ТЕХНИКАЛЫҚ ҒЫЛЫМДАР
ТЕХНИЧЕСКИЕ НАУКИ
TECHNICAL SCIENCE

CENTRAL ASIAN SCIENTIFIC JOURNAL
ISSN 3135-3061
2026. №4 (32), Том 2. С. 291–298
DOI: https://doi.org/10.5281/zenodo.24197903
Сайт журнала: https://cajournal.kz

УДК 004.056.5:004.8

ГИБРИДНАЯ АРХИТЕКТУРА ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ОБНАРУЖЕНИЯ И ПРЕДОТВРАЩЕНИЯ DDOS-АТАК В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ

А.Ю. Авдеева, А.М. Джаканова, Е.А. Мясникевич
Алматы, Казахстан

Авдеева Алёна Юрьевна — Студент 4 курса, Кафедра Кибербезопасность, Алматы, Казахстан.

Джаканова Аружан Мураткызы — Студент 4 курса, Кафедра Кибербезопасность, Алматы, Казахстан.

Мясникевич Ева Андреевна — Студент 4 курса, Кафедра Кибербезопасность, Алматы, Казахстан.

© А.Ю. Авдеева, А.М. Джаканова, Е.А. Мясникевич, 2026

Аннотация: В работе рассматривается проектирование гибридной системы обнаружения и предотвращения распределённых атак типа «отказ в обслуживании» на основе поведенческого анализа сетевого трафика и методов машинного обучения. Цель исследования — обосновать архитектуру, объединяющую программные датчики, программно-аппаратные механизмы обработки пакетов и облачные средства фильтрации с учётом требований к скорости принятия решения. На основе научных публикаций предложены состав сетевых признаков, совместное использование классификации, кластеризации и эвристической оценки риска, порядок адаптивного обновления моделей и многоуровневая политика реагирования. Сформулированы критерии проверки на размеченных сетевых потоках и на временных окнах, в том числе полнота обнаружения, частота ложных блокировок, пропускная способность и сквозная задержка от появления признаков атаки до защитного действия. Целевое время обнаружения рассматривается как проектное требование — от долей секунды до нескольких секунд. Результатом является концептуальная спецификация и протокол последующей экспериментальной валидации. Исследование имеет проектно-аналитический характер: промышленная реализация и измерение сквозной задержки не проводились.

Ключевые слова: DDoS-атаки, машинное обучение, поведенческий анализ, обнаружение аномалий, кластеризация, облачная защита, программируемые сети, адаптивная фильтрация, задержка обнаружения.

Введение

Обеспечение доступности веб-приложений, облачных платформ и корпоративных сетевых сервисов становится самостоятельной задачей информационной безопасности. Распределённая атака типа «отказ в обслуживании» (Distributed Denial of Service, DDoS) нарушает работу системы за счёт согласованного формирования трафика из множества источников. В зависимости от механизма воздействие может быть направлено на пропускную способность канала, таблицы состояний сетевых устройств, серверные соединения либо обработку запросов прикладного уровня. Классификация подобных воздействий и способов защиты представлена в работе J. Mirkovic и P. Reiher [1].

Традиционные средства противодействия опираются на ограничение скорости, фильтрацию протокольных аномалий, репутационные списки и известные сигнатуры. Такие

меры позволяют быстро отсекать очевидные нарушения, но не всегда различают злонамеренную нагрузку и массовый приток легитимных пользователей. В обзоре S. T. Zargar и соавторов [2] показано, что защита от DDoS требует согласования методов обнаружения с механизмами ограничения и перенаправления трафика. Для высокоскоростных сетей дополнительную сложность составляет необходимость анализировать большие объёмы событий без чрезмерного роста задержки и потерь пакетов [3].

Предметную область настоящего исследования образуют программные, программно-аппаратные и облачные средства обнаружения и предотвращения DDoS-атак. В отличие от системы, использующей только один классификатор, предлагается согласовать несколько типов аналитики: быстрые эвристики, обучение на размеченных потоках, выявление отклонений от локального профиля поведения и кластеризацию подозрительных событий. Результат аналитики должен поступать в контур реагирования, способный выбирать действие в соответствии с уверенностью модели и потенциальным ущербом от ошибочной блокировки.

