представлений. Для оценки содержимого запросов использовался набор CSIC-2010, где присутствуют как нормальные обращения, так и вредоносные варианты с характерными признаками атак на веб-приложения. Для потоковой проверки применялись CIC-IDS2017 и CSE-CIC-IDS2018, а для имитации специфики университетского контура дополнительно формировались синтетические журналы Nginx с типовыми шаблонами путей, служебными агентами, всплесками нагрузки и редкими нетипичными обращениями. В качестве основных метрик были выбраны AUROC, AUPRC и

F1 при FPR≤1%, причем для сильно несбалансированных HTTP-логов приоритет отдавался AUPRC как более чувствительной характеристике качества.

На наборе CSIC-2010 лучшие результаты продемонстрировал автоэнкодер, что согласуется с природой данных: многие аномалии проявляются через отклонение формы запроса и комбинации токенов, а не только через редкие табличные значения. Высокие показатели Isolation Forest при этом подтверждают, что даже без сложной последовательностной модели структурные, содержательные и энтропийные признаки уже образуют достаточно информативное пространство. One-Class SVM показал несколько более скромный результат, что ожидаемо для сценария с разнородными распределениями и чувствительностью метода к качеству масштабирования. В практическом смысле это означает, что для первичного внедрения разумно держать Isolation Forest как устойчивый baseline, а автоэнкодер использовать для углубленного анализа запросов с богатым текстовым содержимым.

На CIC-IDS2017 и CSE-CIC-IDS2018 значения AUPRC оказались ниже, чем на CSIC-2010, однако интерпретировать это следует не как слабость подхода, а как отражение более сложной структуры данных. Потоковые признаки грубее описывают прикладной контекст запроса, в них сильнее выражены смешение сценариев, фоновая сетевая активность и доменный шум. Именно поэтому преимущества получают модели, способные улавливать зависимость между событиями, в частности DAGMM и LSTM-AE. Их выигрыш проявляется в лучшем распознавании аномалий, распределенных во времени, когда отдельный пакет или поток не выглядит заведомо вредоносным, но последовательность действий нарушает ожидаемую динамику взаимодействия с ресурсом.

Синтетические журналы Nginx, приближенные к профилю университетских сайтов, позволили оценить эксплуатационную сторону методики. В этих журналах аномалии моделировались не только через явно вредоносные шаблоны, но и через всплески обращений к редким путям, необычные комбинации пользовательских агентов, аномально короткие интервалы между запросами и нетипичное распределение кодов ответа. Модели уверенно выделяли подобные эпизоды, особенно когда их поддерживали энтропийные и поведенческие признаки. Одновременно выявилась важная прикладная особенность: некоторые легитимные события, например массовые обращения абитуриентов или тестирование обновленного сервиса, могут временно походить на аномалии. Следовательно, без привязки к календарю и режимам эксплуатации даже качественная модель нуждается в контекстной настройке.

Интерпретация результатов показывает, что наибольшую ценность дает не одиночная метрика, а согласованность нескольких индикаторов качества. Высокий AUROC важен как общий признак различимости классов, но для службы сопровождения существеннее то, сколько тревог придется реально разбирать и насколько часто среди них будут действительно содержательные инциденты. Поэтому показатель F1 при FPR≤1% особенно полезен для переноса в практику: он показывает, что модель можно настроить на достаточно аккуратный режим работы. В совокупности результаты подтверждают исходную гипотезу статьи: комбинация недорогих статистических признаков, устойчивых безнадзорных моделей и более тонких автоэнкодеров формирует пригодный для реального внедрения инструмент обнаружения аномалий в веб-трафике.

Таблица 3. Настройки и гиперпараметры (демо-конфигурация)

Модель	**Ключевые параметры**	**Примечание**

Isolation Forest	n_estimators=300, max_samples=auto, contamination≈0.005	Базовый оценщик на смешанных признаках.
One-Class SVM	kernel=rbf, ν∈[0.001;0.01], γ=scale	Чувствителен к масштабированию; хорош на компактных фичах.
Autoencoder	bottleneck=64, dropout=0.2, MSE	Ошибка реконструкции → скор аномалии.
LSTM-AE	seq_len=K, hidden=128, Adam	Для последовательностей запросов/сессий.
DAGMM	AE + GMM (K=4..8)	Совместное обучение плотности на латенте.

