17. Kumar J., Mills R. T., Hoffman F. M., Hargrove W. W. Parallel using large data sets. *Procedia Computer Science*, 2011, no. 4, pp. 1602–1611.

18. Shaukat K., Masood N., Shafaat A. B., Jabbar K., Shabbir H. et al. Dengue Fever in Perspective of Clustering Algorithms. *Data Mining Genomics Proteomics*, 2015, vol. 6, no. 176. DOI:10.4172/2153-0602.1000176.

19. Ximing Lv. Zhou Lan, Guo Xiaona. Research on P2P Network Loan Risk Evaluation Based on Generalized DEA Model and R-Type Clustering Analysis under the Background of Big Data. *Financial Risk Management*, 2017, vol. 6, no. 2, pp. 163–190.

20. Yin J. A., Meng Y., Jin Y. Developmental approach to structural self-organization in reservoir computing. *IEEE Transactions on Autonomous Mental Development*, 2012, no. 4 (4), pp. 273–289.

DOI 10.21672/2074-1707.2020.49.4.020-032
УДК 004:614

НЕЙРОСЕТЕВАЯ ТЕХНОЛОГИЯ ОБНАРУЖЕНИЯ АНОМАЛЬНОГО СЕТЕВОГО ТРАФИКА

Статья поступила в редакцию 05.12.2019, в окончательном варианте – 26.02.2020.

Частикова Вера Аркадьевна, Кубанский государственный технологический университет, 350072, Российская Федерация, г. Краснодар, ул. Московская, 2,
кандидат технических наук, доцент, e-mail: chastikova_va@mail.ru

Жерлицын Сергей Анатольевич, Кубанский государственный технологический университет, 350072, Российская Федерация, г. Краснодар, ул. Московская, 2,
студент, e-mail: kpytooooo@gmail.com

Воля Яна Игоревна, Кубанский государственный технологический университет, 350072, Российская Федерация, г. Краснодар, ул. Московская, 2,
студент, e-mail: volya_y@mail.ru

Сотников Владимир Владимирович, Кубанский государственный технологический университет, 350072, Российская Федерация, г. Краснодар, ул. Московская, 2,
студент, e-mail: bubert9@mail.ru

Рассмотрены существующие методы анализа сетевого трафика, указаны их возможности и ограничения. Продемонстрирована актуальность решаемой задачи. Обоснована целесообразность использования нейросетевого подхода к обнаружению аномалий сетевого трафика. Исследована эффективность использования алгоритмов роевого интеллекта применительно к задаче обучения нейронных сетей, выявлены особенности данных алгоритмов. Реализована объектно-ориентированная библиотека для выявления сетевых атак с использованием нейросети с архитектурой многослойного персептрона. На данном этапе исследования для обучения нейросети и оценки качества распознавания трафика был применен датасет KDD Cup 1999 Data. Описаны преимущества и недостатки реализованного решения. Представлен способ устранения распространенного недостатка датасетов, связанного с несбалансированностью обучающих данных. Описаны используемые технологии: архитектура нейронной сети, алгоритм обучения, способ уменьшения размерности обрабатываемых данных. На втором этапе был использован набор данных CSE-CIC-IDS2018. Предложена нейросетевая модель, построенная на базе архитектуры LSTM и эмбеддинговой сетей. Для обучения разработанной системы предложено применение алгоритма Adam, основанного на градиентном спуске. На основе использования названных алгоритмов, моделей и технологий был реализован, а затем и протестирован программный комплекс для обнаружения сетевых атак.

Ключевые слова: нейронная сеть, сетевая атака, многослойный перцептрон, роевой интеллект, LSTM-сеть, эмбеддинговая сеть, Focal Loss, алгоритм Adam

NEURAL NETWORK TECHNOLOGY FOR DETECTING ANOMALOUS NETWORK TRAFFIC

The article was received by the editorial board on 05.12.2019, in the final version – 26.02.2020.

Chastikova Vera A., Kuban State Technological University, 2 Moskovskaya St., Krasnodar, 350072, Russian Federation,
Cand. Sci. (Engineering), Associate Professor, e-mail: chastikova_va@mail.ru

Zherlitsyn Sergey A., Kuban State Technological University, 2 Moskovskaya St., Krasnodar, 350072, Russian Federation,
student, e-mail: kpytooooo@gmail.com

