УДК 004.42:004.75 doi: 10.15827/0236-235X.151.484-498 2025. Т. 38. № 3. С. 484–498

Подход к обнаружению DDoS-атак в сети ЦОД с использованием сочетания графов и импульсной нейронной сети

Е.В. Пальчевский [1]✉, В.В. Антонов [2], Д.А. Петросов [1]

[1] Финансовый университет при Правительстве Российской Федерации, г. Москва, 125167, Россия
[2] Уфимский университет науки и технологий, г. Уфа, 450076, Россия

Ссылка для цитирования
Пальчевский Е.В., Антонов В.В., Петросов Д.А. Подход к обнаружению DDoS-атак в сети ЦОД с использованием сочетания графов и импульсной нейронной сети // Программные продукты и системы. 2025. Т. 38. № 3. С. 484–498. doi: 10.15827/0236-235X.151.484-498
Информация о статье
Группа специальностей ВАК: 2.3.1, 2.3.6
Поступила в редакцию: 30.09.2024 После доработки: 30.03.2025 Принята к публикации: 14.04.2025

Аннотация. В данной статье предложен комплексный подход к выявлению и блокировке DDoS-атак в сетях центров обработки данных (ЦОД). Подход основан на сочетании методов графового анализа и импульсной нейронной сети (SNN). Предполагается представление топологии сети в виде ориентированного графа, где узлы соответствуют элементам инфраструктуры (серверы, маршрутизаторы, коммутаторы), а ребра отражают каналы связи между ними. Это позволяет эффективно визуализировать структуру сети, отслеживать изменения в режиме реального времени и выявлять зоны аномальной активности. Для дальнейшего анализа собранные метаданные сетевого трафика преобразуются в вектор признаков, который подается на вход SNN. Такая нейронная сеть обучается на исторических данных о сетевых взаимодействиях и способна прогнозировать поведение трафика с учетом его временных паттернов. При обнаружении признаков DDoS-атаки модель формирует правила для утилиты iptables, указывая, какой трафик следует заблокировать (DROP), а какой разрешить (ACCEPT). Данный механизм автоматической фильтрации позволяет значительно сократить задержку между моментом появления аномалий и принятием защитных мер, повышая тем самым общий уровень безопасности ЦОД. В ходе экспериментов была проведена оценка предложенного решения в сравнении с популярными моделями глубокого обучения (CNN, LSTM, GAN). Результаты показали, что SNN достигает точности обнаружения в 89,12 %, что выше, чем у CNN (82,14 %) и LSTM (83,00 %). Кроме того, доля ложных срабатываний (10,88 %) существенно ниже по сравнению с альтернативными методами. Таким образом, использование графовых методов и SNN совместно с автоматизированной генерацией правил iptables обеспечивает эффективное, масштабируемое и своевременное выявление DDoS-атак в сетях ЦОД, снижая риск перегрузки оборудования и минимизируя финансовые потери от простоев.
Ключевые слова: обнаружение DDoS-атак, центры обработки данных, графовые методы, импульсная нейронная сеть, модели глубокого обучения, кибербезопасность, мониторинг сетевого трафика
Благодарности. Статья подготовлена в рамках государственного задания № 1023032800215-3-1.2.1 Финансового университета при Правительстве Российской Федерации

Введение. В эпоху стремительной цифровизации *центры обработки данных* (ЦОД) стали опорой критической инфраструктуры: только за 2024 г. европейские и российские организации обработали через собственные и колокационные площадки свыше 83 млрд мгновенных транзакций, а 74 % этого потока пришлось на ЦОД уровня Tier III и выше. В промышленном секторе картина схожая: согласно IEEE, 70 % предприятий Индустрии 4.0 вынесли системы управления производственными процессами и системы сбора и управления данными SCADA именно в дата-центры, что позволило сократить среднее время незапланированных остановок оборудования на 17 % за три года [1, 2].

Бурное развитие облачных сервисов и IoT-экосистемы многократно увеличило вычислительную и сетевую нагрузку. Более того, американская компания Gartner зафиксировала рост объема данных, проходящих через корпоративные площадки, с 16 Збайт в 2020 году до 38 Збайт в 2024 году, при этом доля микросервисного трафика превысила 60 %. Одновременно ENISA зарегистрировала 129 DDoS-кампаний мощностью свыше 1 Тбит/с, 42 % из которых были многовекторными и целились именно в промышленные и финансовые (банковские) ЦОД [3–5].

Для противодействия таким атакам необходимо их раннее выявление и предотвращение на основе современных методов мониторинга и анализа аномалий. Однако классические пороговые и статистические решения, а также стандартные модели машинного и глубокого обучения зачастую не справляются с распределенными и сложными угрозами, когда вредоносный трафик маскируется под легитимный. Учитывая динамичность и масштабность сетей ЦОД с множеством взаимодействующих компонентов (серверов, маршрутизаторов и т.п.), требуются нестандартные подходы, например, импульсные нейронные сети (*Spiking Neural Networks*, SNN). Они способны обучаться на данных о сетевом трафике и предсказывать аномальные события, основываясь на выявленных паттернах [6–8], что позволяет эффективно обнаруживать и даже прогнозировать потенциальные атаки в режиме реального времени.

Идея настоящего исследования заключается в представлении сетевых взаимодействий в ЦОД в виде ориентированных графов, где узлы представляют собой сетевые устройства, а ребра – каналы связи между ними. Такой подход позволяет визуализировать сетевые взаимодействия, находить и анализировать аномалии в сетевом трафике. Графовые методы анализа сетевого трафика с использованием ИИ позволяют не только отслеживать изменение трафика в реальном времени, но и выявлять закономерности и аномальные паттерны, указывающие на потенциальную угрозу. Например, увеличение интенсивности трафика на отдельных узлах может свидетельствовать о подготовке к DDoS-атаке, а изменения в структуре маршрутизации могут указывать на вредоносные действия внутри сети. При этом преимуществами графового подхода анализа сетевого трафика являются его универсальность и возможность применения к различным типам сетей, а также эффективное включение в него искусственных нейронных сетей.

Новизной данного исследования является сочетание графовых методов и SNN. В отличие от классических моделей (*Convolutional Neural Networks* – CNN [9, 10], *Long Short-Term Memory* – LSTM [11, 12]) SNN более точно учитывает временные паттерны сетевого трафика за счет механизма спайков, кроме того, она способна обрабатывать динамические изменения в топологии сети. Это обеспечивает более эффективное и раннее обнаружение DDoS-атак, особенно когда вредоносный трафик распределен по множеству источников. Практическая значимость предложенного подхода заключается в том, что его можно интегрировать в существующие системы мониторинга ЦОД, позволяя в реальном времени блокировать подозрительные соединения и таким образом снижать риск перегрузки сети. Существенные выгоды от подобной интеграции включают улучшение времени реакции на аномальные всплески трафика, сокращение простоев систем и минимизацию финансовых потерь вследствие атак. Предложенное решение масштабируется под различные сценарии – от крупных корпоративных ЦОД до распределенных облачных платформ. Использование динамических графов, отражающих текущие изменения сетевого трафика, в сочетании с обучающейся на лету SNN, повышает точность детектирования и уменьшает число ложных срабатываний, что особенно важно в условиях высоких нагрузок.

Таким образом, целью данного исследования является предложение нового подхода к обнаружению аномалий в сетевом трафике ЦОД с использованием графов, математических методов анализа и SNN. В работе будут рассмотрены различные типы графов и алгоритмов, применяемых для анализа сетевых взаимодействий, включая алгоритмы Дейкстры, Беллмана – Форда, а также модели глубокого обучения с их последующим сравнительным анализом.

Основные задачи исследования:

– построение модели сети ЦОД в виде ориентированного графа с учетом динамики сетевых взаимодействий;

– применение алгоритмов поиска кратчайших путей для анализа структуры сети и выявления участков с аномальными характеристиками;

– создание модели динамического графа, способного адаптироваться к изменениям в сетевой структуре и предсказывать возможные угрозы на основе накопленных данных;

– применение и адаптация SNN, описанной в предыдущих исследованиях [6–8], для обнаружения DDoS-атак.

