Разработка самообучаемой спайковой нейронной сети для упреждающего реагирования на внешние информационные воздействия различной природы

Е.В. Пальчевский, В.В. Антонов, Л.Е. Родионова, Л.А. Кромина

***Аннотация* — Цифровизация и интеллектуализация в рамках массового внедрения киберфизических систем «Индустрия 4.0» превратились в настоящий тренд, в котором кибертехнологии обеспечивают автоматизированное и автоматическое управление, большую эффективность и повышенную безопасность. При этом интеграция таких технологий в критически важные инфраструктурные объекты зачастую подвергается киберугрозам, и как следствие – кибератакам, нарушая при этом не только конфиденциальность и целостность данных, но и доступность, например, при помощи DDoS-атак, что говорит о несовершенстве большинства методов фильтрации данных атак на самых различных уровнях модели OSI. Это приводит к тому, что многие организации, физические и вычислительные ресурсы которых имеют выход во внешнюю глобальную сеть Интернет, сталкиваются с недоступностью собственных сервисов, что приводит к невозможности предоставлять необходимые данные и услуги, как для собственных сотрудников, так и для клиентов, что влечет за собой финансовые потери компании от простоя оборудования. Для минимизации потерь от данной проблемы предлагается использовать спайковую (импульсную) нейронную сеть с целью фильтрации атак внешним несанкционированным трафиком (DDoS).**

Основными особенностями предлагаемой нейронной сети являются как высокие скорость и качество (за счет постоянного обучения на больших данных) самообучения, так и быстрое реагирование на DDoS-атаки (в том числе и на те, которые неизвестны), а также структурная зависимость (количество нейронов и слоев импульсной нейронной сети) от физических (вычислительных) ресурсов сервера/кластера. Предложен модифицированный метод вложенных математических моделей самообучения (обучения без учителя) импульсной нейронной сети, в основу которого входит стандартный метод обучения ИНС с обратным распространением ошибки (градиентного спуска), что позволяет импульсной нейронной сети быстро и эффективно обучаться с целью фильтрации атак внешним несанкционированным трафиком.

Проведено длительное тестирование (в реальных условиях на физических серверах кластера) разработанной импульсной нейронной сети в двух режимах: простой (работа кластера в штатном режиме) и в режиме защиты (фильтрации) от DDoS-атак. Тестирование показало достаточно низкую загруженность физических ресурсов кластера как в режиме простоя (CPU – 0.3%, SSD – 0.4%, RAM – 0.5%), так и в режиме защиты от DDoS-атак (CPU – 6.9%, SSD – 4.0%, RAM – 3.1%), что позволяет повысить доступность данных при атаках внешним несанкционированным трафиком без нарушения работоспособности рабочей среды.

***Ключевые слова* — DDoS-атаки, фильтрация DDoS-атак, спайковая нейронная сеть, импульсная нейронная сеть.**

I. Введение

За последние несколько лет прогресс в информационных технологиях достиг рекордных темпов роста, что, в свою очередь, неизменно ведет к процессу глобализации в рамках создания качественно новых систем, включающих в себя синтез решений искусственного и биологического интеллектов, а также физических объектов (например, появление киберфизических систем) для достижения каких-либо целей, особенно в области информационных технологий (в том числе и сетевых технологий). Как правило, традиционные сети требуют больших усилий и знаний для формирования какой-либо единой стратегии управления, в том числе и принятия решений при каких-либо ситуациях (особенно критических). Зачастую подобная ответственность возложена на системных администраторов, поскольку управление объектами, расположенными в одной сети, зачастую реализовывается на одном физическом устройстве. Сложность такого управления заключается в том, что при любом изменении сетевой конфигурации необходима соответствующая настройка связанных между собой устройств, что повышает риски к возникновению проблем в работе объектов сети, влечет за собой локальную и внешнюю глобальную недоступность ресурсов. Более того, есть и проблема того, что традиционным и современным сетям не хватает динамического и автоматического реагирования и повышенной устойчивости при каких-либо аномалиях (например, изменение сетевой нагрузки) в сетях (особенно современных, как следствие – глобальных). Основными факторами, служащими появлению таких аномалий, являются распределенные атаки типа «отказ в обслуживании» (DDoS-атаки). За последние годы DDoS-атаки стали одними из самых доступных (в силу своей дешевизны) и опасных (по причине большого количества вредоносным программным обеспечением зараженных персональных компьютеров и физических

серверов), что делает их одним из самых популярных видов киберугроз. Так, в 2021 году объем внешнего сетевого несанкционированного трафика при выполнении данного вида атак достиг 2,4 Тбит/с (на корпорацию Microsoft) с количеством запросов ~ 20 млн/с (на организацию Яндекс). Исходя из этого необходимо важно четко понимать динамику, тенденцию и классификацию таких атак для разработки алгоритмов с последующей реализацией эффективных методов поддержки принятия решений с целью фильтрации DDoS.

Целью данного исследования является разработка и реализация метода поддержки принятия решений для фильтрации большого объема внешнего сетевого трафика. Основной научной новизной и основными отличиями исследования являются:

− предлагается принципиально новое решение при защите от DDoS, основанное на импульсной нейронной сети с многосетевой структурой, что показало качественные результаты не только при обработке внешнего сетевого трафика, но и при нагрузке во время DDoS-атак;

− для улучшения качества фильтрации DDoS-атак разработан новый вид нейрона в импульсной нейронной сети, основывающийся на модели Фитц Хью-Нагумо.

До сих пор были разработаны модели DDoS-атак, исполняемые на самых различных уровнях модели OSI, основанные на: частичном/полном управлении внешним сетевым трафиком [1]; аналитическом анализе сетевого трафика [2]; иерархии [3] и семантике [4]. Но зачастую данные атаки инициируют лишь на четырех уровнях модели OSI (по причине особенности взаимодействия аппаратной и программной частей с сетевой инфраструктурой): сетевой и транспортный (низкоуровневые атаки), а также сеансовый и прикладной (высокоуровневые атаки). Но важно отметить, что ежедневно появляется множество новых методов DDoS-атак, реагирование и принятие решений на которые необходимо осуществлять четко и быстро для достижения минимизации последствий в рамках той или иной DDoS-атаки. Разумеется, на сегодняшний день существуют множество видов фильтрации DDoS-атак, которые можно подразделить на две категории: программные и программно-аппаратные. Программные средства обеспечения доступности данных можно подразделить на концептуальные (основанные на внедрении специализированных правил фильтрации сетевого трафика) [5–9] и интеллектуальные [10–29]. К программно-аппаратным системам обеспечения доступности информации (защиты от DDoS-атак) можно отнести специализированные маршрутизаторы «NetScoutArbor» [30], а также сетевые кластеры, для управления которыми используется специализированное программное обеспечение [31]. Как видно из обзора, на сегодняшний день преобладают именно интеллектуальные виды защиты от DDoS-атак, зачастую основанные на искусственных нейронных сетях. К основным преимуществам программных средств обеспечения доступности данных можно отнести малые финансовые затраты (по причине отсутствия необходимости закупки специализированного сетевого оборудования) и простоту настройки, в качестве недостатка необходимо отметить достаточно низкое качество фильтрации внешнего сетевого трафика. В случае с программно-аппаратными способами защиты от DDoS-атак все обстоит иначе: качество данных продуктов в рамках фильтрации DDoS-атак остается высоким, но и финансовые затраты на оборудование (например, для создания собственного сетевого узла/кластера) остаются на высоком уровне.

Но существенными минусами приведенных аналогичных решений является отсутствие необходимой гибкости под различные виды DDoS-атак, а также высокая ресурсоемкость и в большинстве случаев – ручное управление для фильтрации сетевого трафика. Решение данных проблем способствовало бы качественному обеспечению полноценной доступности данных для ресурсов, имеющих выход во внешнюю глобальную сеть, что положительно скажется на работоспособности каждого физического сервера в кластере. Наше решение относится к программно-аппаратному виду защит от DDoS-атак. При всем этом импульсные нейронные сети фактически не применяются в рамках защиты от DDoS-атак по причине отсутствия необходимых методов обучения и кодировки данных с внешнего сетевого интерфейса под данную задачу.