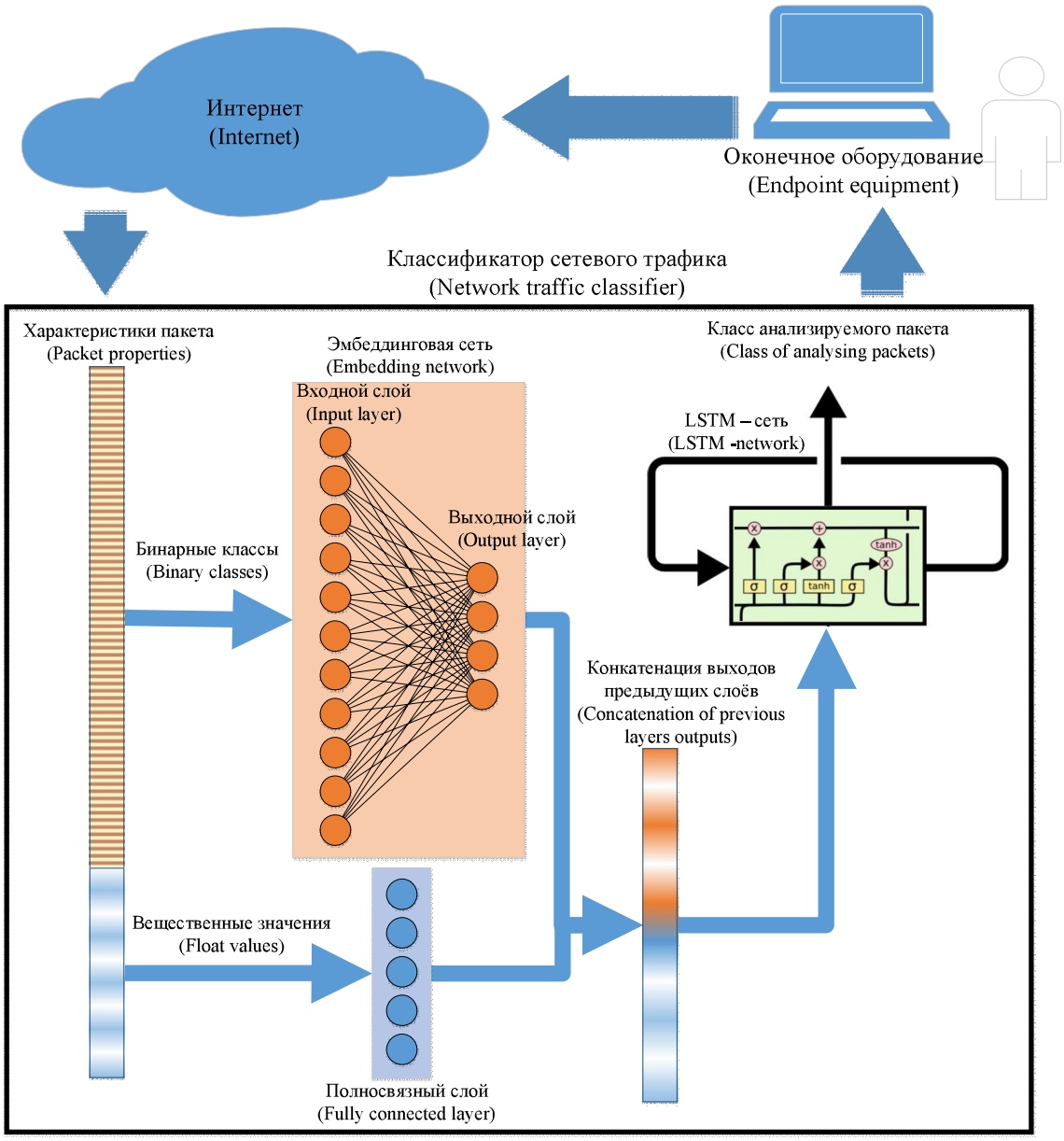
Volya Yana I., Kuban State Technological University, 2 Moskovskaya St., Krasnodar, 350072, Russian Federation,
student, e-mail: volya_y@mail.ru

Sotnikov Vladimir V., Kuban State Technological University, 2 Moskovskaya St., Krasnodar, 350072, Russian Federation,
student, e-mail: bubert9@mail.ru

Existing network traffic analysis methods are considered. The relevance of the problem is demonstrated. The efficiency of swarm intelligence algorithms as applied to the task of training neural networks is analyzed, the features of these algorithms are revealed. An object-oriented library for detecting network attacks using a multi-layer perceptron neural network architecture has been implemented. The advantages and disadvantages of the implemented solution are described. A method is proposed for eliminating the widespread lack of datasets related to the imbalance of training data. The technologies used are described: the architecture of the neural network, the learning algorithm, a method of reducing the dimension of the processed data. A neural network model based on the LSTM architecture and embedding networks is proposed. To train the developed system, the use of the Adam algorithm based on gradient descent is proposed. Based on the use of the above algorithms, models and technologies, a software package for detecting network attacks was implemented and then tested.

Key words: neural network, network attack, perceptron, swarm intelligence, LSTM network, embedding network, Focal Loss, Adam algorithm

Graphical annotation (Графическая аннотация)



Введение. Функционирование современного информационного общества требует обеспечения стабильного доступа физических лиц и организаций к базам данных и различным сетевым ресурсам в любое время. Однако такой доступ может прерываться, в том числе и из-за сетевых атак. Они производятся с целью нарушения работы информационной системы или затруднения доступа к ней; могут привести к значительным убыткам пострадавших лиц и компаний.

Актуальность данной проблемы определяется большим количеством происходящих в мире сетевых атак, что видно из графика на рисунке 1 [9].

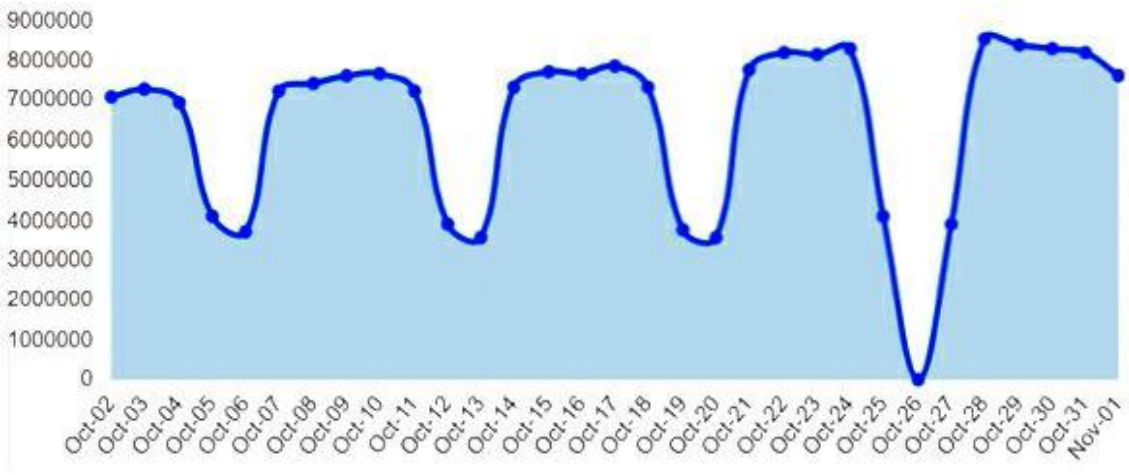


Рисунок 1 – Количество сетевых атак за период с 02.10.2019 до 01.11.2019. Сетевым атакам подвергаются устройства жителей и организаций по всему миру. Согласно статистике «Лаборатории Касперского», количество атак с каждым годом увеличивается, при этом их сложность также растет [9]. Стандартные средства детектирования сетевых атак не всегда способны обнаружить угрозы, что может привести к утрате работоспособности интернет-ресурсов и значительному ущербу для компании

Общая характеристика существующих методов обнаружения сетевых атак. Один из распространенных способов обнаружения сетевых атак – это анализ IP-адресов, с которых приходят запросы на веб-сервер, и анализ данных сетевого трафика на полное совпадение с имеющейся базой признаков сетевых атак. В связи с возрастающей сложностью угроз данный метод имеет низкую эффективность. Одна из возможностей обойти такой алгоритм обнаружения – скрыть или проксировать IP-адрес злоумышленника. При этом в случае реализации угрозы средствами защиты будут заблокированы хосты обыкновенных пользователей, и они не смогут получить доступ к ресурсам.