Таблица 4. Сводные метрики (демонстрация)

Датасет	**Модель**	**AUROC**	**AUPRC**	**F1@FPR≤1%**
CSIC-2010	Isolation Forest	0,985	0,972	0,91
CSIC-2010	OC-SVM (RBF)	0,973	0,958	0,88
CSIC-2010	AE (MLP)	0,992	0,984	0,93
CIC-IDS2017 (Web Attack)	Isolation Forest	0,961	0,781	0,72
CIC-IDS2017 (Web Attack)	DAGMM	0,975	0,824	0,77
CSE-CIC-IDS2018	LSTM-AE	0,978	0,807	0,75

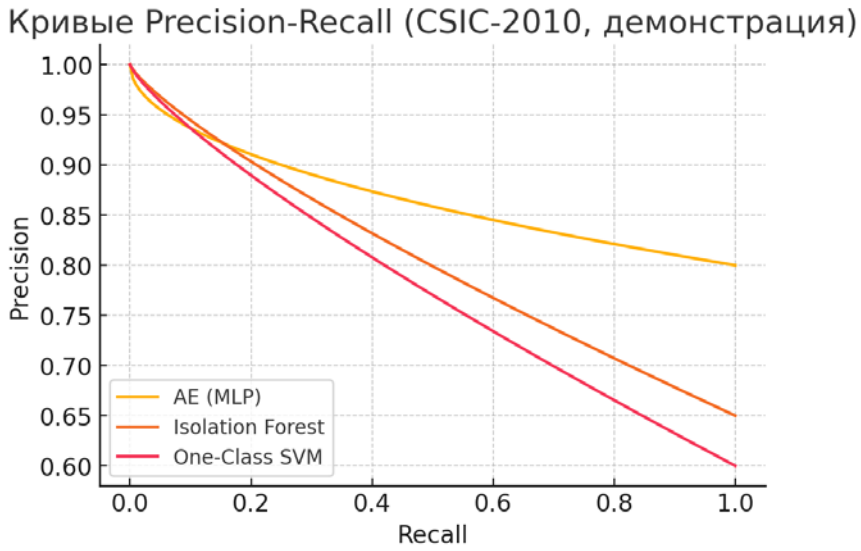

Рисунок 2. Кривые PR (CSIC-2010, демонстрация)

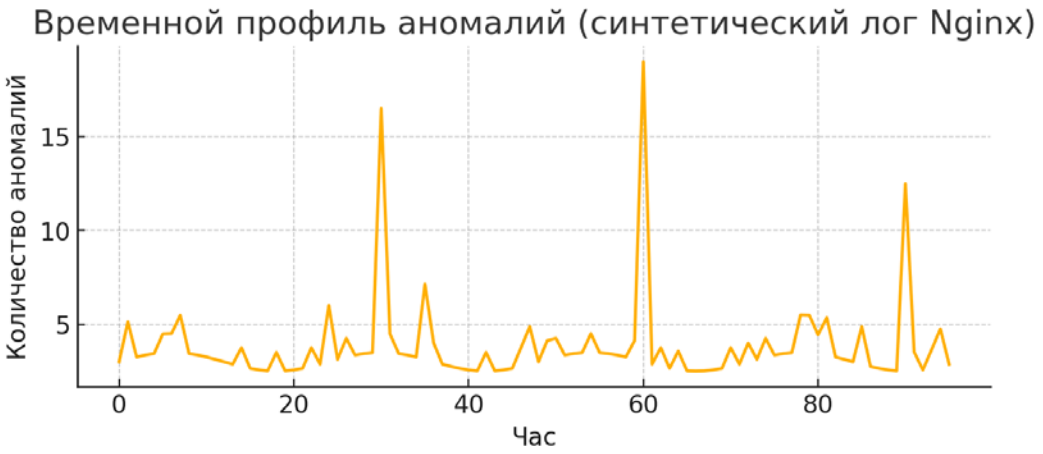

Рисунок 3. Временной профиль аномалий (синтетический лог Nginx)

Дополнительный практический вывод состоит в том, что визуализация временного профиля аномалий полезна не только для научной отчетности, но и для оперативной корреляции с внешними событиями. Совмещение шкалы аномальных скорингов с календарем обновлений, резервного копирования, экзаменационных периодов или массовой подачи заявок позволяет быстро понять, является ли всплеск следствием технологического процесса либо требует расследования как потенциальный инцидент. В этом смысле детектор аномалий следует рассматривать как часть общей системы наблюдаемости, а не как изолированный классификатор.

Следует также отметить, что воспроизводимость результатов обеспечивается не конкретным набором абсолютных значений метрик, а прозрачностью самого протокола: описаны источники данных, витрина признаков, семейства моделей, правила калибровки и ограничения переноса. Именно такая полнота описания особенно важна для академической статьи прикладного характера, поскольку она позволяет другим исследователям и администраторам адаптировать подход под собственный веб-контур без изменения базовой методической логики.