Таким образом, возникает необходимость разработки и применения принципиально нового решения в области обеспечения доступности информации при защите от DDoS, основанного на базе импульсной нейронной сети нового (третьего) поколения, что позволит повысить качество фильтрации низко- и высокоуровневых атак внешним несанкционированным трафиком.

II. Схема поддержки принятия решений и постановка задачи для фильтрации DDoS-атак

Одним из основных параметров воздействия на работоспособность физического сервера в сети Интернет является N – внешний сетевой трафик, передаваемый от хоста (h) к клиенту (c), в нашем случае – на физические серверы кластера, за определенный период времени t. В данной связи введем обозначение – внешний сетевой трафик, передаваемый h-хостами c-клиентам за определенный период времени t, при этом $h = 1,...,n$ и $c = 1,...,k$ (n–общее количество хостов, с которых передается трафик, k–количество физических серверов в кластере – клиентов). Задача фильтрации DDoS-атак заключается в том, чтобы в определенный период времени t обработать максимальное количество внешнего сетевого трафика N_c^h для минимизации рисков потери работоспособности клиента c. Решать данную задачу предлагается в четыре этапа, что представлено на рис. 1.

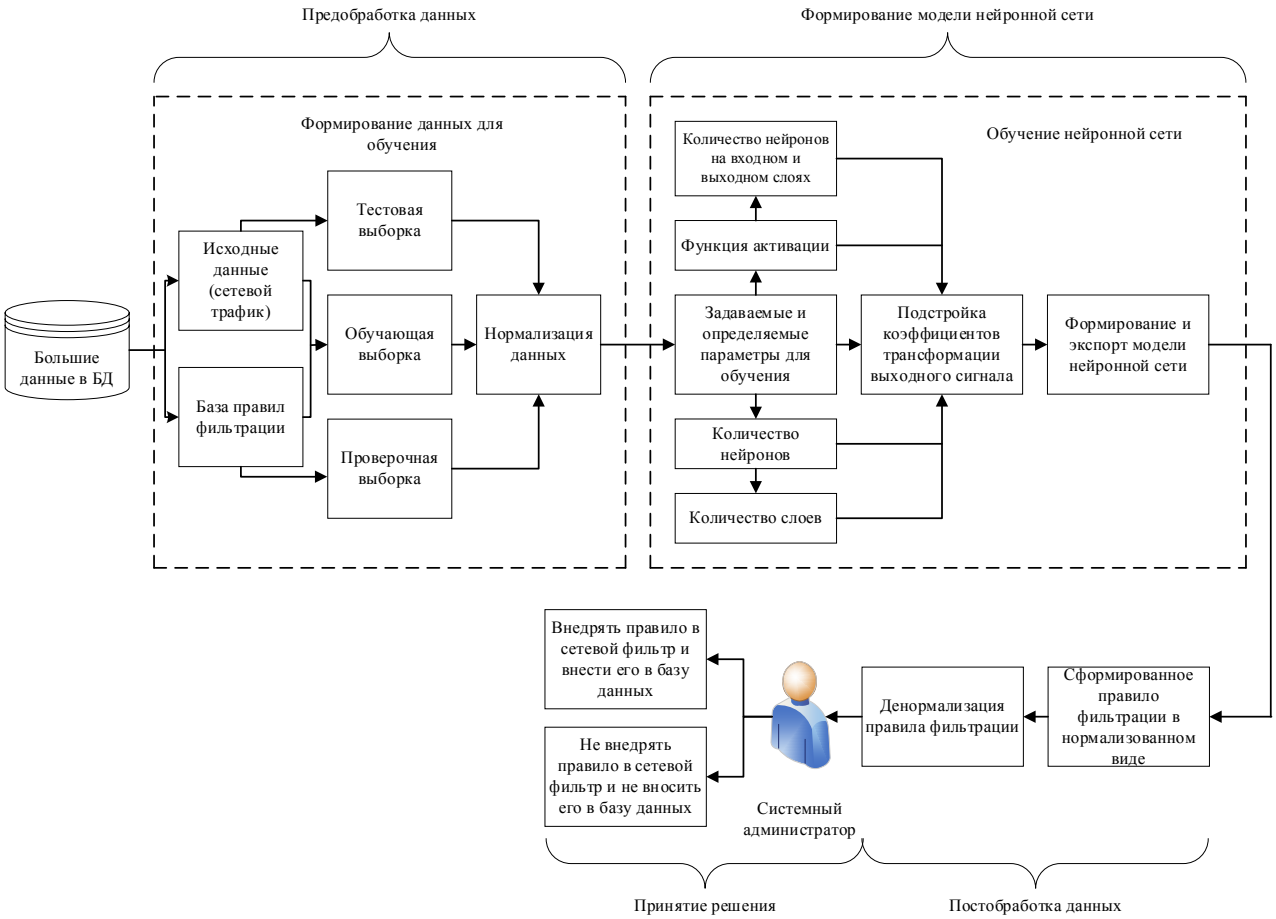

Рис. 1. Схема поддержки принятия решений для защиты от DDoS-атак.

На первом этапе, выполняющемся каждый *i*-ый час (по причине круглосуточного выхода физических серверов кластера во внешнюю глобальную сеть Интернет), т.е. *i*=24 (обучения в сутки), осуществляется автоматический выбор параметров импульсной нейронной сети: наработанная база правил фильтрации; общее количество нейронов ($NINN_{total}$) и слоев (NNL_{total}), количество нейронов на входном и выходном слоях, количество правил фильтрации (подбирается системным администраторов экспериментально).

Общее количество нейронов и слоев в импульсной нейронной сети определяется по формулам (1-5):

$$R_c^{sc} = \begin{cases} R_{ram} = R - R1 \\ R_{cpu} = C - C1 \end{cases}, \quad (1)$$

где R_c^{sc} – количество свободных ресурсов на физическом сервере (*c*) в кластере (*sc*) перед обучением нейронной сети, R_{ram} – количество свободной оперативной памяти физического сервера в кластере перед обучением нейронной сети, *R* – суммарная оперативная память на физическом сервере в кластере перед обучением нейронной сети, R_{cpu} –количество свободных ресурсов CPU физического сервера в кластере перед обучением нейронной сети, *R*1 – используемая оперативная память на физическом сервере в кластере перед обучением нейронной сети, *C*– суммарные ресурсы CPUфизического сервера в кластере перед обучением нейронной сети, *C*1 –используемые ресурсы CPUфизического сервера в кластере перед обучением нейронной сети. Соответственно, под каждый нейрон было отдано 100 кб RAM и установлено ограничение не более 1% нагрузки на CPU на каждом физическом сервере в кластере, исходя из чего $NINN_{total}$ определяется соотношением (2):

$$NINN_{total} = R_{ram} \cdot (9.537 \cdot 10^{-7}). \quad (2)$$

Таким образом, NNL_{total} определяется по соотношению (3):

$$NNL_{total} = NNL_i + NNL_h + NNL_o, \quad (3)$$

где NNL_i – входной слой нейронной сети, NNL_h – промежуточные слои нейронной сети (4-5), NNL_o – выходной слой нейронной сети. Тогда:

$$NINN_{hlk} = NINN_{total} - (NINN_{ilk} + NINN_{olk}), \quad (4)$$

где $NINN_{hlk}$ – количество нейронов скрытых промежуточных слоев нейронной сети, $NINN_{ilk}$ – количество нейронов входного слоя нейронной сети, $NINN_{olk}$ – количество нейронов выходного слоя нейронной сети. Исходя из найденного значения $NINN_{hlk}$ вычисляются NNL_h :

$$NNL_h = \frac{NINN_{hlk}}{NINN_{ilk}}. \quad (5)$$

Количество нейронов на входном и выходном слоях

Количество нейронов (6) на входном слое напрямую зависит от количества подаваемых данных (в нашем случае – от поступающего на внешний сетевой интерфейс трафика *f*) и определяется следующим образом:

$$NINN_{ilk} = \psi\left(N_{np}\right), \quad (6)$$

где N_{np} – количество сетевых пакетов в секунду.