Также существуют системы поиска сетевых атак, основанные на проверке соответствия сетевого трафика определенным гибким правилам. Данный способ позволяет отфильтровать пакеты данных с целенаправленно искаженным содержимым, переполнение буфера определённых сервисов, но потребуется значительное время, чтобы проанализировать новые варианты сетевых атак и дополнить систему правил.

Помимо вышеперечисленного в настоящее время используются такие методы обнаружения сетевых атак, как статистические методы, экспертные системы и нейронные сети. Они применяются как комплексно, так и в качестве отдельных средств анализа сетевого трафика. Такой подход является более гибким и позволяет упростить процесс актуализации системы безопасности [1].

Данная работа посвящена исследованию и разработке средств защиты на базе нейронных сетей, способных отличить вредоносные подключения от нормальных с целью последующего принятия мер по обеспечению безопасности защищаемой системы.

Подготовительный этап построения модели нейросетевой системы. Выбор архитектуры нейронной сети определяется параметрами анализируемой информации. Для обучения модели была использована база данных сетевых атак, находящаяся в открытом доступе KDD Cup 1999 Data [13], находящаяся в открытом доступе в интернете. Из всех приведенных на каждый пункт параметров сетевого трафика были выбраны 28 [5], на основе которых можно определить наличие вредоносной активности в сети. Все данные были конвертированы из текстового формата в цифровой и нормализованы для корректной работы нейронной сети.

Для названного датасета нормализация заключалась в приведении каждого параметра к диапазону от 0 до 1 в соответствии с формулой:

$$p_{in} = \frac{p_i - p_{min}}{p_{max} - p_{min}}.$$

На основе анализа различных архитектур нейронных сетей для решения поставленной задачи был выбран многослойный перцептрон, так как его архитектура способна симулировать множественные условные взаимосвязи.

Подобная нейронная сеть принимает на вход вектор входных значений – параметры сетевого трафика. Каждый нейрон связан со всеми элементами последующего слоя, нейроны, в нашем случае, имеют сигмоидальную функцию активации [5].

Способ обучения нейросетевой системы. Для обучения сети были применены: генетический алгоритм, алгоритм серых волков, алгоритм светлячков [2–5].

Алгоритм светлячков [2] и алгоритм серых волков [3] показали высокую эффективность при решении задачи глобальной оптимизации многоэкстремальных функций. Это и стало критерием выбора их в качестве алгоритмов обучения нейронных сетей.

Каждый из рассматриваемых методов опирается на вычисление фитнес-функции и направлен на поиск её глобального минимума. В данной задаче эту роль выполняет среднеквадратическое отклонение результатов, полученных на выходе нейронной сети, от ожидаемых значений, хранящихся в базе. Общая схема полного цикла обучения нейронной сети представлена на рисунке 2.