Обзор существующих решений

В современных сетях традиционно применяют широкий спектр решений для защиты от DDoS-атак, включая аппаратные и программные файрволы, статические системы IDS/IPS, механизмы управления потоком трафика rate limiting и списки блокировки подозрительных IP-адресов. Однако такие подходы чаще всего основаны на жестко заданных пороговых значениях (сигнатурах), что может приводить к высокому числу ложных срабатываний, недостаточной гибкости и неспособности быстро адаптироваться к новым и нетривиальным сценариям атак. Кроме того, сложность современных многоуровневых и распределенных атак нередко превышает возможности ресурсно ограниченных систем, не рассчитанных на анализ высокопоточного трафика в реальном времени. Эти ограничения стимулируют разработку более совершенных методов детектирования и противодействия, основанных на интеллектуальном анализе трафика и гибких моделях машинного обучения. Особое внимание уделяется эволюции технологий – от сигнатурных способов до предиктивных моделей, которые прогнозируют атаки в режиме реального времени.

Как показано в [13, 14], трансформерные модели, такие как DistilBERT и PruneBERT, эффективно применяются для анализа сетевого трафика, повышая точность детектирования DDoS-атак. Однако ключевым ограничением данного подхода остается высокая вычислительная сложность: использование таких моделей требует значительных ресурсов, что осложняет их масштабирование в системах реального времени.

В [15] описывается программное приложение, реализующее теоретико-игровую модель «защитник – нарушитель» от DDoS-атак на основе системы Syncookied. Данное приложение демонстрирует эффективность против SYN-флуда за счет технологии SYN-куки, благодаря чему можно обрабатывать до 300–350 тыс. пакетов в секунду. Интеграция теоретико-игровой модели (равновесие Нэша) обеспечивает выбор оптимальных стратегий защиты, адаптируясь к действиям злоумышленника. Недостатки включают ограниченную поддержку параметров TCP (например, только 4 значения MSS), игнорирование части опций протокола, зависимость от корректной настройки пороговых значений (E1, E2) для фильтрации (это может стать причиной ложных срабатываний или пропуска атак), а также активацию защиты только при обнаружении атаки, что требует точного мониторинга. Кроме того, вычислительная модель на основе теории игр предполагает идеализированные условия распределения трафика, при этом возможно снижение эффективности в реальных динамичных средах с адаптивными атаками.

Исследование [16] посвящено разработке адаптивной модели глубинного обучения для обнаружения и различения DDoS-атак и flash-событий, которая использует методы уменьшения размерности и оптимальный отбор признаков с помощью гибридного алгоритма WGMHGSO. Модель объединяет DCTN для извлечения пространственных признаков и TRANS-LSTM в моделировании долгосрочных временных зависимостей. Основными преимуществами являются высокая точность, адаптивность к изменяющимся условиям трафика и возможность учитывать сложные паттерны атак. К недостаткам относятся высокая вычислительная сложность и необходимость тщательной настройки гиперпараметров, что может усложнить ее практическое применение.

В [17] разработана модель мониторинга DDoS-атак в режиме реального времени в сетях edge-computing, основанная на сотрудничестве между edge-серверами и IoT-устройствами для оперативного выявления атак посредством анализа расстояния между признаковыми векторами. Модель использует алгоритмы кластеризации (k-means) и классификации (k-NN) с пороговыми значениями, что позволяет отделять нормальный трафик от атакующего и автоматически обновлять обучающие модели. Основными преимуществами являются сокращение времени обучения и хранения модели, а также эффективное разделение атакующего и нормального трафика в реальном времени. К недостаткам относятся сложность точной настройки пороговых значений, зависимость точности от качества исходных данных и потенциальные трудности адаптации модели к новым типам атак.

Исследование [18] посвящено разработке эффективной модели обнаружения и предотвращения DDoS-атак в секторе FANET с использованием гибридного подхода, объединяющего глубокие временные сверточные сети и LSTM, а также оптимизированного гибридного алгоритма HWSCS для настройки гиперпараметров и оптимального отбора признаков. Модель не только выявляет DDoS-атаки в реальном времени, но и предотвращает их воздействие посредством переадресации трафика с использованием оптимального маршрутизирования. Среди основных преимуществ – оперативность реакции, эффективное распределение ресурсов и высокая точность обнаружения, которая, однако, может приводить к избыточной чувствительности и ложноположительным срабатываниям. Кроме того, сложность вычислительных процессов, необходимость тонкой настройки параметров и интеграция нескольких алгоритмов могут затруднить ее практическое применение.

Современные гибридные модели и методы анализа сетевого трафика, хотя и обеспечивают высокую точность и скорость реакции, имеют несколько общих недостатков: повышенная вычислительная сложность, сложность настройки гиперпараметров, привязка к конкретным наборам признаков и затрудненная адаптация к новым типам атак. Эти ограничения особенно заметны в высоконагруженных сетях ЦОД, где критичными факторами остаются масштабируемость, низкая задержка и способность к своевременному выявлению угроз.

Смягчить данные проблемы позволяют SNN. Во-первых, они работают на основе событий, что упрощает обработку пространственно-временных паттернов и позволяет избежать чрезмерной нагрузки при высоком потоке трафика. Во-вторых, благодаря более точному моделированию биологической активности нейронов снижается риск ложных срабатываний при анализе зашумленного или неполного набора данных. Кроме того, асинхронный принцип вычислений SNN, по сравнению с классическими моделями deep learning (CNN, LSTM), способствует более гибкой адаптации к новым атакам и потенциально снижает энергопотребление при масштабировании в условиях ЦОД. Все это делает SNN перспективным направлением при создании систем мониторинга и фильтрации DDoS-атак.

Представление сети ЦОД как ориентированного графа

Сеть, в которой может возникнуть DDoS-атака, можно представить как ориентированный граф $G = (V, E)$, в котором $V = \{v_1, v_2, \ldots, v_n\}$ – множество узлов (серверы, маршрутизаторы, коммутаторы, компьютеры); $E = \{e_1, e_2, \ldots, e_m\} \subseteq V \times V$ – множество ребер, представляющих соединения между узлами (каналы связи). В данном случае вызывают интерес свойства узлов (объектов) в графе, описывающие наличие проходящих через них маршрутов, то есть свойства связности узлов. Математически множество подобных описаний, содержащее описание правил функционирования системы на таком графе, представляет собой БД исследуемой предметной области. Очевидно, что для разных объектов однородные описания могут иметь свойства нечеткого соответствия друг другу, и в этой базе должны быть учтены множества правил их сопоставления, что позволяет интерпретировать такое математическое описание как некоторое приближение к цифровому двойнику сети ЦОД. Это открывает путь к выявлению в ней аномалий в сетевом трафике (DDoS-атак). Причем по свойствам объекты такой системы (серверы, коммутаторы, маршрутизаторы и т.п.) как квалифицированные типы, так и не имеющие такой квалифицированности объединены в сложную техническую систему, в которой наряду со свойствами отдельных объектов присутствуют и свойства на уровне всей системы (эмерджентность). Это приводит к появлению неучтенных состояний и свойств сетевых потоков. С математической точки зрения такое описание представляется алгоритмом, учитывающим следующее.

1. Каждому типу объектов ставится в соответствие множество признаков, имеющих значимые для исследования параметры. Эффективно реализовать данную часть алгоритма можно при помощи метода семантических дифференциалов.

2. Из множества признаков производится семантический отбор ограниченного количества, без которых невозможно функционирование системы. Эффективно реализовать данную часть алгоритма можно за счет применения одного из методов экспертных оценок, например, метода разъезженного пути Шульце [19].

3. Используя полученные в предыдущем пункте признаки, можно формировать базис информационного пространства для каждого объекта – узла сети. Значения базисных векторов должны учитывать описанную выше нечеткость сопоставления и могут быть определены с помощью математического аппарата нечеткой логики путем построения функций принадлежности одного множества другому. Следовательно, если обозначить множество нечетких признаков двух объектов как E_j и E_k, то соответствующая функция принадлежности $\mu_{E_j}(E_k)$ может быть определена как $\mu_{E_j}(E_k) = \dfrac{|E_k \cap E_j|}{|E_k \cup E_j|}$. Определенную таким образом функцию принадлежности будем использовать для построения лингвистических переменных.