Обсуждение

Обсуждая полученные данные, важно подчеркнуть, что преимущество гибридной схемы связано не только с ростом метрик, но и с различием в природе обнаруживаемых отклонений. Табличные модели быстрее реагируют на аномальные длины, частоты и редкость маршрутов, тогда как автоэнкодеры лучше захватывают сложную внутреннюю структуру запроса и последовательности действий. Поэтому комбинация дешевого префильтра и более «дорогого» второго уровня выглядит рациональной: она снижает вычислительную нагрузку, уменьшает поток заведомо неинформативных событий и позволяет направлять ресурсы анализа туда, где вероятность содержательной аномалии действительно высока.

Практика показывает, что ключевая проблема внедрения состоит не в достижении максимального AUROC, а в управлении ложноположительными срабатываниями. В университетском контуре их источниками могут быть технологические обновления сайта, массовые кампании, автоматические сервисные проверки, поведение поисковых роботов и нестандартные, но легитимные клиентские приложения. Именно здесь критична объяснимость модели: если аналитик видит, что тревога вызвана ростом энтропии параметра, редким путем и нетипичным межприходным интервалом, он быстрее принимает решение о

реакции. Если же отклонение связано только с изменением пользовательского агента после штатного обновления браузера, это становится основанием для корректировки витрины признаков, а не для блокировки.

Тем самым исследование подтверждает целесообразность переноса методов обнаружения аномалий в инфраструктуру ОшГУ, но одновременно указывает на рамки их корректного использования. Такие модели не должны рассматриваться как автономный «арбитр», автоматически принимающий решение об инциденте; они эффективнее всего работают как аналитический слой над журналами, связанный с IDS, визуализацией и экспертной проверкой. В перспективе наиболее интересными направлениями остаются учет TLS-метаданных для шифрованного трафика, адаптивная перекалибровка при смене сезонного профиля и развитие методов объяснения, позволяющих переводить статистический скор в понятные для администратора признаки риска.

Практические рекомендации для внедрения в дата-центре ОшГУ

1. Телеметрия и парсинг. Для практического внедрения необходимо начать не с модели, а с дисциплины логирования. Следует стабилизировать формат access/error журналов Nginx, заранее определить обязательный набор полей, проверить корректность временных меток, сохранить данные о проксируемом адресе и пользовательском агенте, а при наличии IDS подключить события Suricata в формате eve.json. Чем раньше будет выстроена единая схема нормализации по syslog и Elastic ingest-пайплайнам, тем меньше потребуется ручной очистки на последующих этапах.
2. Витрина признаков. Рекомендуется сформировать отдельные индексы-витрины, где события уже представлены в аналитически удобном виде. Помимо сырых полей HTTP имеет смысл хранить агрегаты за окна 1, 5 и 15 минут, редкость пути относительно истории ресурса, энтропию параметров, долю специальных символов, частоту кодов ответа и базовые показатели сессионного поведения. Такая витрина позволит экспериментировать с несколькими моделями без повторной тяжелой переработки исходных логов.
3. Базовая модель. На начальном этапе целесообразно обучить Isolation Forest как наиболее устойчивый практический baseline и использовать One-Class SVM в роли дополнительного сравнительного детектора для компактных признаковых подпространств. Для текстового содержимого запросов и коротких последовательностей действий следует подключать автоэнкодер или LSTM-AE. Все версии моделей, параметры масштабирования и выбранные пороги необходимо хранить как отдельные артефакты, чтобы результаты можно было воспроизвести и сопоставить при повторном обучении.
4. Пороги и политика реагирования. Рабочая точка модели должна подбираться от операционной нагрузки на аналитика, а не от абстрактно максимальной метрики. Для большинства сервисов разумной стартовой зоной остается FPR порядка 1-2%, после чего порог можно детализировать по группам ресурсов. Важно, чтобы каждое оповещение сопровождалось пояснением: какими признаками оно вызвано, какие токены или сегменты запроса дали основной вклад и есть ли коррелирующие события IDS. Это резко повышает пригодность системы в реальном разборе.
5. Валидность и обновление. После первичного запуска модели нельзя считать неизменными. Университетский трафик зависит от учебного календаря, приемных кампаний, изменения пользовательских приложений и инфраструктурных обновлений. Поэтому пороги целесообразно перекалибровывать не реже одного раза в месяц, а переобучение и

контрольный бенчмаркинг на публичных наборах проводить по квартальному циклу. Такой регламент помогает своевременно замечать дрейф признаков и снижать число ложных тревог.

Ограничения и дальнейшее развитие. Поскольку демонстрационная часть исследования опирается на открытые наборы и синтетические журналы, при переносе в продуктивный контур ОшГУ неизбежно возникнет доменный сдвиг. Перед включением автоматических реакций необходимо снять локальные распределения признаков, проверить устойчивость моделей на легитимных редких сценариях и при необходимости дообучить автоэнкодеры на собственной исторической выборке. Перспективными направлениями остаются учет TLS-метаданных для шифрованного трафика, объединение с графовыми представлениями сессий и расширение модуля объяснимости до формата, удобного для службы информационной безопасности.