Цель работы — разработать научно обоснованную функциональную архитектуру интеллектуальной системы с целевым временем обнаружения от 0,1 до 5 с, определить набор входных признаков и описать методику независимой экспериментальной проверки. Указанный диапазон задаёт ориентир проектирования, а не измеренный показатель создаваемого решения. Научно-методическая новизна предлагаемых принципов состоит в едином контуре оценки риска, объединяющем разнотипные детекторы с проверяемой политикой реагирования, измерением сквозной задержки и контролируемым обновлением модели при изменении сетевого поведения.

Сравнительный анализ методов обнаружения DDoS-атак

Методы обнаружения DDoS-атак различаются по природе входных данных, требованиям к разметке и вычислительной сложности. В первой группе находятся фиксированные пороговые и эвристические правила. Они могут отслеживать частоту SYN-пакетов, несоответствие числа установленных соединений и попыток подключения, всплески UDP-трафика, а также долю ответов сервера с ошибками. Их преимущество — прозрачная логика и возможность выполнения в плоскости обработки пакетов. Главный недостаток заключается в зависимости от заранее заданных границ, которые недостаточно учитывают суточную сезонность и событийные пики.

Вторую группу составляют алгоритмы классификации с учителем: логистическая регрессия, деревья решений, случайный лес, градиентный бустинг и нейросетевые модели. Классификаторы устанавливают статистические зависимости между признаками и метками «нормальный трафик» / «атака». В исследовании LUCID продемонстрирован подход к эффективной классификации сетевых потоков с использованием компактной свёрточной сети [4]. Вместе с тем для применения подобных решений в новой инфраструктуре требуется учитывать расхождение между обучающими данными и текущим распределением сетевого трафика.

Программно-аппаратные решения могут выполнять агрегацию пакетов непосредственно на коммутаторах и сетевых адаптерах, уменьшая объём телеметрии, передаваемой в центральный анализатор. В работе F. Musumeci и соавторов [5] исследовано объединение алгоритмов машинного обучения с программируемой плоскостью данных P4 для обнаружения SYN-flood. Этот подход подтверждает перспективность распределения функций между быстрым сетевым трактом и более сложной моделью, но его реализация зависит от поддерживаемых инструкций и памяти оборудования.

Третья группа объединяет обнаружение аномалий без полного набора размеченных атак и кластеризацию. Например, автоэнкодер может быть обучен восстанавливать типичное поведение сети, а отклонение между исходным и восстановленным вектором рассматривается как сигнал необычной активности. Подобная идея реализована в системе Kitsune [6]. Кластеризация, в свою очередь, выделяет группы окон или потоков, сходных по

интенсивности и структуре, а редкие либо удалённые группы передаются на дополнительную проверку. При этом аномалия сама по себе ещё не означает DDoS: обзор V. Chandola и соавторов [7] подчёркивает зависимость результата от определения «нормы» и природы наблюдаемых данных.

Таблица 1 — Сопоставление методов интеллектуального обнаружения DDoS-атак

Подход	Преимущества	Основные ограничения
Пороговые правила и эвристики	Быстрое выполнение, объяснимость, простая реализация	Ошибки при легитимных всплесках и изменении профиля нагрузки
Классификация с учителем	Хорошее различение изученных классов, оценка уверенности	Необходимы размеченные данные; риск переобучения и утечек
Кластеризация и поиск аномалий	Поиск нетипичных групп без исчерпывающей разметки	Аномалия не равна атаке; сложно выбрать чувствительность
Автоэнкодеры и глубокие модели	Учет нелинейных связей между признаками	Вычислительные затраты, интерпретируемость и переносимость
Гибридный анализ	Сочетание быстрых сигналов, ML и контекста	Сложность калибровки и согласования действий модулей

Таким образом, наиболее перспективным представляется не универсальный алгоритм, а архитектура, допускающая независимое получение сигналов и объединение их в контролируемое решение. В малонагруженной корпоративной сети достаточным может оказаться программный анализатор; на высокоскоростном магистральном сегменте часть сбора статистики целесообразно перенести на сетевой адаптер или коммутатор, а для публичного сервиса предусмотреть облачную очистку трафика.