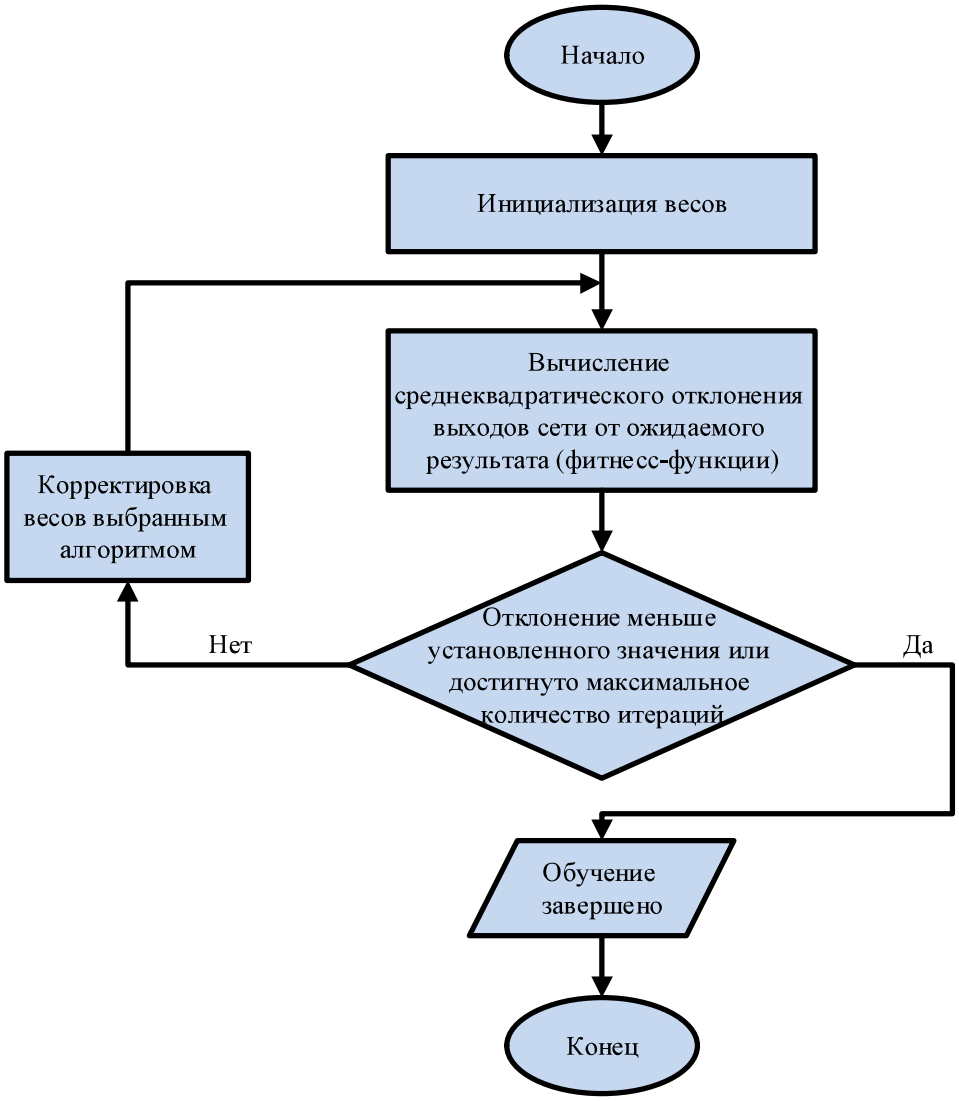


Рисунок 2 – Схема полного цикла обучения сети

Для предварительного тестирования эффективности алгоритмов при решении задачи обучения нейронной сети были разработаны и реализованы на языке программирования C# объектно-ориентированная библиотека и программный комплекс для создания, настройки, сохранения и обучения нейронных сетей различных видов, на основе которой проводились исследования [2–5]. Согласно их результатам, алгоритм светлячков показал высокое быстродействие и относительно низкую степень отклонения от точки глобального экстремума.

В рамках данного этапа исследования алгоритм светлячков использовался в качестве основного алгоритма обучения. Он продемонстрировал устойчивость при усложнении структуры нейронной сети: как при увеличении количества нейронов на скрытых слоях, так и при увеличении глубины сети. Результаты применения алгоритма при разных конфигурациях сети представлены на рисунке 3.

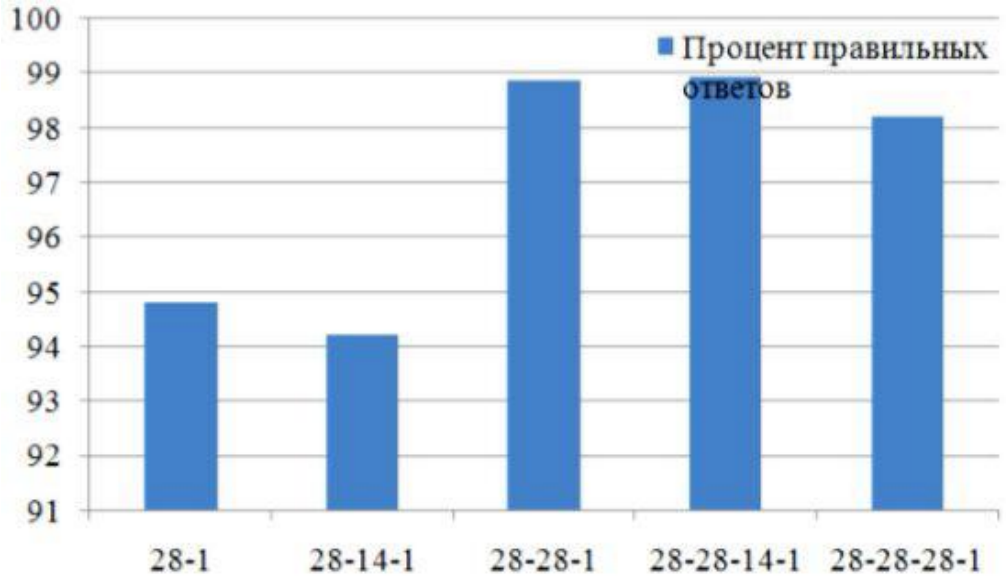


Рисунок 3 – Эффективность обучения различных конфигураций нейронной сети алгоритмом светлячков

В данной диаграмме подписи по оси ординат являются условными обозначениями архитектуры нейросети. Каждое из чисел каждой подписи столбца обозначает слой нейросети, а значение числа – количество нейронов на данном слое. Так, например, запись 28-14-1 указывает, что нейросеть обладает входным слоем с 28 нейронами, одним скрытым слоем с 14 нейронами и выходным слоем, состоящим из одного нейрона.

Результат промежуточного этапа позволил выявить недостатки разработанной модели. Однако более подробный анализ провести было невозможно в силу малого объема примеров отдельных видов атак, отсутствия средств мониторинга и неполноты (отдельные типы атак были представлены в критически малом объёме, менее 50 примеров) обучающего датасета.

Новая обучающая выборка. Продолжение исследования стало возможным благодаря появлению новой, более полной, актуальной обучающей выборки. Поэтому на втором этапе данной работы используется набор данных Канадского университета кибербезопасности CSE-CIC-IDS2018 – это один из крупнейших открытых датасетов на текущий момент [8]. Он был сгенерирован в специально развернутой сети, состоящей из 50 хостов злоумышленников, 420 клиентских хостов и 30 серверов, и включает в себя 7 различных векторов атак: брутфорс, эксплуатацию уязвимости Heartbleed, ботнет-атаку, атаку на отказ в обслуживании, распределенную атаку на отказ в обслуживании, атаки на веб-приложения, а также атаку на сеть изнутри с помощью бэкдора в одном из клиентских хостов [1].

Проблема несбалансированности обучающей выборки и предлагаемый подход к ее решению. Проблемой, характерной для всех датасетов с аномальным сетевым трафиком, является крайняя степень несбалансированности примеров для различных классов данных: доброкачественный трафик может составлять до 80 % от общего объема обучающей выборки, в то время как отдельные векторы атак могут быть представлены в менее чем одном проценте сетевых пакетов. Легко классифицируемые доброкачественные пакеты вносят наибольший вклад в градиент функции потерь, на которой основаны применяемые алгоритмы обучения, негативным образом влияя на обучение.