4. Зафиксируем сетевой поток каждого объекта, в результате чего синтаксическая процедура, позволяющая оперировать элементами множества значений в данном сетевом потоке, становится константой и может не учитываться.

5. Семантические пространства объектов S_i могут быть представлены как

$$S_i = \left\{a_i, T^i(a_i), U, M_i\right\}, \quad (1)$$

где a_i – имя лингвистической переменной объекта; $T^i(a_i)$, U, M_i – стандартное описание лингвистической переменной, то есть множество всех возможных значений рассматриваемых объектов; $i = 1, ..., L$ – номер объекта в исследуемой системе.

Учитывая, что лингвистическая переменная a_j описывает в разных объектах однотипные признаки, она будет принадлежать сразу нескольким пространствам:

$$S_i = \langle a_j, T^i(a_j), U, M_i \rangle,\quad i = 1, \ldots, L. \tag{2}$$

Рассматривая возможную аномалию, то есть DDoS-атаку, как сетевой поток данных от множества источников к одному или нескольким объектам, можно поставить ей в соответствие ребра в графе, соединяющие аналогичные узлы. Это приводит к задаче анализа свойств некоторого пространства, например, атакуемого объекта, в виде пересечений вызывающих интерес пространств S_i, $i = 1, \ldots, L$, что ведет к появлению нового пространства, содержащего только их общую часть:

$$S^{a_j} = \bigcap_{i=1}^{L} S_i = \bigcap_{i=1}^{L} \{a_j, T^i(a_j), U, M_i\}. \tag{3}$$

Для решения задачи нечеткого сопоставления похожих признаков нередко разрабатывается система сопоставимых справочников – конверторов, реализующих нечеткое соответствие $T^i(a_j)$ для всех значений i. Таким образом, для каждого исследуемого объекта, входящего в число образующих ранее указанное семантическое пространство, должно быть определено хотя бы одно понятие, описывающее объект с ненулевой степенью соответствия и открывающее возможность разделения понятий в разных пространствах и объединения их за счет семантически близких терминов в новом пространстве (рис. 1) [20].

Теперь можно смоделировать алгоритмы на графах, описывающих состояние системы и детализирующих структуру связей объекта, относящихся к формализованным семантическим признакам. Для маршрутизации трафика между двумя узлами v_s (источник) и v_t (назначение, то есть приемник) часто применяется задача поиска кратчайшего пути. Это можно описать как задачу минимизации суммы весов ребер между этими узлами:

$$P_{st} = \sum_i \sum_j w_{ij},\ (v_i, v_j) \in P_{st}.$$

Пусть P_{st} – путь от узла v_s до узла v_t, тогда задача состоит в нахождении такого пути, который минимизирует сумму весов, характеризующих пропускную способность каналов-ребер матрицы связности. Для нахождения кратчайших путей между двумя узлами v_i и v_j в сети можно применять алгоритмы Дейкстры (при неотрицательных весах) или Беллмана – Форда (в случае, если некоторые ребра могут иметь отрицательные веса). Эти методы позволяют выявить минимальный маршрут, что важно, например, для определения наиболее загруженных сегментов сети или для расчета задержек в реальном времени. В то же время в практических задачах мониторинга сетей ЦОД зачастую

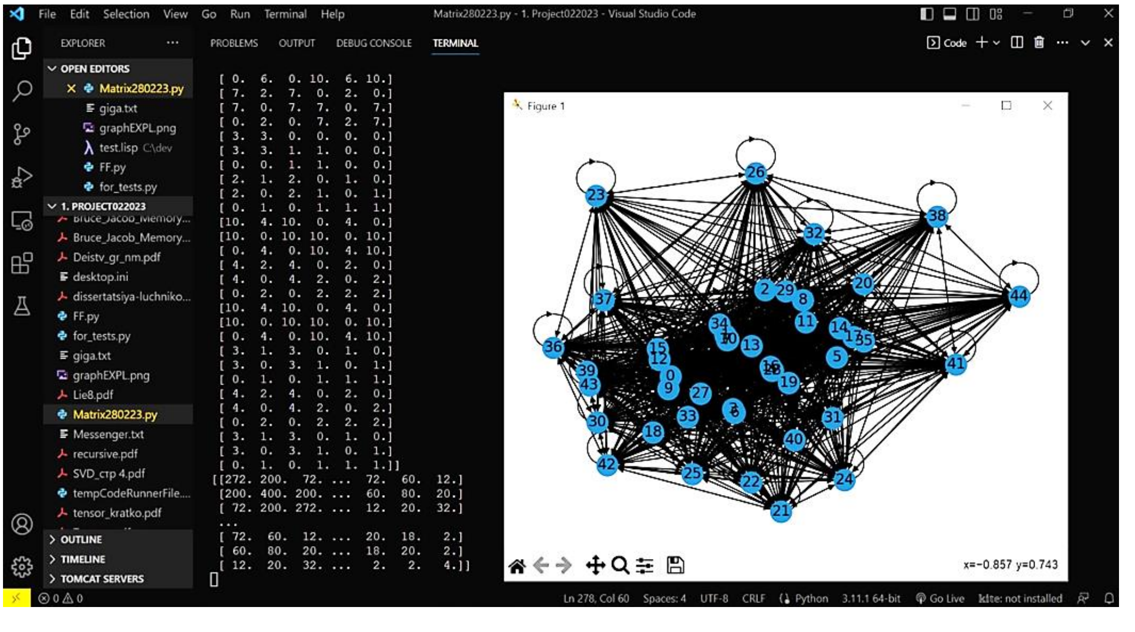

Рис. 1. Ориентированный граф сети ЦОД

Fig. 1. Directed graph of the DPC network

необходимо выделить остовную структуру ориентированного графа, например, для упрощенной визуализации или анализа связности с учетом заданного (выделенного) ребра. В неориентированных сетях эту задачу решают построением каркаса минимального веса. Для ориентированных графов аналогом служит понятие остовного дерева, которое можно получить при помощи алгоритмов типа Чу – Лю.

Поиск аномалий в сетевом трафике ЦОД

Далее приходим к отображениям в себя, к сужению построения проекций объектов, представленных в виде векторов, в новое интегрированное пространство – квазипроекцию. Данное пространство формируется с учетом правил формулирования признаков (мер), содержащихся в свойствах объектов, описанных выше. И учитывая принадлежность объектов, точнее их квазипроекций единому пространству, можно проводить сравнения деревьев графа (проекций) с другими и выделять неявные связности за счет алгебры Ли. Фактически каждое дерево графа (квазипроекция объекта) можно отобразить в вектор признаков в данном интегрированном пространстве. Такой вектор представляет собой семантическое описание состояния узла или группы взаимодействий в сети. Далее эти векторы признаков используются в качестве входных данных для модели машинного обучения. В частности, импульсная нейронная сеть получает на вход именно такие нормализованные векторы, сформированные на основе свойств графа. Это означает, что семантическое пространство графовых признаков непосредственно интегрировано в модель нейронной сети: сходные по признакам состояния сети вызовут схожую реакцию группы нейронов, тогда как сильно отклоняющиеся (аномальные) состояния приведут к заметному изменению паттерна активации нейронов SNN.

По полученным деревьям путем группирования однородных признаков сформируем модель в виде матрицы смежности, в которой отображаются элементы изучаемой системы и связи между ними – когнитивной карты. Исключаем из построения параллельные ребра и петли. Каждое ребро $e = (v_i, v_j)$ характеризуется весом w_{ij}, который может представлять пропускную способность канала, время задержки передачи данных, текущую загрузку сетевого канала.

Матрица связности для графа будет выглядеть следующим образом:

$$A = \begin{bmatrix} 0 & w_{12} & \dots & w_{1n} \\ w_{21} & 0 & \dots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \dots & 0 \end{bmatrix},$$

где w_{ij} – вес ребра между узлами v_i и v_j.