Архитектура предлагаемой интеллектуальной системы

Входом предлагаемой системы служат сетевые пакеты, сведения о соединениях и прикладная телеметрия: журналы HTTP-запросов, коды ответов, данные о времени обслуживания и нагрузке процессора. Сбор выполняется несколькими датчиками, размещёнными на границе сети, у серверов и, при необходимости, в облачном сегменте. Сведения об IP-адресах и идентификаторах соединений используются лишь в пределах задач корреляции и реагирования; для хранения аналитических выборок предусматриваются минимизация и псевдонимизация персонально значимых полей.

Первый технологический уровень — быстрый сбор и агрегация. В программной реализации он может использовать стандартные механизмы мониторинга ОС и потоковые счётчики, в программно-аппаратной — eBPF/XDP, SmartNIC либо коммутаторы с P4. Исходные события агрегируются в последовательные или перекрывающиеся окна. Для эксперимента предлагается сопоставить длительности 0,25; 1 и 5 с. Короткие окна позволяют быстрее сформировать сигнал, но обладают более высокой дисперсией статистических признаков; длинные дают более устойчивую оценку, увеличивая время накопления наблюдений.

Базовые признаки включают скорость пакетов и байтов, количество TCP SYN, ACK и RST, соотношение SYN/ACK, долю UDP, число уникальных источников и назначений, энтропию распределения адресов, частоту новых соединений и долю ошибок прикладного сервиса. Формирование признаков необходимо организовать на уровне окна и с документированным порядком обработки отсутствующих значений. Утечка прямой метки сценария, времени выполнения атаки и неизменных лабораторных адресов в набор предикторов должна быть исключена, иначе высокая точность классификатора может отражать особенности стенда, а не выявление вредоносного поведения.

Аналитический контур содержит три логически независимых модуля. Модуль поведенческого анализа сравнивает текущие характеристики с динамическим профилем нормальной активности, учитывая время суток и сервисную роль узла. Сигнал рассчитывается по отклонениям интенсивности и структуры трафика, например с помощью скользящего среднего и устойчивых оценок разброса. Модуль классификации получает

признаки и выдаёт оценку принадлежности окна к известным классам атак. Модуль аномалий группирует нетипичные события или оценивает степень их удалённости от накопленных нормальных кластеров.

Раздельность модулей позволяет не считать единственный экстремальный показатель достаточным основанием для блокировки. Например, рост числа SYN-пакетов при одновременном пропорциональном росте ACK может отражать законный приток клиентов. Напротив, увеличение числа источников, высокое отношение SYN/ACK и деградация прикладного сервиса вместе образуют более сильный комплексный сигнал. Итоговая оценка риска создаётся на основании согласованных и нормированных выходов модулей; её пороги выбираются только после калибровки на отдельной выборке.

После объединения сигналов политика реагирования выбирает один из нескольких уровней воздействия. При низком риске события только журналируются; при среднем выполняется усиленный мониторинг, проверка дополнительных показателей или мягкое ограничение скорости; при высоком, подтверждённом несколькими независимыми признаками, допускается временная блокировка, фильтрация по классу трафика либо перенаправление на облачную площадку очистки. Система должна предусматривать исключения для критически важных адресов, ограниченное время действия правил и возможность немедленного отката ошибочного решения.

Архитектура должна разделять собственно обнаружение и предотвращение. Программный анализатор может выдать предупреждение, не изменяя маршруты; программируемый коммутатор — установить краткосрочное правило фильтрации; облачный сервис — распределить либо очистить массовый входящий поток. Оператору следует предоставлять номер версии модели, объяснение основных факторов риска и сведения о применённой мере. Для интерпретации сложных классификаторов могут использоваться методы оценки вклада признаков, однако вывод подобного метода требует отдельной проверки устойчивости [8].

Таблица 2 — Распределение функций между уровнями защиты

Уровень	Основные функции	Контрольные требования
Программный	Агрегация потоков, расчёт признаков, классификация, журналирование	CPU/RAM, очередь, пропуски событий, обновление модели
Программно-аппаратный	Счётчики на коммутаторе/SmartNIC, предварительная фильтрация	Ограничения памяти, стабильность правил, время реакции
Облачный	Корреляция между площадками, очистка и перераспределение трафика	Сетевые задержки, доступность канала, конфиденциальность
Контур управления	Калибровка оценок, управление версиями и откат	Аудит действий, проверка ложных блокировок

Требование работы в режиме реального времени необходимо описывать через сквозную задержку, а не только через скорость вызова ML-модели. Время до решения определяется накоплением наблюдений в окне, пересылкой телеметрии, построением признаков, вычислением оценки, обработкой очереди и выполнением защитного действия. Для выбранной конфигурации предлагается измерять задержку от момента появления первых диагностически значимых признаков до предупреждения, а отдельно — до применения фильтрующего правила. Уменьшение времени предсказания с миллисекунд до микросекунд не обеспечивает субсекундную защиту, если само окно накопления длится несколько секунд.