Для борьбы с этой проблемой было принято решение использовать функцию потерь Focal Loss (FL) [20], изначально предложенную группой Facebook AI Research для использования в задачах распознавания образов, но уже успешно апробированную в других областях классификации в условиях несбалансированности классов. Она применяет веса не к классам сетевых атак в целом, а к конкретным пакетам, классификация которых вызвала наибольшие затруднения у модели [11].

FL основана на подсчете перекрестной энтропии между предсказанным и истинным распределением вероятности появления класса в выборке. При этом стандартная формула

$$CE(P_t) = -log\ log\ (P_t)$$

дополняется множителем:

$$FL(P_t) = -(1-P_t)^{\gamma} * log\ log\ (P_t),$$

где γ – гиперпараметр функции.

Это позволяет сбалансировать влияние на результат функции потерь каждого из классов данных, представленных в обучающей выборке.

Проблема разреженности данных. Некоторые из используемых при обучении характеристик пакетов из датасета [8] хранились не в числовой форме, а в виде названий и терминов, в общем виде – строк. Для предотвращения потери данных, оптимальным решением является применение унитарного кодирования. Каждый такой строковый параметр заменяется набором колонок, каждая из которых соответствует одному из возможных текстовых значений параметра. При этом в колонку, соответствующую содержащемуся в конкретном примере значению раскладываемого параметра, вносится число 1, а остальные колонки заполняются нулями.

Ввиду того, что достаточно большое количество информационных полей каждого сетевого пакета закодировано унитарным кодом, обучающая выборка представляет собой разреженную матрицу. В целях снижения размерности обрабатываемых данных часть столбцов данной матрицы отображается в компактные вектора-эмбеддинги (от англ. embedding – вложение) [19] средствами самой нейронной сети, которые затем конкатенируются с остальными информационными признаками.

Эмбеддинговый фрагмент нейросети представлен в виде 2-х полносвязных слоёв, первый из которых обладает значительно большей размерностью, чем второй.

Данная конструкция является первым функциональным блоком модели. Он располагается параллельно с одним полносвязным слоем, на который подаётся небинарная часть характеристик пакета сетевого трафика. Графическая визуализация эмбеддинговой нейросети приведена на рисунке 4.

Приведенная схема демонстрирует главное свойство нейросети названного типа – количество входных параметров значительно больше количества выходных, что способствует повышению их компактности и понижению размерности в целях экономии вычислительных ресурсов.

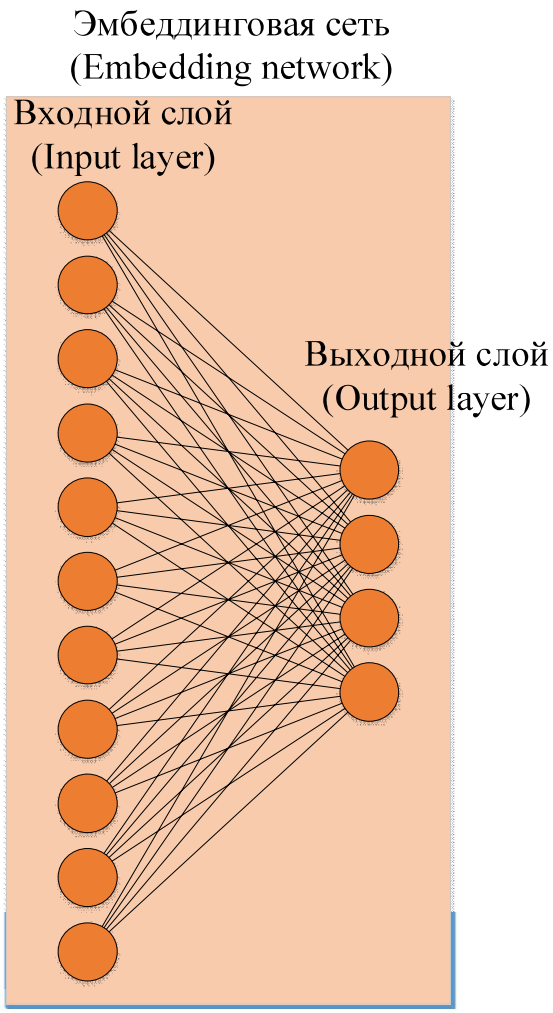


Рисунок 4 – Схема эмбеддинговой сети

Рекуррентные нейронные сети с долгой кратковременной памятью (LSTM). LSTM-сети (англ. long short-term memory – долгая кратковременная память) [12] являются одной из наиболее эффективных существующих нейросетевых моделей для обработки упорядоченных наборов данных (наряду с Gated Recurred Units сетями) [10, 12]. В отличие от классических рекуррентных сетей, они позволяют аккумулировать в себе информацию за продолжительное время, позволяя избежать проблему взрывающихся и затухающих градиентов [10], путем хранения своего состояния $s_i^{(t)}$ в нейронах одних скрытых слоев и управления этим состоянием с помощью нейронов других слоев, скрытых внутри LSTM-ячеек. Эти нейроны управляются рекуррентными связями внутри каждой ячейки. Схематичное изображение LSTM-сети приведено на рисунке 5 [12].