В случае с выходным слоем число нейронов зависит от количества сетевых правил для фильтрации сетевого трафика, которые хотим внедрить в файрвол нашего кластера (одно правило приравнивается к одному нейрону).

Структура разработанной ИмНС представлена на рис. 2.

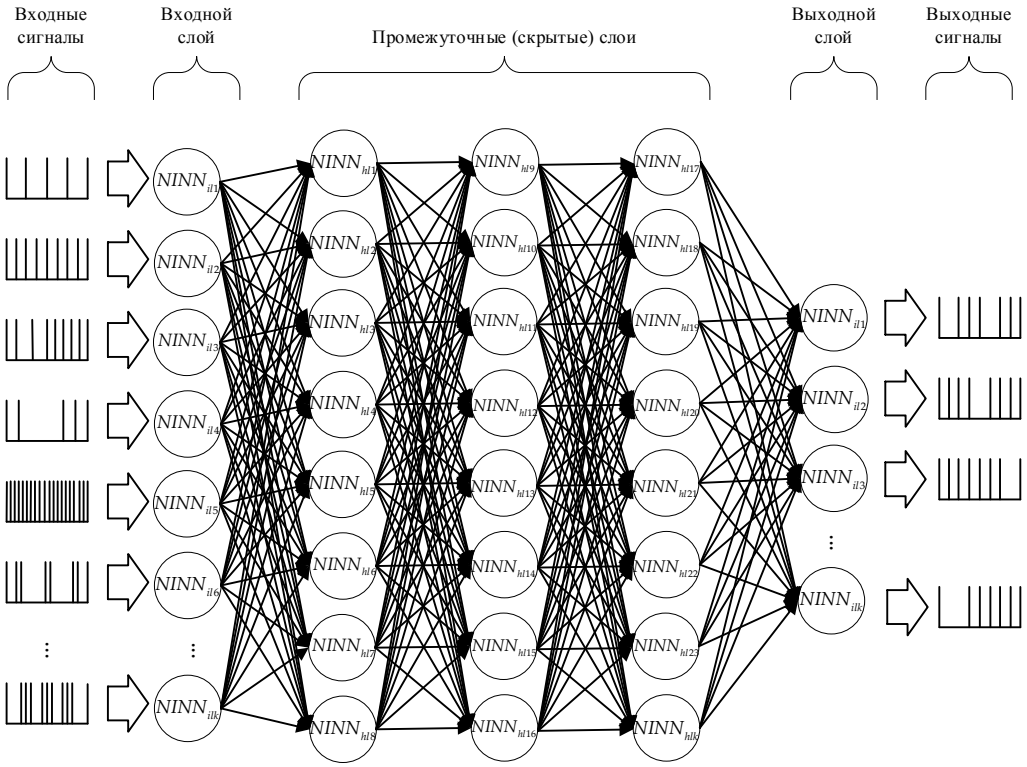

Рис. 2. Структура разработанной ИмНС.

Изначально (рис. 2) данные (сетевой трафик и база правил iptables) считываются с внешнего сетевого интерфейса каждого физического сервера в кластере с дальнейшим кодированием сетевых пакетов для их последующего представления в ИмНС методом порядка следования импульсов (аналог нормализации данных (7)), что представлено на рис. 3: в нашем случае от расположения того или иного спайка зависит скорость передачи данных и их обработка нейронной сетью. Но в связи с тем, что нейронные сети работают со значениями от 0 до 1 более корректно (на выходе получаем более точный результат), нами было принято решение реализовать специализированный кодировщик данных: числовые (таблица 1, формула (7)) и буквенные (в случае IPv6 и правил фильтрации, таблицы 1–2) значения преобразовываются в диапазон от 0 до 1, что представлено в виде импульсов (данных).

$$\overrightarrow{N_{np}} = \frac{\overrightarrow{N_{np}} - \overrightarrow{N_{\min}}}{\overrightarrow{N_{\max}} - \overrightarrow{N_{\min}}}, \quad (7)$$

где $\overrightarrow{N_{\min}}$ и $\overrightarrow{N_{\max}}$ – минимальная и максимальная длины битово-байтовых значений в структуре сетевого пакета по всем данным, подаваемым на входной слой нейронной сети для всех $\overrightarrow{N_{np}}$.

Денормализуем данные по обратной формуле (8):

$$\overrightarrow{N_{np}} = \overrightarrow{N_{np}} \cdot \left(\overrightarrow{N_{\max}} - \overrightarrow{N_{\min}}\right) + \overrightarrow{N_{\min}}. \quad (8)$$

Таблица 1 – Заданные значения для кодировщика с целью дальнейшего представления данных в ИмНС

Значение поля сетевого пакета	Ключевое слово (протокол)	Значение для кодировщика
0	Reserved	0
1	ICMP	0,1
2	IGMP	0,2
4	IP	0,3
6	TCP	0,4
17	UDP	0,5
89	OSPF	0,6

Таблица 2 – Заданные буквенные значения (в случае с IPv6) для кодировщика с целью дальнейшего представления данных в ИмНС

Значение поля сетевого пакета	Значение для кодировщика
a	0,01
b	0,02
c	0,03
d	0,04
e	0,05

После преобразования (кодирования) данных в рамках процесса обучения и фильтрации внешнего сетевого трафика ИмНС в автоматическом режиме

обучается предложенным далее (номер раздела) методом обучения без учителя (самообучение) с целью выработки (генерации) на выходном слое нейронной сети правил фильтрации DDoS-атак с их последующей записью в базу данных.

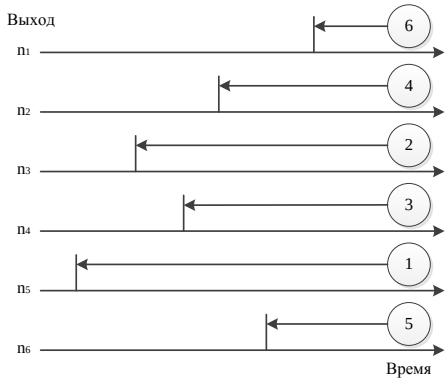

Рис. 3. Порядковый способ представления информации в разработанной ИмНС

III. Теоретико-множественная модель взаимодействия категорий объектов в нейронных сетях

Исходя из раздела 2 для нашей предметной области применима теоретико-множественная модель, представляющая собой многосетевую архитектуру нейронной сети. Правила применения взаимодействий объектов представлены в ранее опубликованной работе [35]. Соответственно, имеем множество нейронных сетей $NN = \{NN_1, ..., NN_n\}$ и множество решений нейронных сетей $R = \{R_1, ..., R_s\}$. Изначально рассмотрим правила взаимодействия объектов нейронной сети в виде квадрата Декарта, далее в виде расслоенного квадрата Декарта (рис. 4). Декартов квадрат морфизмов $f : NN_3 \to NN_n$ и $g : NN_1 \to NN_n$ – это объект NN_2 и морфизмы $p : NN_2 \to NN_3$ и $q : NN_2 \to NN_1$, такие что $f \circ p = g \circ q$ и для любого объекта ИмНС и морфизмов $m : NN \to NN_3$ и $n : NN \to NN_1$, если $f \circ m = g \circ n$, то существует уникальный морфизм $u : NN \to NN_2$, такой что $p \circ u = m$ и $q \circ u = n$, т.е. следующая диаграмма коммутативна.