Заметим, что полученная матрица связности $W = [w_{ij}]$ может быть интерпретирована не только как статическая модель связей сети, но и как матрица весовых коэффициентов в нейронной сети. В предлагаемом подходе именно эта матрица задает структуру связей SNN: каждому узлу графа соответствует нейрон, а каждое ребро с весом w_{ij} выступает в роли синаптической связи от нейрона j к нейрону i. Иными словами, распространение сигнала по графу можно отразить через распространение активности по нейронной сети с матрицей связности W. Формально это означает, что если вектор $S(t) = \left[s_1(t), s_2(t), ..., s_N(t)\right]^T$ описывает активность (спайки) нейронов сети на шаге времени t, то влияние связей определяется произведением $W_s(t)$. Компонента i этого произведения $\sum_{j=1}^{N} w_{ij} s_j(t)$ отражает суммарный вклад всех соседних узлов j в активацию узла i. Таким образом, матрица смежности фактически выступает в роли матрицы синаптических весов, определяющих нейроны (узлы), влияющие друг на друга при распространении импульсов активности.

Приходим к адаптивному динамическому графу за счет упорядочивания по времени описанных выше статических графов, каждый переход между которыми может быть описан математическим аппаратом теории категорий. В динамическом графе существует определенное количество узлов – сетевого оборудования ЦОД (например, маршрутизаторы, коммутаторы, серверы). Каждую цепочку узла можно будет достраивать до квадрата Декарта. На основании проведенных исследований представим ориентированный граф коммутативной диаграммой, где вершины – это объекты, стрелки – морфизмы, а результат композиции морфизмов не зависит от выбранного направленного пути между этими объектами.

Далее введем обозначения: выделим множество свойств каждого объекта, при этом эталонные (полученные после обучения ИНС – *Perfect predictions learned,* Ppl) показатели обозначим как Ppl_i, а текущие (реальные – *Practical forecasts*, Pf) – Pf_i. Декартов квадрат морфизмов $f: Ppl_1 \to O_1$ и $g: Ppl_2 \to O_1$ – это объект O_1 и морфизмы p: $Ppl_i \to Ppl_1$ и q: $Ppl_i \to Ppl_1$ такие, что $f \circ p = g \circ q$ и для любого объекта O_1 и морфизмов $m: O_1 \to Ppl_1$ и n: $O_1 \to Ppl_2$, если $f \circ m = g \circ n$, то существует уникальный морфизм $u: O_1 \to Ppl_i$ такой, что $p \circ u = m$ и $q \circ u = n$, то есть диаграмма (рис. 2) коммутативна.

Коммутативность очень важна для настоящего исследования, так как гарантирован одинаковый результат независимо от последовательности анализа объектов.

Проведем аналогичные рассуждения и для текущих показателей (рис. 3). Декартов квадрат $f': Ppl_1 \to O_1$ и $g': Pf_2 \to O_1$ – это объект O_i. Морфизмы $p': Pf_i \to Pf_1$ и $q': Pf_i \to Pf_2$ такие, что $f \circ p = g \circ q$ и для любого объекта O_1, и для морфизмов $m': O_1 \to Pf_1$ и $n': O_1 \to Pf_2$, если $f' \circ m' = g' \circ n'$, то существует уникальный морфизм. Более того, Pf_i формализуется в виде морфизмов, обладающих следующими свойствами: результат взаимодействия информационных объектов может быть представлен декартовым произведением $Pf_1 \otimes Pf_2$ с морфизмами $p_1': Pf_1 \otimes Pf_2 \to Pf_1$ и $p_2': Pf_1 \otimes Pf_2 \to Pf_2$ такими, что для любого объекта Ppl_i с морфизмами $q': Pf_i \to Pf_1$ и $p': Pf_i \to Pf_2$ существует единственный морфизм (отклонение графа) $n': Pf_i \to Pf_1 \otimes Pf_2$, при котором выделяется диаграмма в форме коммутативного треугольника.

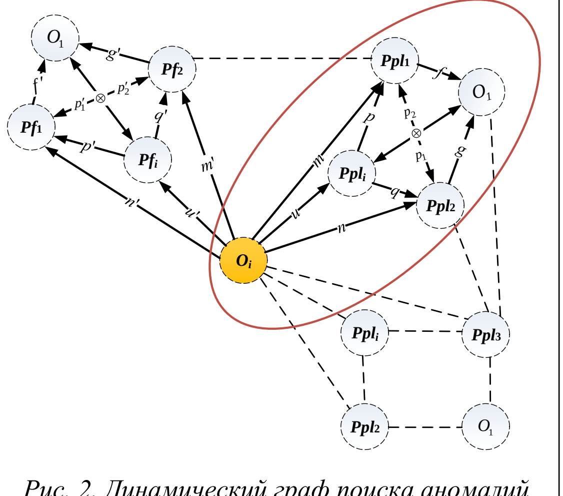

Рис. 2. Динамический граф поиска аномалий трафика в сети ЦОД (эталонные показатели)

Fig. 2. Dynamic graph for traffic anomaly detection in the DPC network (reference metrics)

Важно подчеркнуть, что динамика графа (изменение структуры и весовых коэффициентов во времени) может быть непосредственно учтена в SNN. В данной постановке SNN оперирует не фиксированной матрицей смежности, а последовательностью матриц $W(t) = [w_{ij}(t)]$, зависящих от времени t. Это означает, что по мере появления новых узлов или изменения загрузки каналов (веса w_{ij} увеличиваются или уменьшаются) синаптические связи нейронной сети перестраиваются соответствующим образом. Формально, если $v_i(t)$ обозначает мембранный потенциал нейрона i на шаге t, то можно ввести зависимость входного тока от текущего состояния графа $v_i(t+1) = \beta \cdot v_i(t) + \sum_{j=1}^{N} w_{ij}(t) s_j(t) - L$, где β – коэффициент утечки мембранного потенциала; L – коэффициент утечки/сброса; $s_j(t)$ – индикатор спайка нейрона j на шаге t. Таким образом, изменение весов $w_{ij}(t)$ во времени непосредственно влияет на динамику $v_i(t)$, то есть на рост или спад потенциала нейрона i. Это эквивалентно механизму синаптической пластичности, при котором нейронная сеть адаптируется к текущему состоянию сети ЦОД. Коммутативность графовой модели здесь обеспечивает согласованность: независимо от порядка, в котором анализируются изменения связей, результирующее состояние SNN будет единообразным для данного $W(t)$. В конечном счете аномальные изменения в структуре графа (например, резкое появление нового соединения или аномальный рост веса на определенном ребре) вызовут непропорциональное изменение входного тока некоторых нейронов, что приведет к генерации ими спайков вне характерного паттерна. Это дает нейросети сигнал о возможном наличии аномалии в данный момент времени, то есть решение настоящей задачи сведется к рекурсивному выполнению действий с использованием (3). В результате достраивания до квадрата Декарта каждой цепочки отношений появляется рекуррентный путь. Следовательно, пока система сохраняет свою целостность, он будет уникальным. Таким образом, решается задача прослеживаемости всех объектов системы.

На рисунках 2 и 3 O_i – ключевой элемент критической инфраструктуры ЦОД, то есть центральный маршрутизатор. В рамках пред-

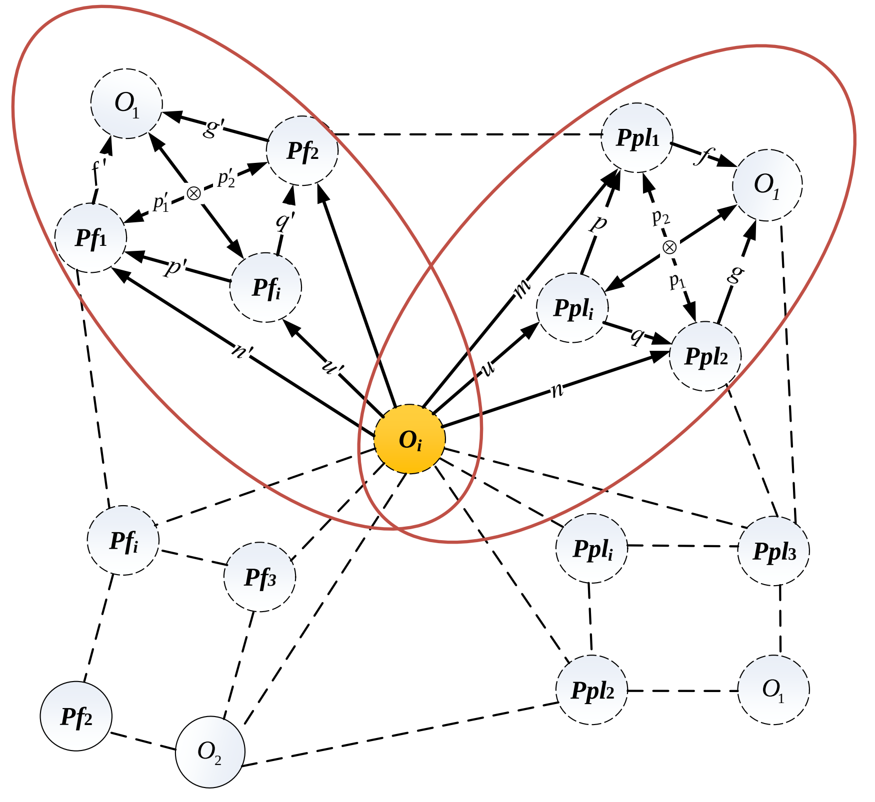

Рис. 3. Динамический граф поиска аномалий трафика в сети ЦОД (текущие показатели)