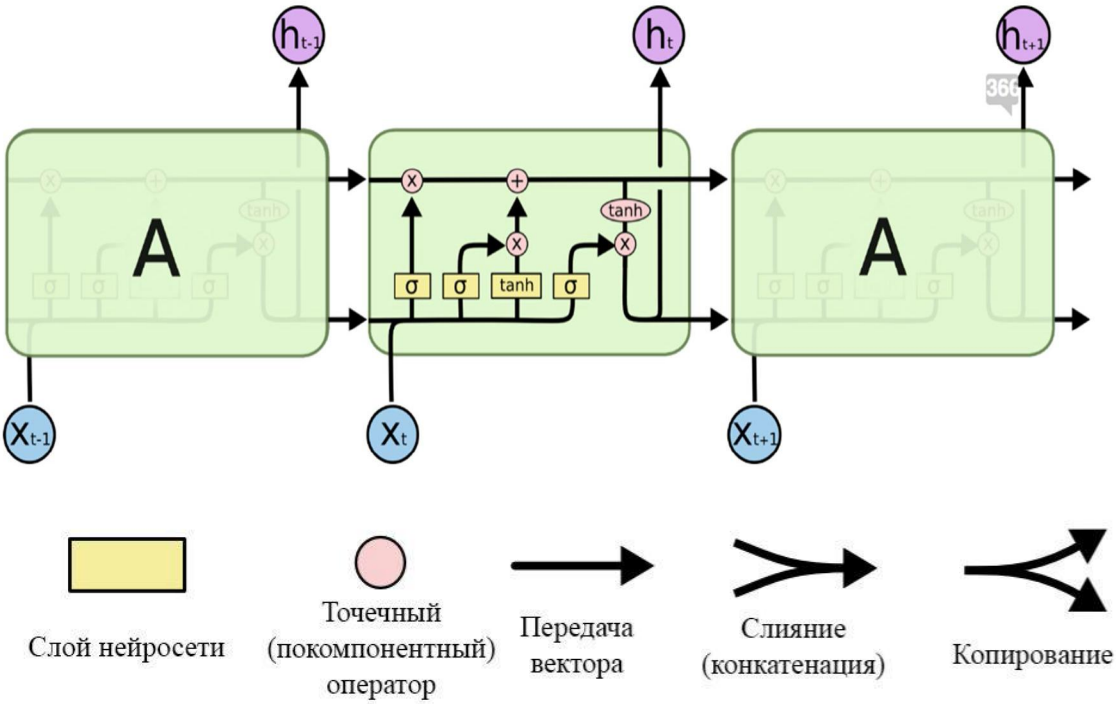


Рисунок 5 – Пример однослойной LSTM-сети: x_t – t-й вектор входных данных; h_t – t-й вектор выходных данных; σ – сигма-функция активации; tanh – функция активации – гиперболический тангенс; + и × – операторы покомпонентного сложения и умножения векторов соответственно

LSTM-сеть состоит из рекуррентно соединенных ячеек, заменяющих собой привычный для классических рекуррентных сетей скрытый слой. Входные данные обрабатываются обычными искусственными нейронами. Значенит их активаций может быть аккумулировано нейронами состояния (нейронами, конвертирующими входные данные и данные с предыдущего состояния в вектор текущего состояния), если это позволяет значение входного вентиля. Нейроны состояния соединены линейной (то есть не содержащей функции активации) рекуррентной связью друг с другом, которая управляется вентилем забывания (рис. 5). Выходное значение каждой LSTM-ячейки управляется соответствующим вентилем забывания. Каждый вентиль снабжен сигмоидальной функцией активации, в то время как входной слой может быть реализован с помощью любой нелинейной функции.

Подобная архитектура позволяет LSTM-сети аккумулировать информацию на протяжении долгого времени. Однако, после того как информация была использована, ячейка может «забыть» свое старое состояние с помощью соответствующего вентиля. К примеру, если обрабатываемый трафик состоит из нескольких сессий и для качественного анализа нейросети необходимо помнить контекст каждой из сессий в отдельности, то LSTM-сеть может обучиться автоматически забывать старый контекст при запуске обработки новой сетевой сессии.

Алгоритм обучения нейронной сети. Для оптимизации нейронной сети был выбран алгоритм Adam (“adaptive moments”), являющийся градиентным алгоритмом оптимизации первого порядка с адаптивной скоростью обучения [16]. Он совмещает преимущества таких методов оптимизации, как RMSProp и momentum [16], обладая при этом большей вычислительной устойчивостью, особенно на ранних этапах обучения. Это достигается коррекцией смещения оценки моментов первого и второго порядков [11]. Алгоритм работает следующим образом:

1. Задать входные параметры:
1.1. Размер шага ϵ (оптимальное значение: 0,001).
1.2. Скорости убывания оценок моментов первого и второго порядка $\rho_1, \rho_1 \in [0,1)$ (оптимальные значения: 0,9 и 0,999 соответственно).
1.3. Константу δ, обеспечивающую вычислительную стабильность (оптимальное значение: 10^{-8}).
1.4. Начальные параметры θ (инициализируются случайным образом).
2. Инициализировать моменты первого и второго порядков $s = 0, r = 0$ и шаг $t = 0$.
3. Пока не будет получена заданная точность, итеративно повторять выполнение следующих шагов:

3.1. Извлечь минибатч из m элементов из обучающей выборки $\{x^{(1)}, \cdots, x^{(m)}\}$, а также соответствующие им метки $y^{(i)}$.

3.2. Вычислить градиент $g \leftarrow \frac{1}{m} \nabla_\theta \sum_i L(f(x^{(i)}; \theta), y^{(i)})$.

3.3. Увеличить шаг $t \leftarrow t + 1$.

3.4. Вычислить несмещенную оценку момента первого порядка $s \leftarrow \rho_1 s + (1 - \rho_1) g$.

3.5. Вычислить несмещенную оценку момента второго порядка $r \leftarrow \rho_2 r + (1 - \rho_2) g \odot g$.

3.6. Вычислить смещение момента первого порядка: $\hat{s} \leftarrow \frac{s}{1-\rho_1^t}$.

3.7. Вычислить смещение момента второго порядка: $\hat{r} \leftarrow \frac{r}{1-\rho_2^t}$.

3.8. Вычислить изменение параметров: $\Delta\theta = -\epsilon \frac{\hat{s}}{\sqrt{\hat{r}} + \delta}$.

3.9. Обновить параметры: $\theta \leftarrow \theta + \Delta\theta$.

Сравнительный анализ указанных алгоритмов оптимизации приведен на рисунке 6.