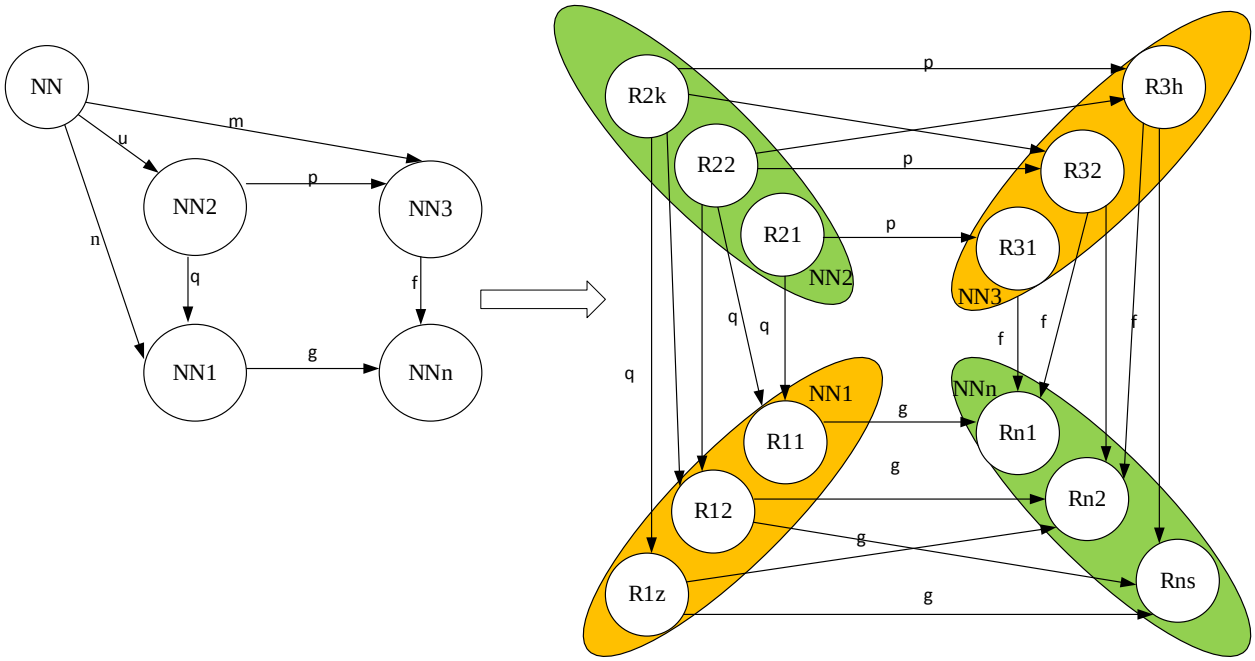

Рис. 4. Структура декартова квадрата

Взаимодействие этих объектов открывает возможность их дальнейшей композиции в виде достраивания. Новые отношения позволяют достраивать объекты до квадрата Декарта. В результате получаем множество объектов, идентифицируемых и прослеживаемых по пространству и времени. На каждом шаге построения указанной нейронной сети выбирается некоторое оптимальное управление в предположении об оптимальности всех последующих шагов, динамика модели поведения определяется модель генерации сигнала в нейроне базе модели Фитц Хью – Нагумо [32, 36–38]. При этом каждую нейронную сеть можно рассмотреть в виде отдельных процессов и отдельной нейронной сети. Возможность системы взаимодействовать с многоуровневой системой, причем общая схема не будет претерпевать существенных изменений.

IV. Структура нейрона

В предлагаемой импульсной нейронной сети за основу берется модель Фитц Хью-Нагумо [38], представляющая собой упрощенную модель, воспроизводящую основные свойства волн возбуждения

в модели Ходжкина-Хаксли и содержащую в себе две переменные: быструю переменную, соответствующую мембранному потенциалу в полной модели, и медленную переменную. Авторами внесены изменения, суть которых состоит в том, что в отличие от оригинальной модели добавлена возможность реагирования каждого *j*-го нейрона на внешние воздействия (в нашем случае — на изменение видов и типов DDoS-атак (раздел 2) и как следствие – на динамику модели (структура нейрона представлена на рис. 5) с нелинейным поведением восстанавливающей переменной.

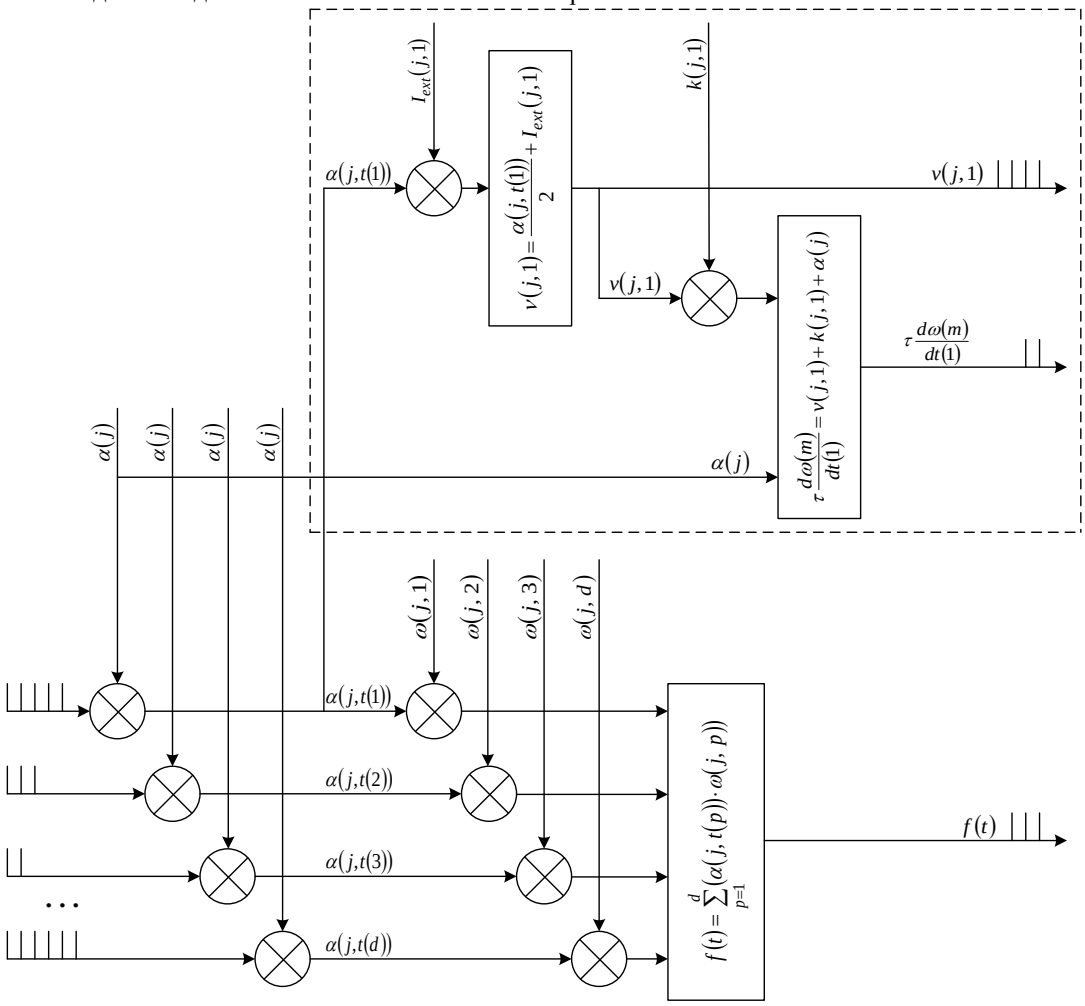

Рис. 5. Структура нейрона в разработанной ИмНС

В модели нейрона (рис. 5): j – порядковый номер нейрона; $j=\left(\overline{1,..,m}\right)$, где *m* – это количество нейронов; $\alpha(j)$ – это начальный потенциал *j*-го нейрона, *p* – порядковый номер импульса, $p=\left(\overline{1,..,d}\right)$, где *d* – это количество импульсов; $\alpha\left(j,t(p)\right)$ – потенциал j-го нейрона во время угасания p-го импульса; $\omega(j,p)$ – коэффициент восстановления j-го нейрона в результате p-го импульса; $f(j)$ – функция активации j-го нейрона:

$$f(j)=\sum_{p=1}^{d}\left(\alpha\left(j,t(p)\right)\cdot\omega(j,p)\right) \quad (9)$$