Для обновления моделей предусматривается выделенный контур обучения, не выполняющий изменение рабочего классификатора автоматически. Новая версия сначала проверяется на независимом временном участке и в теневом режиме, когда решения вычисляются, но не влияют на трафик. После оценки ложных срабатываний и устойчивости к изменению классов версия может быть развёрнута с возможностью отката. Подобное разделение соответствует подходу к концептуальному дрейфу, при котором характеристики потока изменяются во времени и требуют контролируемой адаптации [9].

План экспериментальной проверки

Представленная архитектура является проектной спецификацией. Ниже приведён воспроизводимый план проверки её компонентов; приведённые требования не следует трактовать как уже достигнутые показатели. Для первого этапа подходит общедоступный набор CICDDoS2019, описанный I. Sharafaldin и соавторами [10]. Датасет включает PCAP-захваты и размеченные потоковые признаки, полученные в контролируемой инфраструктуре, и может использоваться для оценки классификации известных DDoS-сценариев. Однако потоковые CSV не тождественны последовательным временным окнам: для испытаний точной задержки и поведения в потоке необходимо отдельно обработать исходные PCAP с сохранением времени поступления пакетов.

В исследовательском протоколе предлагается использовать базовые пороговые правила, логистическую регрессию, случайный лес, градиентный бустинг, изолирующий поиск аномалий и гибридную схему, объединяющую их оценки. При этом конфигурации сравниваются при одинаковых признаках и одной процедуре разбиения данных. На первом этапе измеряется способность отделять известные атаки от нормального трафика; затем создаются сценарии с изменённой долей легитимных всплесков, сдвигом распределения интенсивности и ранее не встречавшимися атаками. Работа R. Sommer и V. Paxson [11] обосновывает необходимость проверки вне ограниченного «замкнутого мира» тренировочных меток.

Для предотвращения утечки информации следует отделять обучающие, валидационные и тестовые данные по временным интервалам и, где это возможно, по независимым сеансам, источникам и сценариям. Повторяющиеся потоки, производные одного захвата и почти одинаковые признаки должны выявляться до разбиения. Стандартизация, отбор характеристик и подбор порогов выполняются только на обучающей и валидационной частях; тестовая выборка должна оставаться закрытой до итоговой оценки. Если используются данные из нескольких источников, необходимо дополнительно фиксировать происхождение, формат, качество меток и долю пропусков.

Основными метриками предлагаются Precision, Recall, F1, PR-AUC, частота ложноположительных блокировок и доля пропущенных атак. В условиях низкой распространённости опасного трафика Accuracy может существенно завышать представление о реальном качестве, поэтому для несбалансированных выборок важен анализ Precision–Recall-кривой [12]. Классификационная производительность измеряется отдельно от задержки накопления окна, сетевого ввода, расчёта признаков, очередей и действия фильтра. Дополнительно следует оценивать 50-й, 95-й и 99-й процентили сквозной задержки, пропускную способность в пакетах в секунду, память и долю потерянных событий.

Таблица 3 — Проверяемые гипотезы и критерии оценки предлагаемой системы

Гипотеза	Схема сравнения	Критерий оценки
H1. Объединение моделей снижает число ошибок	Отдельный классификатор / гибридная схема	F1, PR-AUC, FP и FN
H2. Поведенческий контекст уменьшает ложные блокировки	Без профиля нагрузки / с профилем	FP при легитимных всплесках
H3. Анализ в сетевом тракте сокращает задержку	Только программный датчик / P4 или SmartNIC	Сквозная задержка p50/p95/p99
H4. Облачный контур повышает устойчивость к перегрузке	Без облачной очистки / с очисткой	Доступность сервиса, потери пакетов
H5. Контролируемая адаптация сохраняет качество	Фиксированная модель / обновлённая модель	F1 при дрейфе, FP, стоимость обновления

Целевой диапазон задержки 0,1–5 с предлагается проверять на нескольких длинах окон, типах сетевого тракта и сценариях нагрузки. В протоколе необходимо заранее зафиксировать момент начала отсчёта: первый аномальный пакет, начало устойчивого превышения профиля либо момент появления достаточного числа признаков для обоснованного решения. Разные определения дают несопоставимые значения задержки. Для

проверки реагирования следует измерить не только время формирования предупреждения, но и фактическое время установки и отмены сетевого правила.