Fig. 3. Dynamic graph for traffic anomaly detection in the DPC network (current metrics)

ложенного метода возможно применить SNN для анализа сетевого трафика. Благодаря этому можно выявить все объекты сети, в том числе их IP-адреса, содержащиеся в сетевых пакетах, и построить матрицу связности, а также вычислить весовые коэффициенты этой матрицы. Далее возможно выявление/появление новых объектов, обоснованное/необоснованное изменение плотности потоков по ребрам, что отражается на весовых коэффициентах. Рекурсивно выполняя действия согласно (2) и (3), выбирая непустые пространства, а также за счет прослеживаемости можно выявить IP-адрес (объект) атакующего и сетевой маршрут сетевого трафика к сетевому адресу объекта, в ходе чего может быть обнаружена аномалия в сетевом трафике.

Таким образом, математическая модель сети ЦОД в виде динамического графа позволяет выявлять аномалии за счет анализа структуры и ее изменений. Следующим шагом является объединение этого графового подхода с возможностями SNN.

Представим SNN, интегрирующую полученную модель: матрица смежности графа задает структуру связей между ее нейронами, а вычисленные признаки сетевого трафика (семантические координаты узлов в интегрированном пространстве) подаются на ее вход. Такая нейросеть моделирует распространение трафика как процесс распространения спайков по динамическому графу, она обучена распознавать паттерны DDoS-атак. Объединение графа и SNN позволяет учесть одновременно топологические свойства сети и временную динамику трафика, обеспечивая более высокую точность и надежность обнаружения аномалий. Именно за счет этой интеграции достигается точность ~89 % при минимальных ложных срабатываниях, что существенно лучше, чем у традиционных CNN или LSTM.

SNN с интеграцией динамического графа поиска аномалий трафика в сети ЦОД

Исходные данные и признаки. Датасет, на основе которого обучается SNN, содержит сетевые метаданные трафика, включая как числовые, так и категориальные признаки. Фрагмент признаков:

$x_1 \in R^+ : Flow\ Duration$
(*продолжительность сетевого потока*)
$x_2 \in N : Total\ Fwd\ Packets$
(*общее количество входящих сетевых пакетов*)
$x_3 \in N : Total\ Bwd\ Packets$
(*общее количество исходящих сетевых пакетов*)
$x_4 \in \{0, 1, ..., 65535\} : Dst\ Port$ (*сетевой порт*)
$x_5 \in \{0, 1, 2\} : Protocol$ (*протокол кодируется численно*: $TCP = 0,\ UDP = 1,\ ICMP = 2$)
$x_6 \in R^+ : Flow\ Bytes / s$ (*скорость передачи данных в байтах в секунду*)
$x_7 \in R^+ : Flow\ Packets / s$ (*скорость передачи данных в пакетах в секунду*)
$... d = 78.$

Целевая метка представлена соотношением

$$y = \begin{cases} 0 & (Benign, \text{нормальный трафик}), \\ 1 & (Attack, \text{атака}), \end{cases}$$

где y – бинарная метка, указывающая на наличие или отсутствие DDoS-атаки. Прежде чем данные будут поданы в модель, их необходимо предварительно обработать (нормализовать).

Предобработка данных для числовых и категориальных признаков. Для числовых признаков применяется минимально-максимальная нормализация, которая преобразует значения признаков к диапазону [0, 1]:

$$x_i^{norm} = \frac{x_i - \min\left(x_i^{train}\right)}{\max\left(x_i^{train}\right) - \min\left(x_i^{train}\right)},$$

где x_i^{train} – значение признака x_i на обучающей выборке; x_i^{norm} – нормализованное значение признака x_i. Для категориального признака *Protocol* используется численное кодирование:

$$x_5 = ProtocolCode\left(x_5^{raw}\right),$$

где x_5^{raw} – исходное значение протокола (например, TCP, UDP, ICMP); *ProtocolCode* – функция, преобразующая строковое значение протокола в числовое. Далее данные готовы для подачи в модель.

Математическая модель SNN. Архитектура модели состоит из входного, выходного и двух скрытых слоев. Во входном слое данные представляются в виде вектора нормализованных признаков:

$$X = \left[x_1^{norm}, x_2^{norm}, ..., x_d^{norm}\right] \in R^d,$$

где X – вектор входных признаков размерности d; x_i^{norm} – нормализованное значение i-го признака. На первом скрытом слое выполняется линейное преобразование входных данных с последующей батч-нормализацией и дропаутом:

$$h_1 = Dropout\left(BatchNorm\left(W_1 X + b_1\right)\right),$$

где $W_1 \in R^{h_1 \times d}$ – матрица весов скрытого слоя h_1; $b_1 \in R^{h_1}$ – вектор смещений скрытого слоя h_1; $h_1 = 150$; *BatchNorm* – операция, стабилизирующая обучение; *Dropout* – операция, предотвращающая переобучение. Далее применяются LIF-нейроны, моделирующие поведение биологических нейронов:

$$V_1^{(t)} = \beta V_1^{(t-1)} + (1 - \beta) \cdot h_1,$$

$$spk_1^{(t)} = \begin{cases} 1 & \text{если } V_1^{(t)} > V_{th} \left(\text{порог}\right), \\ 0 & \text{иначе}, \end{cases}$$

где $V_1^{(t)}$ – мембранный потенциал нейронов на временном шаге t; $\beta = 0.85$ – коэффициент утечки мембранного потенциала; $V_{th} = 1.0$ – пороговое значение для генерации спайка. На втором скрытом слое выполняется аналогичное преобразование, но с использованием выходов первого слоя:

$$h_2 = Dropout\left(BatchNorm\left(W_2 \cdot spk_1 + b_2\right)\right),$$

где $W_2 \in R^{h_2 \times h_1}$ – матрица весов скрытого слоя h_2; $b_2 \in R^{h_2}$ – вектор смещений скрытого слоя h_2; $h_2 = 75$ – количество нейронов в скрытом слое h_2.

LIF-нейроны:

$$\begin{cases} V_2^{(t)} = \beta V_2^{(t-1)} + (1 - \beta) \cdot h_2, \\ spk_2^{(t)} = \Theta\left(V_2^{(t)} - V_{th}\right), \end{cases}$$

где Θ – пороговая функция (ступенчатая функция Хевисайда). На выходном слое выполняется линейное преобразование:

$$output = W_3 \cdot spk_2 + b_3,$$

где $W_3 \in R^{1 \times h_2}$ – матрица весов выходного слоя; $b_3 \in R^{h_3}$ – смещение выходного слоя. После описания архитектуры модели осуществляется процесс обучения.

Функция потерь и обучение. Для обучения модели используется бинарная кросс-энтропия, которая измеряет разницу между предсказанными и истинными метками:

$$L = -\frac{1}{N} \sum_{i=1}^{N} \Big[y_i \log \sigma\left(output_i\right) + \\ + \left(1 - y_i\right) \cdot \log\left(1 - \sigma\left(output_i\right)\right) \Big],$$

где $\sigma(x) = \dfrac{1}{1 + e^{-x}}$ – сигмоидная функция активации, преобразующая выход модели в вероятность; y_i – истинная метка для i-го примера; $output_i$ – выход модели для i-го примера; N – количество примеров в батче.