Данная модель нейрона описывается следующей системой уравнений:

$$\begin{cases} v(j,p)=\frac{\alpha(j,t(p))}{2}+I_{ext}(j,p) \\ \tau\frac{d\omega(m)}{dt(p)}=v(j,p)+k(j,p)+\alpha(j) \end{cases}, \text{при } p=\left(\overline{1,..,d}\right),\ j=\left(\overline{1,..,m}\right), \quad (10)$$

где $I_{ext}(j,p)$ – коэффициент внешнего воздействия p-го импульса на j-й нейрон; $t(p)$ время угасания p-го импульса; $k(j,p)$ –коэффициент реагирования j-го нейрона на p-й импульс; $v(j,p)$ – динамика мембранного потенциала j-го нейрона от воздействия p-го импульса; τ – временная постоянная корреляции шума.

Реагирование на виды и типы DDoS-атак было предложено ранее авторами в статье [39], но в текущем случае отличие заключается в принципиально новой структуре нейрона ИмНС, что позволяет достаточно быстро и в кратчайшие сроки обнаружить атаки внешним несанкционированным трафиком, проанализировать их и отфильтровать с помощью вырабатываемых нейронной сетью правил фильтрации.

V. Обучение импульсной нейронной сети модифицированным для ИмНС методом обратного распространения ошибки

Зачастую в импульсных нейронных сетях используют метод обучения STDP по причине простоты математической и технической реализации, а также достаточно большой эффективности при задачах классификации и кластеризации. Но из-за использования в данном методе исключительно локальных правила обучения основным недостатком является сложность осуществления обратного прохода при обучении на сигнале ошибки. Соответственно, ошибочный сигнал доступен строго на выходном слое импульсной нейронной сети и поток данных, передаваемый внутри нейронной сети, остается однонаправленным. В связи с вышеизложенным предлагается метод обратного распространения для импульсных нейронных сетей. Основная суть заключается в накоплении потенциала в мембране нейрона за счет полученных на вход импульсов (спайков). Исходя из этого, если накопленный потенциал достигает порогового значения (в нашем случае – 1), то нейрон активируется и пропускает импульс (спайк) в следующий слой с последующим уменьшением своего потенциала (обозначим данный процесс как P) до минимального значения (в нашем случае – 0), что описывается уравнениями (11-13):

$$P = \frac{d\alpha(t_d)}{dt_d} = -\alpha(t_d) + I(t), \quad (11)$$

$$I(t) = \sum_{d=1}^{NNIN_{total}} \left(w_d \cdot \sum_{i=1}^{d} (t_1 - t_i) \right), \quad (12)$$

$$t_1 - t_i = \begin{cases} 1, \textit{если } t_1 = t_i \\ 0, \textit{иначе} \end{cases}, \quad (13)$$

где $I(t)$ – взвешенная сумма входных импульсов (спайков) в текущий момент времени (t).

Важно отметить, что группы импульсов представляют собой входные паттерны. Соответственно, чтобы сгенерировать импульсные входы, необходимо их распределить в специализированные спайковые группы по Пуассону [33] и подать в сеть. Таким образом, с течением времени общая взвешенная сумма входных спайковых групп$S(t)$ описывается следующим образом:

$$S(t) = \left(NNIN_{total} - 1 \right) \sum_{i=1}^{d} (t_1 - t_i). \quad (14)$$

При этом утечка потенциала на выходном слое учитывается большой разницей в утечке потенциала в текущей момент времени (t). Исходя из вышеописанного описываем функцию активации $f(t)$:

$$f(t) = \sum_{i=1}^{d} \exp\left(-\frac{t_1 - t_i}{P} \right). \quad (15)$$

Сами весовые коэффициенты рассчитываются по стандартному методу обратного распространения ошибки [34].

Таким образом, предложенный метод существенно увеличивает скорость и точность обучения импульсной нейронной сети, что показано в следующем разделе.

VI. Апробация и сравнительный анализ

В данном разделе представлены результаты апробации, оценка производительности, а также скорость и точность обучения ИмНС в сравнении с другими методами.

Далее в рамках реализации защиты от DDoS-атак была произведена апробация разработанной импульсной нейронной сети в режиме простоя — без DDoS (таблица 3) и при массивных DDoS (таблица 4).

Таблица 3. Потребление ресурсов в режиме простоя

Ситуация	**День эксперимента (p_i)**									
	1	2	3	4	5	6	7	8	9	10
	$N_c^h(t)$, гбит/сек									
	1,0	1,2	1,8	2,0	2,5	2,6	2,7	2,8	2,9	3,5
	Загруженность CPU ($RND_{cpuload}^{sc}$), %									
Старт (st_i)	0,1	0,1	0,1	0,1	0,1	0,1	0,2	0,2	0,2	0,3
Перезагрузка (res_i)	0,2	0,3	0,2	0,1	0,2	0,2	0,3	0,3	0,3	0,4
Анализ трафика (an_i)	0,4	0,4	0,5	0,5	0,5	0,6	0,7	0,7	0,8	0,9
	Загруженность SSD ($RND_{ssdload}^{sc}$), %									
Старт (st_i)	0,1									
Перезагрузка (res_i)	0,1	0,2	0,2	0,3	0,3	0,4	0,5	0,6	0,7	0,8
Анализ трафика (an_i)	0,3	0,4	0,6	0,7	0,8	1,0	1,1	1,2	1,4	1,5
	Загруженность ОЗУ ($RND_{ramload}^{sc}$), %									
Старт (st_i)	0,1									
Перезагрузка (res_i)	0,2	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
Анализ трафика (an_i)	0,5	0,6	0,7	0,8	0,9	1,0	1,1	1,2	1,3	1,4

Таблица 4. Потребление ресурсов в режиме защиты от DDoS-атак

Ситуация	**День эксперимента (p_i)**									
	1	2	3	4	5	6	7	8	9	10
	Емкость DDoS-атаки ($N_c^h(t)$), гбит/сек									
	2,3	3,5	4,0	4,4	4,8	5,8	6,4	7,9	8,3	9,9
	$N_{пр}$, млн. шт./сек									
	0,5	1,0	1,5	2,0	2,5	3,0	4,5	7,0	9,5	9,9
	Загруженность CPU ($RD_{cpuload}^{sc}$), %									
Старт (st_i)	3,1	4,2	4,6	4,9	5,6	6,0	7,5	8,0	9,0	9,3
Перезагрузка (res_i)	4,1	4,8	5,2	6,3	7,0	7,5	8,7	9,0	9,5	9,7
Анализ трафика (an_i)	5,2	5,6	6,0	6,5	7,3	8,2	8,6	8,9	9,3	9,9
	Загруженность SSD ($RD_{ssdload}^{sc}$), %									
Старт (st_i)	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	2,3
Перезагрузка	2,0	3,2	4,1	4,3	4,8	5,0	5,5	6,0	6,7	7,5

(res_i)										
Анализ трафика (an_i)	2,4	3,0	3,5	4,0	4,7	5,6	7,0	8,2	8,6	9,3
	Загруженность ОЗУ ($RD^{sc}_{ramload}$), %									
Старт (st_i)	0,8	0,9	1,3	1,5	1,9	2,3	2,8	3,0	3,2	3,7
Перезагрузка (res_i)	1,3	1,6	1,9	2,5	2,9	3,8	4,0	5,1	5,7	5,9
Анализ трафика (an_i)	1,7	1,9	2,3	2,7	3,6	4,0	4,5	5,2	5,7	6,4

Таким образом, по данным таблиц 5–6 была рассчитана на основании формул (16-21) средняя нагрузка по загруженности CPU, SSD и ОЗУ, что представлено в таблице 5.