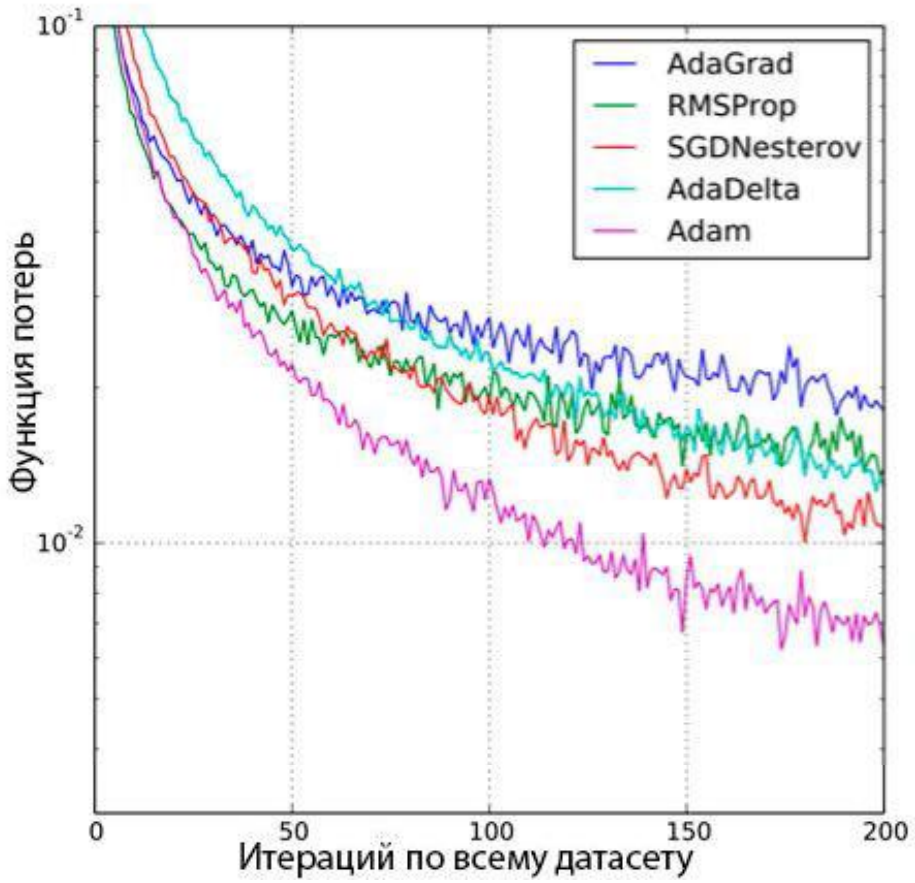


Рисунок 6 – Сравнение алгоритмов оптимизации

Из приведённого графика видно, что выбранный в качестве основного в данном исследовании алгоритм Adam демонстрирует наибольшую производительность, обеспечивая значительный отрыв в прогрессе уменьшения функции потерь уже с 30-й итерации.

Итоговая нейросетевая архитектура. С учетом особенностей, используемых для обучения данных и функций описанных элементов, была разработана нейросетевая архитектура, организованная следующим образом (табл.).

Таблица – Классы сетевого взаимодействия

Название слоя (тип слоя)	Длина выходного вектора	Число параметров	Соединен со слоем
Traffic (входной слой)	196	0	-
Lamda_1 (лямбда-выражение)	157	0	Traffic
Dense_1 (полносвязный слой)	128	20224	Lamda_1
Lamda (лямбда-выражение)	39	0	Traffic
Batch_normalizationv21 (слой батч-нормализации)	128	512	Dense_1
Dense (полносвязный слой)	16	640	Lamda
Dense_2 (полносвязный слой)	64	8256	Batch_normalizationv21
Batch_normalizationv2 (слой батч-нормализации)	16	64	Dense
Batch_normalizationv22 (слой батч-нормализации)	64	256	Dense_2
Concatenate (конкатенирующий слой)	80	0	Batch_normalizationv2, Batch_normalizationv22
Bidirectional (двунаправленная LSTM-сеть)	64	28928	Concatenate
Dense_3 (полносвязный слой)	16	1040	Bidirectional
Batch_normalizationv23 (слой батч-нормализации)	16	64	Dense_3
Dense_4 (полносвязный слой)	10	170	Batch_normalizationv23

Графическая визуализация данного набора слоёв представлена на рисунке 7.