Важной частью испытаний является калибровка риска: значение 0,9 на выходе классификатора не всегда означает подтверждённую вероятность атаки 90 %. Методы температурного масштабирования и другие подходы к согласованию уверенности с наблюдаемой частотой ошибок обсуждаются C. Guo и соавторами [13]. В условиях эксплуатации пороги необходимо выбирать исходя из потерь от пропуска атаки и ущерба от ошибочной блокировки легитимного клиента, а не исключительно по максимальному F1.

Практическое применение и ограничения

Предложенная система потенциально применима в университетских сетях, дата-центрах, провайдерских сегментах и облачных информационных системах, для которых нарушение доступности критически влияет на работу пользователей. В относительно малом сегменте допустим программный датчик с простыми правилами и периодическим дообучением. При значительном объёме трафика требуется разграничение функций: быстрое обнаружение высокоинтенсивных аномалий в сетевом тракте и сложный анализ агрегированных данных на отдельном узле. В SDN-средах подобное разделение согласуется с исследованиями интеллектуального обнаружения атак на программируемой инфраструктуре [14].

При принятии решения нельзя игнорировать экономическую и функциональную цену ложных срабатываний. Массовая регистрация на сервисе, трансляция или обновление программного обеспечения могут создавать профиль нагрузки, сходный с отдельными признаками DDoS. Поэтому высокие оценки риска целесообразно подтверждать комплексом независимых показателей, а грубую блокировку заменять по возможности краткосрочным ограничением скорости либо проверкой на стороне приложения. В журналах следует хранить объяснение решения, исходную оценку, применённую политику и факт её последующей отмены.

Стабильность работы зависит от безопасности самого аналитического контура. Злоумышленник способен формировать низкоинтенсивные и адаптивные атаки, искажать телеметрию, добиваться накопления ложных меток и провоцировать ошибочное дообучение. Модель следует загружать из доверенного хранилища, данные разметки — проверять, а рабочие версии — подписывать и сохранять вместе с параметрами. Изменение политики без журнала аудита создаёт дополнительный риск, поскольку некорректная фильтрация может вызвать отказ в обслуживании даже при отсутствии внешней атаки.

Необходимо учитывать и ограниченность доступных наборов данных. CICDDoS2019 отражает заданную исследовательскую инфраструктуру и конкретные сценарии; высокая точность на её потоковых признаках не гарантирует сопоставимое качество при иных ботнетах, новом сетевом оборудовании и атаках прикладного уровня. Обобщение возможно только после внешней проверки на дополнительных наборах и в условиях, где поведение легитимных пользователей отличается от обучающей среды. Для алгоритмов деревьев, включая случайный лес, важны интерпретируемость и устойчивость к нелинейным признакам, но ограничения переноса между сетями сохраняются независимо от семейства модели [15].

Ограничения настоящей работы принципиальны: представленная архитектура не развёрнута в продуктивной сети; общий контур реагирования не испытывался на маршрутизаторах, программируемых коммутаторах либо облачных площадках; целевая задержка от 0,1 до 5 с не подтверждена сквозными измерениями. Следовательно, предложенные решения следует рассматривать как обоснованную спецификацию для дипломного проекта и дальнейшей реализации, а не как доказательство достигнутой промышленной эффективности.

Заключение

В работе предложена гибридная архитектура интеллектуальной системы обнаружения и предотвращения DDoS-атак, объединяющая поведенческий анализ трафика,

классификацию с учителем, кластеризацию аномалий и эвристические правила. Рассмотрено распределение функций между программными датчиками, программно-аппаратной плоскостью обработки пакетов, облачным контуром защиты и механизмами централизованного управления. Основным принципом является отделение формирования риска от применения защитного действия с учётом уверенности модели, цены ложных блокировок и возможности отката.