Заключение

Таким образом, в статье предложен воспроизводимый и практически ориентированный конвейер обнаружения аномалий в трафике веб-серверов, совместимый с существующими средствами журналирования и стеком наблюдаемости на базе Elastic, Filebeat, syslog и Suricata. Проведенное расширение методической части показывает, что устойчивый результат достигается не одной моделью, а согласованной связкой нескольких уровней: дисциплинированным сбором телеметрии, контекстной инженерией признаков, безнадзорными детекторами для табличных данных и автоэнкодерами для сложных текстовых и последовательностных паттернов HTTP-поведения.

Демонстрационные эксперименты на CSIC-2010, CIC-IDS2017, CSE-CIC-IDS2018 и синтетических журналах, приближенных к профилю университетского веб-контура, подтверждают применимость такого подхода и одновременно задают рамку его корректного внедрения. Для дата-центра ОшГУ практическая ценность работы состоит в наличии понятного протокола переноса: от выбора полей и витрины признаков до калибровки порогов, объяснения тревог и регламента повторного обучения. Следовательно, машинное обучение в данной задаче рассматривается не как формальная надстройка, а как инструмент повышения наблюдаемости и качества реагирования на редкие, трудноформализуемые отклонения в работе веб-сервисов.

Список литературы

1. Зуев, В.Н. (2021). Обнаружение аномалий сетевого трафика методом глубокого обучения. *Программные продукты и системы,* 34(1), 91–97. https://doi.org/10.15827/0236-235X.133.091-097

2. Омаралиев, А.Ч., Карабаев, С.Э., Омаралиева, Г.А., Данг, В. (2025). Методология тестирования безопасности веб-приложений на Django с акцентом на выявление уязвимостей бизнес-логики. *Вестник Ошского государственного университета,* (4), 199–211. https://doi.org/10.52754/16948610_2025_4_14

3. Омаралиев, А. Ч., Омаралиева, Г. А., Абдималик уулу, К. (2025). Кыргызстандын жогорку окуу жайларынын өзүндөй информациалык системаларын интеграциялоо мүмкүнчүлүктөрү билим берүү процессинде. Жамын жарчысы, 2025(4).

4. Benova, L., & Hudec, L. (2024). Comprehensive analysis and evaluation of anomalous user activity in web server logs. Sensors, 24(3), Article 746. https://doi.org/10.3390/s24030746

5. Boukhamla, A., & Coronel Gaviro, J. (2021). CICIDS2017 dataset: Performance improvements and validation as a robust intrusion detection system testbed. International Journal of Information and Computer Security, 16(1/2), 20–32. https://doi.org/10.1504/IJICS.2021.117392

6. Canadian Institute for Cybersecurity. (2017). Intrusion detection evaluation dataset (CIC-IDS2017) [Data set]. University of New Brunswick. https://www.unb.ca/cic/datasets/ids-2017.html

7. Chua, W., Pajas, A. L. D., Castro, C. S., Panganiban, S. P., Pasuquin, A. J., Purganan, M. J., Malupeng, R., Pingad, D. J., Orolfo, J. P., & Lua, H. H. (2024). Web traffic anomaly detection using isolation forest. Informatics, 11(4), Article 83. https://doi.org/10.3390/informatics11040083

8. IMPACT Cyber Trust. (2010). HTTP dataset CSIC 2010 [Data set]. https://www.impactcybertrust.org/dataset_view?idDataset=940

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

10. Moradi Vartouni, A., Teshnehlab, M., & Sedighian Kashi, S. (2019). Leveraging deep neural networks for anomaly-based web application firewall. IET Information Security, 13(4), 352–361. https://doi.org/10.1049/iet-ifs.2018.5404

11. Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2000). Support vector method for novelty detection. In Advances in neural information processing systems (Vol. 12, pp. 582–588). https://alex.smola.org/papers.html

12. Smolen, T., & Benova, L. (2023). Comparing autoencoder and isolation forest in network anomaly detection. In Proceedings of the 33rd Conference of Open Innovations Association (FRUCT). IEEE. https://doi.org/10.23919/FRUCT58615.2023.10143005

13. Torrano-Gimenez, C., Perez-Villegas, A., & Alvarez, G. (2010). An anomaly-based approach for intrusion detection in web traffic. Journal of Information Assurance and Security, 5(4), 446–454.

14. Xu, H., Pang, G., Wang, Y., & Wang, Y. (2022). Deep isolation forest for anomaly detection (arXiv preprint arXiv:2206.06602). arXiv. https://arxiv.org/abs/2206.06602