Для оптимизации параметров модели используется алгоритм AdamW:

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{\upsilon}_t} + \varepsilon},$$

где θ_t – параметры модели на шаге t; η – скорость обучения; $\hat{m}_t$ и $\hat{\upsilon}_t$ – оценки первого и второго моментов градиентов; ε – малая константа для численной стабильности. После обучения модели можно использовать ее для генерации правил iptables.

Генерация правил iptables. Исходный признак x_4 (порт назначения) был нормализован в процессе предобработки данных. Для генерации правила iptables необходимо вернуть его к исходному значению:

$$Dst\ Port_j^{denorm} = x_4^{norm}\left(\max\left(x_4^{train}\right) - \right.$$
$$\left. - \min\left(x_4^{train}\right)\right) + \min\left(x_4^{train}\right),$$

где x_4^{norm} – нормализованное значение порта для тестового примера X_j; $\max\left(x_4^{train}\right)$ – максимальное значение порта в обучающем наборе данных; $\min\left(x_4^{train}\right)$ – минимальное значение порта в обучающем наборе данных; $Dst\ Port_j^{denorm}$ – денормализованное значение порта, которое будет использоваться в правиле iptables. Далее необходимо декодировать протокол, так как изначально он был закодирован численно в процессе предобработки:

$$Protocol_j^{raw} = InvProtocolCode\left(x_5\right),$$

где x_5 – закодированное значение протокола для тестового примера X_j; $InvProtocolCode$ – функция, обратная кодированию, преобразующая числовое значение протокола в строковое (например, 0 → TCP, 1 → UDP, 2 → ICMP); $Protocol_j^{raw}$ – исходное строковое значение протокола, которое будет использоваться в правиле iptables. На основе предсказания модели и денормализованных и декодированных данных (порт и протокол) формируется правило iptables:

$$iptables - A\ INPUT - p\ Protocol_j^{raw} -$$
$$- dport \left\lfloor Dst\ Port_j^{denorm} \right\rfloor -$$
$$- j\begin{cases} ACCEPT & если\ \sigma\left(output_j\right) > 0.5, \\ DROP & иначе, \end{cases}$$

где $iptables - A\ INPUT$ – команда для добавления правила в цепочку $INPUT$; $Protocol_j^{raw}$ – строковое значение протокола (например, TCP, UDP, ICMP); $\left\lfloor Dst\ Port_j^{denorm} \right\rfloor$ – денормализованное значение порта, округленное до ближайшего целого числа; $\sigma(output_j)$ – сигмоидная функция, применяемая к выходу модели для тестового примера X_j:

$$\sigma\left(output_j\right) = \frac{1}{1 + e^{-output_j}}.$$

При этом если $\sigma(output_j) > 0.5$, то модель считает трафик нормальным, и правило разрешает его (ACCEPT). Если $\sigma(output_j) \leq 0.5$, то модель считает трафик атакой, и правило блокирует его (DROP).

Модификация реализации модели SNN, необходимая для повышения эффективности обучения. Для теоретического описания приведем классическое уравнение LIF, используемое в спайковых сетях. Однако в реализованной экспериментальной части (https://github.com/teelxp/SNN-AntiDDoS) применяется упрощенная одношаговая модель, которая приближенно эмулирует спайковую активность с помощью активации ReLU:

$$spk = ReLU\left(h\right),$$

где h – выход линейного преобразования (например, $h = W_3 \in R^{1\times h_2} X$); $ReLU(x) = \max(0, x)$ – функция активации, заменяющая генерацию спайков. Такое упрощение позволяет эффективно обучать сеть методами обратного распространения ошибки, упростить вычисления, сохраняя базовую идею импульсной (спайковой) архитектуры, а также ускорить сходимость в практических экспериментах.

Как было отмечено, Ppl представляют собой идеальные предсказания, которые модель должна воспроизводить на новых данных. Они служат эталоном для оценки устойчивости модели. Таким образом, можно обозначить эталонными показателями Ppl выходы модели SNN после завершения обучения, соответствующие эталонным данным:

$$output^{train} = W_3 \cdot spk_2^{train} + b_3,$$

где W_3 и b_3 – обученные веса и смещения выходного слоя.

Выходы модели на реальных (тестовых) данных – Pf. При этом Pf отражают способность модели обобщать знания на неизвестные данные. Их согласованность с Ppl, определяемая выражением

$$output^{train} = W_3 \cdot spk_2^{test} + b_3,$$

критична для надежности модели.

Сравнительный анализ моделей обнаружения DDoS-атак с использованием графов

Оценим преимущества и недостатки моделей глубокого обучения для обнаружения DDoS-атак (табл. 1). Поскольку предлагаемое решение основано на использовании SNN, имеет смысл сосредоточиться исключительно на подходах глубокого обучения. В качестве рассматриваемых решений выбраны те модели, которые находятся в открытом доступе.

На основании данных, представленных в таблице 1, можно сделать следующие выводы:

– CNN лучше всего подходят для работы с пространственными данными и сетевым трафиком, но не идеальны для анализа временных зависимостей;

– LSTM и GRU эффективны для анализа временных рядов, при этом LSTM лучше справляется с анализом долгосрочных зависимостей и сетевого трафика, хотя и требует больше ресурсов;

– GAN выделяются своей способностью генерировать данные и обнаруживать аномалии, хотя сложность настройки и высокие требования к ресурсам делают их менее практичными для быстрого развертывания.

В таблице 2 представлено соотношение истинных и ложных срабатываний при обнаружении DDoS-атак и проведено сравнение моделей.

Все модели обучались на одном и том же датасете, включающем [1 048 576] записей трафика (распределение классов: 36,33 % нормальных, 63,67 % DDoS-атак). Для каждой модели использовались одинаковая размерность выборок (80 % – обучающая, 20 % – тестовая), одинаковое количество эпох (120), единый размер батча (4 096). Все метрики вычислялись на одной и той же тестовой выборке. Это обеспечивает корректность прямого сравнения показателей.

Таким образом, на основе анализа данных, представленных в таблице 2, можно сделать вывод, что решение, предложенное авторами статьи, точнее других нейросетевых моделей. Повышение точности способствует обнаружению аномалий в сетевом трафике, что приводит к более эффективной фильтрации DDoS-атак, в конечном итоге снижая нагрузку на ресурсы физических серверов.

Таблица 1

Модели глубокого обучения для обнаружения DDoS-атак (язык программирования Python)

Table 1

Deep learning models for DDoS attack detection (Python)

Модель	Библиотека для реализации	Преимущества модели	Недостатки модели
CNN	PyTorch	Эффективная работа с агрегированными признаками и легкая настройка	Низкое качества анализа временных зависимостей и больших объемов данных
LSTM	TensorFlow	Эффективная работа с большими (длинными) временными рядами, например, с сетевым трафиком за длительный период времени	Потребность в больших вычислительных мощностях
GRU (*Gated Recurrent Unit*) [21, 22]	PyTorch	Эффективная классификация временных зависимостей в сетевом трафике	Чувствительность к аномальным данным в обучающей выборке
GAN (*Generative Adversarial Networks*)	PyTorch	Возможность создания сложных моделей трафика и обнаружения аномалий путем сравнения реального и сгенерированного трафика	Сложность в настройке и обучении, потребность в большом количестве вычислительных ресурсов

Таблица 2

Результаты экспериментов и сравнительный анализ моделей

Table 2

Experimental results and comparative analysis of models

Модель	Detection Rate, %	False Positive Rate, %
SNN (разработанная авторами статьи)	89,12	10,88
CNN	82,14	17,86
LSTM	83,00	17,00
GRU	75,42	24,58
GAN	45,62	54,38

Заключение

В данной статье рассмотрены современные методы обнаружения аномалий в сетевом трафике ЦОД с акцентом на использование графовых методов и моделей глубокого обучения. Проблема DDoS-атак и других угроз, связанных с перегрузкой сетевой инфраструктуры, становится все более актуальной с ростом объемов интернет-трафика и усложнением сетевых взаимодействий. Применение графов для моделирования сетей позволяет эффективно визуализировать и анализировать сетевые связи, выявлять аномальные узлы и маршруты, что способствует своевременному обнаружению угроз.