Таблица 5. Средняя нагрузка на ресурсы вычислительного кластера

Ресурс	**Режим простоя**	**Режим защиты от DDoS-атак**	**Разница, раз**
Центральный процессор, %	0,3	6,9	23,0
Твердотельный накопитель, %	0,4	4,0	10,0
Оперативная память, %	0,5	3,1	6,2

$$RNDS^{sc}_{cpuload} = \frac{\sum_{i=1}^{p} RND^{sc}_{cpuload}(i)}{\sum_{i=1}^{p}(st_i + res_i + an_i)}, \quad (16)$$

$$RNDS^{sc}_{ssdload} = \frac{\sum_{i=1}^{p} RND^{sc}_{ssdload}(i)}{\sum_{i=1}^{p}(st_i + res_i + an_i)}, \quad (17)$$

$$RNDS^{sc}_{ramload} = \frac{\sum_{i=1}^{p} RND^{sc}_{ramload}(i)}{\sum_{i=1}^{p}(st_i + res_i + an_i)}, \quad (18)$$

$$RDS^{sc}_{cpuload} = \frac{\sum_{i=1}^{p} RD^{sc}_{cpuload}(i)}{\sum_{i=1}^{p}(st_i + res_i + an_i)}, \quad (19)$$

$$RDS^{sc}_{ssdload} = \frac{\sum_{i=1}^{p} RD^{sc}_{ssdload}(i)}{\sum_{i=1}^{p}(st_i + res_i + an_i)}, \quad (20)$$

$$RDS^{sc}_{ramload} = \frac{\sum_{i=1}^{p} RD^{sc}_{ramload}(i)}{\sum_{i=1}^{p}(st_i + res_i + an_i)}, \quad (21)$$

в которых $RNDS^{sc}_{cpuload}$ – средняя загруженность CPU в режиме простоя, $RNDS^{sc}_{ssdload}$ – средняя загруженность SSD в режиме простоя, $RNDS^{sc}_{ramload}$ – средняя загруженность RAM в режиме простоя, $RDS^{sc}_{cpuload}$ – средняя загруженность CPU в режиме DDoS, $RDS^{sc}_{ssdload}$ – средняя загруженность SSD в режиме DDoS, $RDS^{sc}_{ramload}$ – средняя загруженность RAM в режиме DDoS, p – количество дней эксперимента.

Разница в таблице 5 рассчитывалась следующим образом (22-24):

$$Dif_{cpu} = \frac{RNDS^{cp}_{cpuload}}{RDS^{cp}_{cpuload}}, \quad (22)$$

$$Dif_{ssd} = \frac{RNDS^{cp}_{ssdload}}{RDS^{cp}_{ssdload}}, \quad (23)$$

$$Dif_{ram} = \frac{RNDS^{cp}_{ramload}}{RDS^{cp}_{ramload}}, \quad (24)$$

в которых Dif_{cpu} – разница в нагрузке на CPU, Dif_{ssd} – разница в нагрузке на SSD, Dif_{ram} – разница в нагрузке на RAM.

Таким образом, нагрузочные значения (таблица 5) на ресурсы всех физических серверов кластера является достаточно низкой (в том числе и в режиме DDoS-атак) из-за возможности равномерного распределения сетевой нагрузке по всему кластеру. При этом кластер во время DDoS-атак был полностью доступен по внешней глобальной и локальной сетям и работал без перебоев благодаря фильтрации DDoS-атак с помощью импульсной нейронной сети. Сравнение методов обучения импульсной нейронной сети представлено в таблице 6.

Таблица 6. Сравнение методов обучения ИмНС без учителя

Параметр	**STDP**	**GUL**	**BCM-правило**	**ABS-правило**	**Предложенный метод обучения**
Скорость обучения «с нуля» при DDoS-атаках, мин	70,00	60,00	150,00	80,00	20,00

Таким образом, предложенный метод обучения показал достаточно высокую скорость (20 минут на полное обучение) при самообучении для защиты от DDoS-атак вычислительного кластера.

Для апробации использовалось следующее оборудование, расположенное в одном из центров обработки данных г. Москвы: 30 физических серверов с процессорами Intel Xeon 5690 и внешним сетевым каналом 20 Гбит/с., 960 ГБ оперативной памяти, твердотельные накопители Samsung 970 Evo Plus.

VII. Заключение

В ходе проведенных исследований были получены следующие результаты:

1. Предложена импульсная нейронная сеть для фильтрации DDoS-атак в автоматическом режиме с возможностью поддержки принятия решений

системным администратором. Данная нейронная сеть является самым новым решением в области нейронных сетей (третье поколение) и ранее не применялась для автоматической фильтрации DDoS-атак (в том числе и генерации правил фильтрации). При этом была разработана и обоснована зависимость количество нейронов и слоев импульсной нейронной сети от физических ресурсов сервера и кластера.

2. В рамках реализованной импульсной нейронной сети предложен модифицированный метод обучения для фильтрации DDoS-атак, основанный на методе обратного распространения ошибки (градиентный спуск). Результаты обучения данным методом с точки зрения быстроты и точности оказались следующими: 20 минут длится обучение с учетом новых данных, а при DDoS-атаках процент ложных срабатываний на легитимные пользовательские запросы предложенных нейронной сетью правил фильтрации составляет 0,1. Таким образом, предложенный метод обучения в рамках фильтрации DDoS-атак зарекомендовал себя положительно, что подтверждают полученные результаты исследования.

3. Проведено тестирование, доказывающее целесообразность использования предлагаемой импульсной нейронной сети не только с точки зрения науки, но и практики: длительное нагрузочное тестирование показало, что низкая загруженность физических ресурсов в режимах простоя/защиты от DDoS-атак (CPU – 0.3%/6.9%, SSD – 0.4%/4.0%, RAM – 0.5%/3.1%) позволяет эффективно функционировать всей рабочей среде (в нашем случае – кластер и находящиеся в нем физические серверы с различными сервисами), и как следствие – не нести какой-либо ущерб от DDoS-атак.

Предложенный метод позволяет реализовать упреждающее реагирование на внешнее воздействие, выходящее за рамки противодействия DDoS атакам, что открывает возможность применения в различных системах управления.

В будущих исследованиях планируется реализации специализированного эффективного механизма для фильтрации DDoS-атак не отдельно взятого кластера, а целых предприятий, что позволит более тщательно апробировать разработанное решение и модифицировать его с научной и практической точек зрения. Считаем, что данное исследование будет полезно не только научному сообществу, но и специалистам-практикам в области IT-индустрии.

Благодарности

Исследования выполнены при поддержке Министерства науки и высшего образования РФ в рамках выполнения Государственного задания № FEUE-2020-0007.

Библиография

[1] Ramanauskaitė, S. et al., **2017**. Modeling of two-tier DDoS by combining different type of DDoS models. *Conference of Electrical, Electronic and Information Sciences (eStream)*, 1–4.

[2] Xiang, Y.; Li, Z., **2006**. An Analytical Model for DDoS Attacks and Defense. *Conference on Computing in the Global Information Technology*, 66.

[3] Luo, J. et al., **2014**. On a Mathematical Model for Low-Rate Shrew DDoS. *IEEE Transactions on Information Forensics and Security 9 (7)*, 2321034.

[4] Eian, M.; Mjølsnes, S.F., **2011**. The modeling and comparison of wireless network Denial of Service attacks. *ACM Symposium on Operating Systems Principles (SOSP) workshop*, 7.