Сформулирован протокол последующей экспериментальной проверки, предусматривающий независимое разделение данных, сравнение с базовыми методами, оценку устойчивости к легитимным всплескам и сдвигу распределения, а также измерение сквозной задержки обнаружения и применения фильтра. Достоверность заявленных преимуществ и возможность выполнения требований реального времени должны быть подтверждены реализацией прототипа и испытаниями на потоковом сетевом стенде. Полученная спецификация определяет методическую основу дальнейшей дипломной разработки.

Список литературы:

1. Mirkovic J., Reiher P. A Taxonomy of DDoS Attack and DDoS Defense Mechanisms // ACM SIGCOMM Computer Communication Review. — 2004. — Vol. 34, № 2. — P. 39–53. — DOI: 10.1145/997150.997156.

2. Zargar S. T., Joshi J., Tipper D. A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks // IEEE Communications Surveys & Tutorials. — 2013. — Vol. 15, № 4. — P. 2046–2069. — DOI: 10.1109/SURV.2013.031413.00127.

3. Haseeb-ur-rehman R. M. A. et al. High-Speed Network DDoS Attack Detection: A Survey // Sensors. — 2023. — Vol. 23, № 15. — Art. 6850. — DOI: 10.3390/s23156850.

4. Doriguzzi-Corin R., Millar S., Scott-Hayward S., Martínez-del-Rincón J., Siracusa D. LUCID: A Practical, Lightweight Deep Learning Solution for DDoS Attack Detection // IEEE Transactions on Network and Service Management. — 2020. — Vol. 17, № 2. — P. 876–889. — DOI: 10.1109/TNSM.2020.2971776.

5. Musumeci F., Fidanci A. C., Paolucci F., Cugini F., Tornatore M. Machine-Learning-Enabled DDoS Attacks Detection in P4 Programmable Networks // Journal of Network and Systems Management. — 2022. — Vol. 30. — Art. 21. — DOI: 10.1007/s10922-021-09633-5.

6. Mirsky Y., Doitshman T., Elovici Y., Shabtai A. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection // Network and Distributed System Security Symposium (NDSS). — 2018. — DOI: 10.14722/ndss.2018.23204.

7. Chandola V., Banerjee A., Kumar V. Anomaly Detection: A Survey // ACM Computing Surveys. — 2009. — Vol. 41, № 3. — Art. 15. — DOI: 10.1145/1541880.1541882.

8. Lundberg S. M., Lee S.-I. A Unified Approach to Interpreting Model Predictions // Advances in Neural Information Processing Systems. — 2017. — Vol. 30. — URL: https://papers.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html (дата обращения: 13.11.2026).

9. Gama J., Žliobaitė I., Bifet A., Pechenizkiy M., Bouchachia A. A Survey on Concept Drift Adaptation // ACM Computing Surveys. — 2014. — Vol. 46, № 4. — Art. 44. — DOI: 10.1145/2523813.

10. Sharafaldin I., Lashkari A. H., Hakak S., Ghorbani A. A. Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy // 2019 International Carnahan Conference on Security Technology. — 2019. — P. 1–8. — DOI: 10.1109/CCST.2019.8888419.

11. Sommer R., Paxson V. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection // 2010 IEEE Symposium on Security and Privacy. — 2010. — P. 305–316. — DOI: 10.1109/SP.2010.25.

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, № 3. — Art. e0118432. — DOI: 10.1371/journal.pone.0118432.

13. Guo C., Pleiss G., Sun Y., Weinberger K. Q. On Calibration of Modern Neural Networks // Proceedings of the 34th International Conference on Machine Learning. — 2017. — Vol. 70. — P. 1321–1330. — URL: https://proceedings.mlr.press/v70/guo17a.html (дата обращения: 13.11.2026).

14. Ahuja N., Singal G., Mukhopadhyay D., Kumar N. Automated DDoS Attack Detection in Software Defined Networking // Journal of Network and Computer Applications. — 2021. — Vol. 187. — Art. 103108. — DOI: 10.1016/j.jnca.2021.103108.

15. Breiman L. Random Forests // Machine Learning. — 2001. — Vol. 45. — P. 5–32. — DOI: 10.1023/A:1010933404324.