Использование алгоритмов Дейкстры и Беллмана – Форда для анализа сетевых взаимодействий позволяет выявить участки сети с аномальными характеристиками, а внедрение моделей глубокого обучения, таких как SNN, значительно расширяет возможности прогнозирования и предотвращения DDoS-атак в реальном времени. Более того, динамические графы, адаптирующиеся к изменениям сетевой структуры, открывают новые перспективы для точного анализа и предсказания потенциальных угроз.

Таким образом, проведенное в статье исследование показало, что применение графов для анализа сетевого трафика в ЦОД является эффективным методом для обнаружения аномалий, таких, как DDoS-атаки. Представление сети в виде ориентированного графа позволяет выявлять узлы и ребра с аномальными параметрами, что повышает эффективность в выявлении аномалий в сетевом трафике с использованием ИИ. Разработанная SNN в предыдущих исследованиях продемонстрировала высокую эффективность: точность обнаружения DDoS-атак составила 89,12 %, что выше по сравнению с CNN (82,14 %) и LSTM (83,00 %). Ложные срабатывания при этом были минимальными – 10,88 % для SNN против 17,86 % для CNN и 17,00 % для LSTM.

Кроме того, внедрение динамических графов, которые способны изменяться со временем, позволяет лучше анализировать сетевое взаимодействие и обнаруживать аномальные события. Этот подход дает возможность моделировать сеть с учетом ее эмерджентности и изменчивости.

Эксперименты с моделями глубокого обучения показали, что предложенное решение на основе SNN оказалось более точным по сравнению с классическими методами, такими как CNN, LSTM и GAN. Это делает его перспективным для практического использования в системе мониторинга сетевого трафика ЦОД.

Дальнейшие исследования могут быть направлены на совершенствование алгоритмов и моделей для более точного прогнозирования атак и уменьшения ложных срабатываний, что улучшит защиту сетевой инфраструктуры от угроз.

Список литературы

1. Yoonjib K., Saqib H., Ghorbani A. Detecting Distributed Denial-of-Service (DDoS) attacks that generate false authentications on Electric Vehicle (EV) charging infrastructure. Computers & Security, 2024, vol. 144, art. 103989.

2. Aishatu A.A., Idris I., Olaniyi O.M., Ojeniyi J.A. DDoS attack detection in SDN: Method of attacks, detection techniques, challenges and research gaps. Computers & Security, 2023, vol. 139, art. 103652.

3. Dayal N., Srivastava S. Analyzing effective mitigation of DDoS attack with software defined networking. Computers & Security, 2023, vol. 130, art. 103269.

4. Alatram A., Sikos L.F., Johnstone M. et al. DoS/DDoS-MQTT-IoT: A dataset for evaluating intrusions in IoT networks using the MQTT protocol. Computer Networks, 2023, vol. 231, no. 1, art. 109809.

5. Абрамов А.Г. Защита от DDoS-атак своими руками: оперативные разработка и внедрение сервиса в Национальной исследовательской компьютерной сети России // Программные продукты и системы. 2022. Т. 35. № 4. С. 572–582. doi: 10.15827/0236-235X.140.572-582.

6. Пальчевский Е.В., Христодуло О.И. Разработка импульсной нейронной сети с возможностью скоростного обучения для нейтрализации DDoS-атак // Программные продукты и системы. 2019. Т. 32. № 4. С. 613–627. doi: 10.15827/0236-235X.128.613-627.

7. Пальчевский Е.В., Антонов В.В., Родионова Л.Е., Кромина Л.А. Разработка самообучаемой спайковой нейронной сети для упреждающего реагирования на внешние информационные воздействия различной природы // INJOIT. 2022. Т. 10. № 7. С. 74–85.
8. Palchevsky E., Antonov V., Filimonov N. et al. Development of an intelligent system for early forecasting and modelling of flood situations on the example of the Republic of Bashkortostan using a proprietary machine and deep learning library. J. of Hydrology, 2024, vol. 633, art. 130978.
9. Le T.-T.-H., Heo S., Cho J., Kim H. DDoSBERT: Fine-tuning variant text classification bidirectional encoder representations from transformers for DDoS detection. Computer Networks, 2025, vol. 262, no. 1, art. 111150.
10. Клименкова А.С. Приложение, реализующее систему защиты Syncookied от DDoS-атак с использованием элементов теории игр // МСиМ. 2023. Т. 65. № 1. С. 86–107.
11. Naorem N.D., Khundrakpam J.S. Detecting and differentiating the DDoS and flash events using adaptive deep learning models. Applied Soft Computing, 2025, vol. 171, no. 4, art. 112669.
12. Li M., Zheng L., Ma X., Li S. Real-time monitoring model of DDoS attacks using distance thresholds in Edge cooperation networks. JISA, 2025, vol. 89, art. 103972.
13. Priyadharshini S.P., Balamurugan P. An efficient DDoS attack detection and prevention model using fusion heuristic enhancement of deep learning approach in FANET sector. Applied Soft Computing, 2024, vol. 167, no. 6, art. 112438.
14. Антонов В.В., Куликов Г.Г., Вояковская Я.С., Пальчевский Е.В. Метод формирования структур цифровых двойников предметно-ориентированных объектов пространственно-открытых источников на основе формализмов теории множеств, графов, теории категорий и теории порождающих языков Хомского // Вестн. ЮУрГУ. Сер.: Компьютерные технологии, управление, радиоэлектроника. 2023. Т. 23. № 2. С. 17–27.
15. Вояковская Я.С., Батуров Н.В., Антонов В.В. Программный модуль определения, категоризации объектов и анализа их связей: Свид. о регистр. ПрЭВМ № 2023619771. Рос. Федерация, 2023.
16. Najar A.A., Naik S.M. Cyber-secure SDN: A CNN-based approach for efficient detection and mitigation of DDoS attacks. Computers & Security, 2024, vol. 139, art. 103716. doi: 10.1016/j.cose.2024.103716.
17. Pasha M.J., Rao K.P., MallaReddy A., Bande V. LRDADF: An AI enabled framework for detecting low-rate DDoS attacks in cloud computing environments. Measurement: Sensors, 2023, vol. 28, art. 100828.
18. Pandithurai O., Venkataiah C., Tiwari S., Ramanjaneyulu N. DDoS attack prediction using a honey badger optimization algorithm based feature selection and Bi-LSTM in cloud environment. Expert Systems with Applications, 2024, vol. 241, art. 122544.
19. Kumar D., Pateriya R.K., Gupta R.K et al. DDoS detection using deep learning. Procedia Computer Science, 2023, vol. 218, pp. 2420–2429.
20. Mahadik S.S., Pawar P.M., Muthalagu R. Edge-HetIoT defense against DDoS attack using learning techniques. Computers & Security, 2023, vol. 132, art. 103347. doi: 10.1016/j.cose.2023.103347.
21. Aguru A.D., Erukala S.B. A lightweight multi-vector DDoS detection framework for IoT-enabled mobile health informatics systems using deep learning. IS, 2024, vol. 662, art. 120209.
22. Gangula R., Mohan V.M., Kumar R. A comprehence study of DDoS attack detecting algorithm using GRU-BWFA classifier. Measurement: Sensors, 2022, vol. 24, no. 4, art. 100570.