[5] Fouladi, R.F.; Kayatas, C.E.; Anarim, E., **2016**. Frequency based DDoS attack detection approach using naive Bayes classification. *International Conference on Telecommunications and Signal Processing (TSP)*, 104-107.

[6] Yuan, X.; Li, C.; Li, X., **2017**. Deep ++Defense: identifying DDoS attack via deep learning. *IEEE International Conference on Smart Computing (SMARTCOMP)*, 1-8.

[7] Singh, K. J.; De, T., **2015**. An approach of ddos attack detection using classifiers. *Emerging Research in Computing, Information, Communication and Applications*, 429-437.

[8] Abdullah, E.C.; Ali B., **2021**. Detection of DDoS attacks with feed forward based deep neural network model. *Expert Systems with Applications*. *169*, 114520.

[9] Viet-Hoang, T.; Olivier, B., **2020**. Beyond socket options: Towards fully extensible Linux transport stacks. *Computer Communications*. *162*, 118-138.

[10] Syed G.A. et.al., **2021**. Generic signature development for IoT Botnet families. *Forensic Science International: Digital Investigation*. 38, 301224.

[11] Ying L. et.al., **2022**. Software-defined DDoS detection with information entropy analysis and optimized deep learning. *Future Generation Computer Systems*. *129*, 99-114.

[12] Amaizu, G.C. et.al., **2021**. Composite and efficient DDoS attack detection framework for B5G networks. *Computer Networks*. *188*, 107871.

[13] Matheus, P. N. et.al., **2021**. Adversarial Deep Learning approach detection and defense against DDoS attacks in SDN environments. *Future Generation Computer Systems*. *125*, 156-167.

[14] Nisha, A. et.al., **2021**. Automated DDOS attack detection in software defined networking. *Journal of Network and Computer Applications*. *187*, 103108.

[15] Silva, L.E.; Coury D.V. **2020**. Network traffic prediction for detecting DDoS attacks in IEC 61850 communication networks. *Computers & Electrical Engineering*. *87*, 106793.

[16] Zhang, L.; Wang, J. et.al., **2022**. A Hybrid Method of Entropy and SSAE-SVM Based DDoS Detection and Mitigation Mechanism in SDN. *Computers & Security*. 102604.

[17] Deepak, K.S.; Tarun, D. et.al., **2021**. Anomaly detection framework to prevent DDoS attack in fog empowered IoT networks. *Ad Hoc Networks*. *121*, 102603.

[18] Huu-Khoi, B. et.al., **2021**. CREME: A toolchain of automatic dataset collection for machine learning in intrusion detection. *Journal of Network and Computer Applications*. *193*, 103212.

[19] Arpita, P.; Gaurav S., **2021**. Serving while attacked: DDoS attack effect minimization using page separation and container allocation strategy. *Journal of Information Security and Applications*. *59*, 102818.

[20] Manjula, H.T.; Neha, M., **2021**. An approach to on-stream DDoS blitz detection using machine learning algorithms. *Materials Today: Proceedings*, 2214-7853.

[21] Congyuan, X.; Jizhong, S.; Xin, D., **2021**. Low-rate DoS attack detection method based on hybrid deep neural networks. *Journal of Information Security and Applications*. *60*, 102879.

[22] Abdullah, S.A.; Jochen, S., **2021**. Deep Learning Algorithms for Detecting Denial of Service Attacks in Software-Defined Networks. *Procedia Computer Science*. *191*, 254-263.

[23] Marios, T.; Christoforos, N., **2021**. Detection of collaborative misbehaviour in distributed cyber-attacks. *Computer Communications*. *174*, 28-41.

[24] Dalia, N.; Fatma, A., **2021**. Hussain, Multifractal detrended fluctuation analysis based detection for SYN flooding attack. *Computers & Security*. *107*, 102315.

[25] Lian, Y. et.al., **2021**. PBCNN: Packet Bytes-based Convolutional Neural Network for Network Intrusion Detection. *Computer Networks*. *194*, 108117.

[26] Zhen, Y. et.al., **2022**. A systematic literature review of methods and datasets for anomaly-based network intrusion detection. *Computers & Security*. 116, 102675.

[27] Jun, Z. et.al., **2021**. Automatically predicting cyber attack preference with attributed heterogeneous attention networks and transductive learning. *Computers & Security*. *102*, 102152.
[28] Weicheng, Q. et.al., **2022**. Hybrid intrusion detection system based on Dempster-Shafer evidence theory. *Computers & Security*. *117*, 102709.
[29] Zahid, H. et.al., **2021**. An effective genetic algorithm-based feature selection method for intrusion detection systems. *Computers & Security*. *110*, 102448.
[30] NetScout — https://www.netscout.com/arbor-ddos (date of access to the page: 20.04.22).
[31] MicroSoft — https://docs.microsoft.com/ru-ru/azure/fxt-edge-filer/cluster-create (date of access to the page: 20.04.22)
[32] FitzHugh, R., **1961**. Impulses and physiological states in theoretical models of nerve membrane. *Biophys. J. 1*, 445–466.
[33] Paninski, L., **2004**. Maximum likelihood estimation of cascade point-process neural encoding models. *Network.* 15, 243–262.
[34] Palchevsky, E.V. et.al., **2020**. Intelligent data analysis for forecasting threats in complex distributed systems. *CEUR Workshop Proceedings*. *2744*, 285-296.
[35] Kulikov, G.G. et.al., **2020**. Formal method of structural-logical identification of functional model of subject area by polycubic data matrix. *Acta Polytechnica Hungarica*. *17 (8)*, 41-59.
[36] Kolmogorov, A.N. et.al., **1937**. Investigation of the equation of diffusion associated with an increase in the amount of substance, and its application to one biological problem. *Moscow State University. Series: Mathematics and mechanics*. *1*, 1-26.
[37] Hodgkin, A.L.; Huxley, A.F., **1952**. A quantitative description of membrane current and its application toconduction and excitation in nerve. *J. Physiol. 117 (4)*, 500-544.
[38] FitzHugh, R., **1969**. Mathematical model sofex citation and propagation in nerve. *Biological Engineering*. 1-85.
[39] Palchevsky, E.V.; Khristodulo, O.I., **2019**. *Development of an impulse neural network with the possibility of high-speed learning to neutralize DDoS attacks*. 32 (4), 561-577.

Статья получена 25 апреля 2022.
Е. В. Пальчевский, Финансовый университет при Правительстве Российской Федерации, Москва, Россия (e-mail: teelxp@inbox.ru).
В.В. Антонов, Уфимский государственный авиационный технический университет, Уфа, Россия (e-mail: antonov.v@bashkortostan.ru).
Л. Е. Родионова, Уфимский государственный авиационный технический университет, Уфа, Россия (e-mail: lurik@mail.ru).
Л. А. Кромина, Уфимский государственный авиационный технический университет, Уфа, Россия (e-mail: luyda-kr@yandex.ru)

Development of a self-learning spike neural network for proactive response to external information impacts of various nature

E.V. Palchevsky, V.V. Antonov, L.E. Rodionova, L.A. Kromina

Abstract: Digitalization and intellectualization as part of the mass introduction of cyber-physical systems "Industry 4.0" has become a real trend in which cyber technologies provide automated and automatic control, greater efficiency, and increased security. At the same time, the integration of such technologies into critical infrastructure facilities is often subject to cyber threats, and as a result, cyber-attacks, violating not only the confidentiality and integrity of data, but also accessibility, for example, using DDoS attacks, which indicates the imperfection of most data filtering methods. attacks at various levels of the OSI model. This leads to the fact that many organizations whose physical and computing resources have access to the external global Internet network face the inaccessibility of their own services, which leads to the inability to provide the necessary data and services for both their own employees and customers, which leads to represents the financial loss of the company from equipment downtime. To minimize losses from this problem, it is proposed to use a spike (impulse) neural network to filter attacks by unauthorized external traffic (DDoS).