Software & Systems doi: 10.15827/0236-235X.151.484-498 2025, 38(3), pp. 484–498

DDoS attack detection in the Data Processing Center network: Combining graphs and a spiking neural network

Evgeny V. Palchevsky [1]✉, Vyacheslav V. Antonov [2], David A. Petrosov [1]

[1] Financial University under the Government of the Russian Federation, Moscow, 125167, Russian Federation
[2] Ufa University of Science and Technology, Ufa, 450076, Russian Federation

For citation
Palchevsky, E.V., Antonov, V.V., Petrosov, D.A. (2025) ‘DDoS attack detection in the Data Processing Center network: Combining graphs and a spiking neural network’, *Software & Systems*, 38(3), pp. 484–498 (in Russ.). doi: 10.15827/0236-235X.151.484-498

Article info
Received: 30.09.2024 After revision: 30.03.2025 Accepted: 14.04.2025

Abstract. The paper presents an integrated approach for detecting and blocking DDoS attacks in Data Processing Center (DPC) networks. The proposed approach integrates graph analysis methods with Spiking Neural Networks (SNN). The method represents network topology as a directed graph with nodes corresponding to infrastructure elements (servers routers switchboards) and edges indicating communication channels. This enables efficient network structure visualization, real-time change monitoring, and anomaly zone detection. For subsequent analysis, the system transforms gathered network traffic metadata into feature vectors, which then goes into the Spiking Neural Network. The Spiking Neural Network trains on historical network interaction data and predicts traffic behavior by analyzing temporal patterns. Upon DDoS attack identification, the model generates iptables rules and specifies which traffic to block (DROP) and which to permit (ACCEPT). This automated filtering mechanism significantly reduces the latency between anomaly detection and the implementation of protective measures, thereby enhancing overall DPC networks security. The authors evaluated the proposed solution and compared it with established deep learning models (CNN, LSTM, GAN). Experimental results demonstrated that the Spiking Neural Network achieved 89.12 % detection accuracy, outperforming both CNN (82.14 %) and LSTM (83.00 %) models. Moreover, the false positive rate (10.88 %) remains significantly lower than alternative methods. The combined use of graph-based methods and SNNs with automated iptables rule generation enables an effective scalable and early DDoS detection in DPC networks. This approach reduces equipment overload risks while minimizing downtime-related financial losses.

Keywords: DDoS attack detection, data processing centers, graph methods, spiking neural network, deep learning models, cybersecurity, network traffic monitoring

Acknowledgements. The study was completed in terms of the state assignment, project no. 1023032800215-3-1.2.1

References

1. Yoonjib, K., Saqib, H., Ghorbani, A. (2024) ‘Detecting Distributed Denial-of-Service (DDoS) attacks that generate false authentications on Electric Vehicle (EV) charging infrastructure’, *Computers & Security*, 144, art. 103989.

2. Aishatu, A.A., Idris, I., Olaniyi, O.M., Ojeniyi, J.A. (2023) ‘DDOS attack detection in SDN: Method of attacks, detection techniques, challenges and research gaps’, *Computers & Security*, 139, art. 103652.

3. Dayal, N., Srivastava, S. (2023) ‘Analyzing effective mitigation of DDoS attack with software defined networking’, *Computers & Security*, 130, art. 103269.

4. Alatram, A., Sikos, L.F., Johnstone, M. et al. (2023) ‘DoS/DDoS-MQTT-IoT: A dataset for evaluating intrusions in IoT networks using the MQTT protocol’, *Computer Networks*, 231, art. 109809.

5. Abramov, A.G. (2022) ‘DIY DDoS Protection: Operational development and implementation of the service in the national research computer network of Russia’, *Software & Systems*, 35(4), pp. 572–582 (in Russ.). doi: 10.15827/0236-235X.140.572-582.

6. Palchevsky, E.V., Khristodulo, O.I. (2022) ‘Development of a spiking neural network with the possibility of high-speed training to neutralize DDoS attacks’, *Software & Systems*, 32(4), pp. 613–627 (in Russ.). doi: 10.15827/0236-235X.128.613-627.

7. Palchevsky, E.V., Antonov, V.V., Rodionova, L.E. et al. (2022) ‘Development of a self-learning spike neural network for proactive response to external information impacts of different nature’, *INJOIT*, 10(7), pp. 74–85 (in Russ.).

8. Palchevsky, E., Antonov, V., Filimonov, N. et al. (2024) ‘Development of an intelligent system for early forecasting and modelling of flood situation on the example of the Republic of Bashkortostan using a proprietary machine and deep learning library’, *J. of Hydrology*, 633, art. 130978.

9. Le, T.-T.-H., Heo, S., Cho, J., Kim, H. (2025) ‘DDoSBERT: Fine-tuning variant text classification bidirectional encoder representations from transformers for DDoS detection’, *Computer Networks*, 262(1), art. 111150.

10. Klimentkova, A.S. (2023) ‘Application implementing a protection system for Syncookied against DDoS attacks using elements of game theory’, *MSiM*, 65(1), pp. 86–107 (in Russ.).

11. Naorem, N.D., Khundrakpam, J.S. (2025) ‘Detecting and differentiating the DDoS and flash events using adaptive deep learning models’, *Applied Soft Computing*, 171(4), art. 112669.

12. Li, M., Zheng, L., Ma, X., Li, S. (2025) ‘Real-time monitoring model of DDoS attacks using distance thresholds in Edge cooperation networks’, *JISA*, 89, art. 103972.

13. Priyadharshini, S.P., Balamurugan, P. (2024) ‘An efficient DDoS attack detection and prevention model using fusion heuristic enhancement of deep learning approach in FANET sector’, *Applied Soft Computing*, 167(6), art. 112438.

14. Antonov, V.V., Kulikov, G.G., Voyakovskaya, Ya.S., Palchevsky, E.V. (2023) ‘A method for constructing digital twin architectures of domain-specific objects in spatially open environments, based on formalisms from set theory, graph theory, category theory, and Chomsky's generative language theory’, *Vestn. of the SUSU. Ser.: Comput. Technologies, Automatic Control, Radio Electronics*, 23(2), pp. 17–27 (in Russ.).

15. Voyakovskaya, Ya.S., Baturov, N.V., Antonov, V.V. (2023) *Software Module of Definition, Categorization of Objects and Analysis of Their Relations*, Pat. RF, №2023619771.

16. Najar, A.A., Naik, S.M. (2024) ‘Cyber-secure SDN: A CNN-based approach for efficient detection and mitigation of DDoS attacks’, *Computers & Security*, 139, art. 103716. doi: 10.1016/j.cose.2024.103716.

17. Pasha, M.J., Rao, K.P., MallaReddy, A., Bande, V. (2023) ‘LRDADF: An AI enabled framework for detecting low-rate DDoS attacks in cloud computing environments’, *Measurement: Sensors*, 28, art. 100828.

18. Pandithurai, O., Venkataiah, C., Tiwari, S., Ramanjaneyulu, N. (2024) ‘DDoS attack prediction using a honey badger optimization algorithm based feature selection and Bi-LSTM in cloud environment’, *Expert Systems with Applications*, 241, art. 122544.

19. Kumar, D., Pateriya, R.K., Gupta, R.K. et al. (2023) 'DDoS detection using deep learning', *Procedia Computer Science*, 218, pp. 2420–2429.

20. Mahadik, S.S., Pawar, P.M., Muthalagu, R. (2023) 'Edge-HetIoT defense against DDoS attack using learning techniques', *Computers & Security*, 132, art. 103347. doi: 10.1016/j.cose.2023.103347.

21. Aguru, A.D., Erukala, S.B. (2024). 'A lightweight multi-vector DDoS detection framework for IoT-enabled mobile health informatics systems using deep learning', *IS*, 662, art. 120209.

22. Gangula, R., Mohan, V.M., Kumar, R. (2022) 'A comprehence study of DDoS attack detecting algorithm using GRU-BWFA classifier', *Measurement: Sensors*, 24(2), art. 100570.

Авторы
Пальчевский Евгений Владимирович [1],
к.т.н., доцент, старший преподаватель,
teelxp@inbox.ru
Антонов Вячеслав Викторович [2],
д.т.н., профессор,
antonov.v@bashkortostan.ru
Петросов Давид Арегович [1],
к.т.н., доцент,
scorpionss2002@mail.ru

[1] Финансовый университет
при Правительстве Российской Федерации,
г. Москва, 125167, Россия
[2] Уфимский университет науки и технологий,
г. Уфа, 450076, Россия

Authors
Evgeny V. Palchevsky [1],
Cand. of Sci. (Engineering), Associate Professor,
Senior Lecturer, teelxp@inbox.ru
Vyacheslav V. Antonov [2],
Dr.Sci. (Engineering), Professor,
antonov.v@bashkortostan.ru
David A. Petrosov [1], Cand. of Sci. (Engineering),
Associate Professor,
scorpionss2002@mail.ru

[1] Financial University under the Government
of the Russian Federation,
Moscow, 125167, Russian Federation
[2] Ufa University of Science and Technology,
Ufa, 450076, Russian Federation