The main features of the proposed neural network are both high speed and quality (due to constant learning on big data) of self-learning, and quick response to DDoS attacks (including those that are unknown), as well as structural dependence (the number of neurons and layers of the impulse neural network) from the physical (computing) resources of the server/cluster. A modified method of nested mathematical models of self-learning (unsupervised learning) of a pulsed neural network is proposed, which is based on the standard ANN training method with error backpropagation (gradient descent), which allows the pulsed neural network to quickly and efficiently learn in order to filter attacks by external unauthorized traffic.

Keywords: DDoS attacks, DDoS filtering, spike neural network, impulse neural network.

REFERENCES

[1] Ramanauskaitė, S. et al., **2017**. Modeling of two-tier DDoS by combining different type of DDoS models. *Conference of Electrical, Electronic and Information Sciences (eStream)*, 1–4.

[2] Xiang, Y.; Li, Z., **2006**. An Analytical Model for DDoS Attacks and Defense. *Conference on Computing in the Global Information Technology*, 66.

[3] Luo, J. et al., **2014**. On a Mathematical Model for Low-Rate Shrew DDoS. *IEEE Transactions on Information Forensics and Security 9 (7)*, 2321034.

[4] Eian, M.; Mjølsnes, S.F., **2011**. The modeling and comparison of wireless network Denial of Service attacks. *ACM Symposium on Operating Systems Principles (SOSP) workshop*, 7.

[5] Fouladi, R.F.; Kayatas, C.E.; Anarim, E., **2016**. Frequency based DDoS attack detection approach using naive Bayes classification. *International Conference on Telecommunications and Signal Processing (TSP)*, 104-107.

[6] Yuan, X.; Li, C.; Li, X., **2017**. Deep ++Defense: identifying DDoS attack via deep learning. *IEEE International Conference on Smart Computing (SMARTCOMP)*, 1-8.

[7] Singh, K. J.; De, T., **2015**. An approach of ddos attack detection using classifiers. *Emerging Research in Computing, Information, Communication and Applications*, 429-437.

[8] Abdullah, E.C.; Ali B., **2021**. Detection of DDoS attacks with feed forward based deep neural network model. *Expert Systems with Applications*. *169*, 114520.

[9] Viet-Hoang, T.; Olivier, B., **2020**. Beyond socket options: Towards fully extensible Linux transport stacks. *Computer Communications*. *162*, 118-138.

[10] Syed G.A. et.al., **2021**. Generic signature development for IoT Botnet families. *Forensic Science International: Digital Investigation*. 38, 301224.

[11] Ying L. et.al., **2022**. Software-defined DDoS detection with information entropy analysis and optimized deep learning. *Future Generation Computer Systems*. *129*, 99-114.

[12] Amaizu, G.C. et.al., **2021**. Composite and efficient DDoS attack detection framework for B5G networks. *Computer Networks*. *188*, 107871.

[13] Matheus, P. N. et.al., **2021**. Adversarial Deep Learning approach detection and defense against DDoS attacks in SDN environments. *Future Generation Computer Systems*. *125*, 156-167.

[14] Nisha, A. et.al., **2021**. Automated DDOS attack detection in software defined networking. *Journal of Network and Computer Applications*. *187*, 103108.

[15] Silva, L.E.; Coury D.V. **2020**. Network traffic prediction for detecting DDoS attacks in IEC 61850 communication networks. *Computers & Electrical Engineering*. *87*, 106793.

[16] Zhang, L.; Wang, J. et.al., **2022**. A Hybrid Method of Entropy and SSAE-SVM Based DDoS Detection and Mitigation Mechanism in SDN. *Computers & Security*. 102604.

[17] Deepak, K.S.; Tarun, D. et.al., **2021**. Anomaly detection framework to prevent DDoS attack in fog empowered IoT networks. *Ad Hoc Networks*. *121*, 102603.

[18] Huu-Khoi, B. et.al., **2021**. CREME: A toolchain of automatic dataset collection for machine learning in intrusion detection. *Journal of Network and Computer Applications*. *193*, 103212.

[19] Arpita, P.; Gaurav S., **2021**. Serving while attacked: DDoS attack effect minimization using page separation and container allocation strategy. *Journal of Information Security and Applications*. *59*, 102818.

[20] Manjula, H.T.; Neha, M., **2021**. An approach to on-stream DDoS blitz detection using machine learning algorithms. *Materials Today: Proceedings*, 2214-7853.

[21] Congyuan, X.; Jizhong, S.; Xin, D., **2021**. Low-rate DoS attack detection method based on hybrid deep neural networks. *Journal of Information Security and Applications*. *60*, 102879.

[22] Abdullah, S.A.; Jochen, S., **2021**. Deep Learning Algorithms for Detecting Denial of Service Attacks in Software-Defined Networks. *Procedia Computer Science*. *191*, 254-263.

[23] Marios, T.; Christoforos, N., **2021**. Detection of collaborative misbehaviour in distributed cyber-attacks. *Computer Communications*. *174*, 28-41.

[24] Dalia, N.; Fatma, A., **2021**. Hussain, Multifractal detrended fluctuation analysis based detection for SYN flooding attack. *Computers & Security*. *107*, 102315.

[25] Lian, Y. et.al., **2021**. PBCNN: Packet Bytes-based Convolutional Neural Network for Network Intrusion Detection. *Computer Networks*. *194*, 108117.
[26] Zhen, Y. et.al., **2022**. A systematic literature review of methods and datasets for anomaly-based network intrusion detection. *Computers & Security*. 116, 102675.
[27] Jun, Z. et.al., **2021**. Automatically predicting cyber attack preference with attributed heterogeneous attention networks and transductive learning. *Computers & Security*. *102*, 102152.
[28] Weicheng, Q. et.al., **2022**. Hybrid intrusion detection system based on Dempster-Shafer evidence theory. *Computers & Security*. *117*, 102709.
[29] Zahid, H. et.al., **2021**. An effective genetic algorithm-based feature selection method for intrusion detection systems. *Computers & Security*. *110*, 102448.
[30] NetScout — https://www.netscout.com/arbor-ddos (date of access to the page: 20.04.22).
[31] MicroSoft — https://docs.microsoft.com/ru-ru/azure/fxt-edge-filer/cluster-create (date of access to the page: 20.04.22)
[32] FitzHugh, R., **1961**. Impulses and physiological states in theoretical models of nerve membrane. *Biophys. J. 1*, 445–466.
[33] Paninski, L., **2004**. Maximum likelihood estimation of cascade point-process neural encoding models. *Network*. 15, 243–262.
[34] Palchevsky, E.V. et.al., **2020**. Intelligent data analysis for forecasting threats in complex distributed systems. *CEUR Workshop Proceedings*. *2744*, 285-296.
[35] Kulikov, G.G. et.al., **2020**. Formal method of structural-logical identification of functional model of subject area by polycubic data matrix. *Acta Polytechnica Hungarica*. *17 (8)*, 41-59.
[36] Kolmogorov, A.N. et.al., **1937**. Investigation of the equation of diffusion associated with an increase in the amount of substance, and its application to one biological problem. *Moscow State University. Series: Mathematics and mechanics*. *1*, 1-26.
[37] Hodgkin, A.L.; Huxley, A.F., **1952**. A quantitative description of membrane current and its application toconduction and excitation in nerve. *J. Physiol. 117 (4)*, 500-544.
[38] FitzHugh, R., **1969**. Mathematical model sofex citation and propagation in nerve. *Biological Engineering*. 1-85.
[39] Palchevsky, E.V.; Khristodulo, O.I., **2019**. *Development of an impulse neural network with the possibility of high-speed learning to neutralize DDoS attacks*. 32 (4), 561-577.