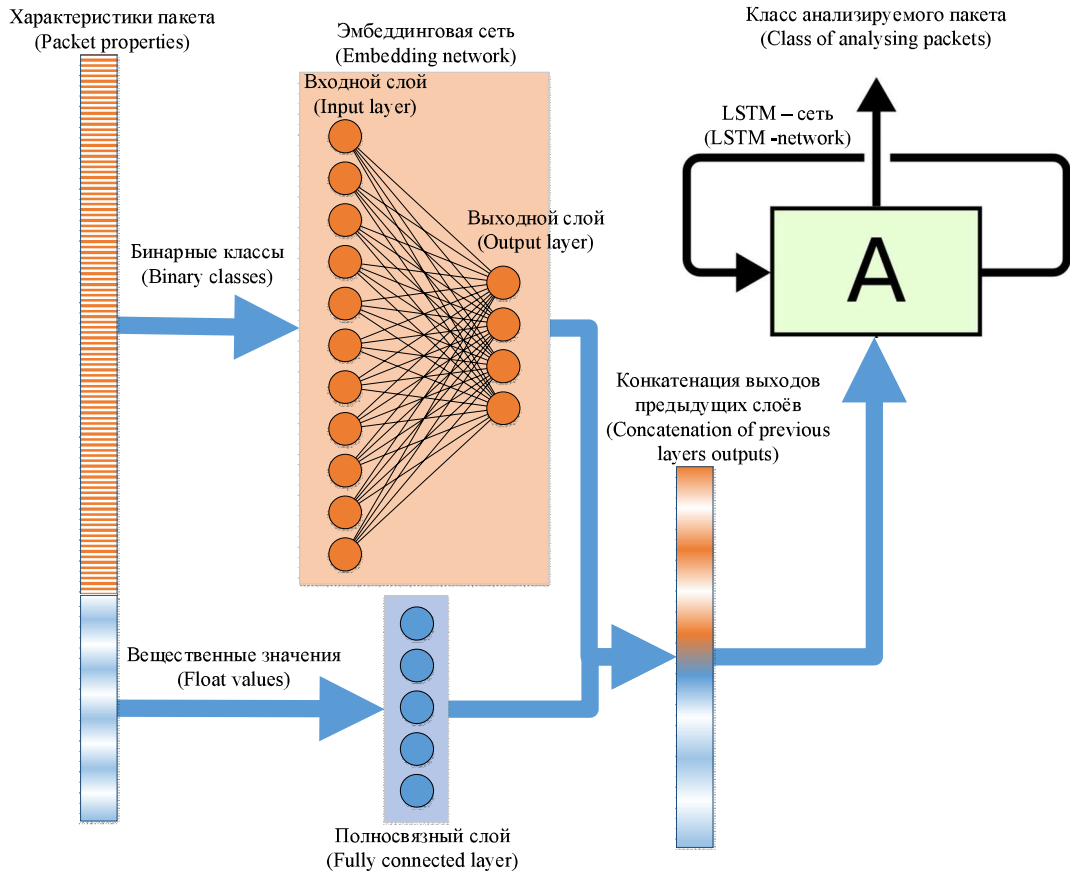


Рисунок 7 – Нейросетевая архитектура системы

Разработанная нейронная сеть содержит 14 слоев и 60154 параметров. Для программной реализации был использован язык программирования Python. В качестве основной библиотеки машинного обучения использовался Keras [14] в сочетании с бэкендом TensoFlow [17].

Практическое обучение модели. Для обеспечения эффективного мониторинга и исследования процесса обучения нейронной сети был разработан ряд функций по сбору и генерации логов, выполняющихся в фоновом режиме. Для визуализации статистических данных применялся модуль TensorBoard [18].

График уменьшения функции потерь изображен на рисунке 8.

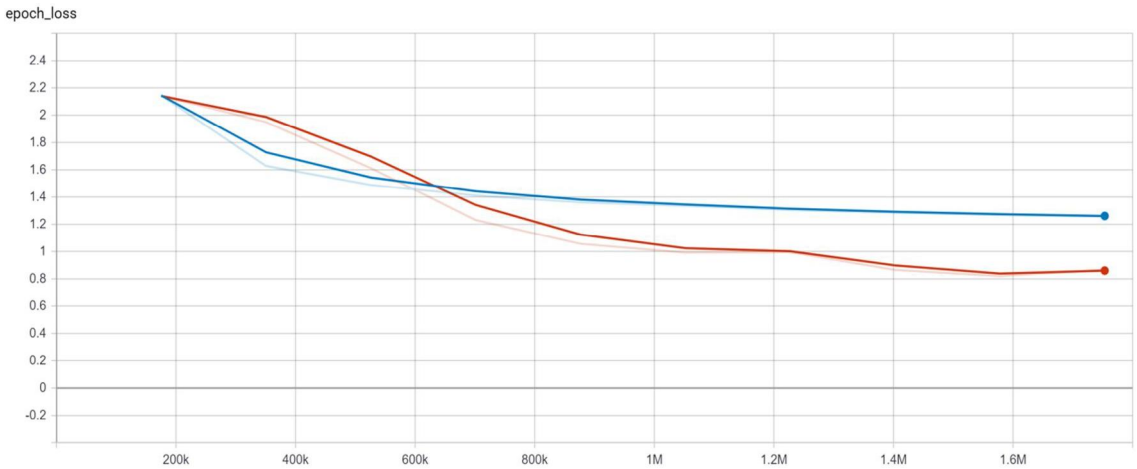


Рисунок 8 – Уменьшение функции потерь в процессе обучения нейронной сети

Полученные данные о функции потерь обладают одной нехарактерной для практики особенностью – начиная с 700000-й итерации значение функции потерь на валидационном датасете представляет меньшее значение, чем на обучающем датасете. Это обусловлено применением функции потерь FL.

Выводы. Разработана и исследована нейросетевая модель анализа трафика на основе персептрона, алгоритмов роевого интеллекта и обучающего набора данных KDD Cup 1999 Data. Полученные результаты показали невозможность построения достаточно точной модели на основе названного датасета в силу его недостаточной репрезентативности.

Был проведен ряд тестов с различными гиперпараметрами нейронной сети глубокой архитектуры. Выявлены оптимальные значения параметров, при которых достигается максимальное значение точности определения аномального сетевого трафика и снижается количество ложных срабатываний. Проведено обучение и анализ точности работы описанной глубокой нейронной сети на новом датасете CSE-CIC-IDS2018. При этом были получены удовлетворяющие современным требованиям значения.

Библиографический список

1. Власенко А. В. Разработка алгоритмов и программ выбора оптимального набора компонент нейтрализации актуальных угроз на основе описания модели и интеграции их в WEB-приложение / А. В. Власенко, П. И. Дзьобан // Вестник Адыгейского государственного университета. Серия 4: Естественно-математические и технические науки. – 2014. – № 3 (142). – С. 189–193.

2. Частикова В. А. Анализ эффективности работы алгоритма светлячков в задачах глобальной оптимизации / В. А. Частикова, Я. И. Воля // Научные труды КубГТУ. – 2016. – № 15. – С. 105–111.

3. Частикова В. А. Исследование алгоритма серых волков / В. А. Частикова, С. А. Жерлицын // Научные труды КубГТУ. – 2016. – № 16. – С. 136–142.

4. Частикова В. А. Исследование эффективности алгоритма поиска косяком рыб в задаче глобальной оптимизации / В. А. Частикова, М. А. Дружинина, А. С. Кекало // Современные проблемы науки и образования. – 2014. – № 4.

5. Частикова В. А. Сравнительный анализ некоторых алгоритмов роевого интеллекта при обнаружении сетевых атак нейросетевыми методами / В. А. Частикова, М. П. Малыхина, С. А. Жерлицын, Я. И. Воля // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. – 2017. – № 129. – С. 106–115.

6. Частикова В. А. Нейросетевой подход к решению задачи построения фоторобота по словесному описанию / В. А. Частикова, С. А. Жерлицын, Я. И. Воля // Известия Волгоградского государственного технического университета. – 2018. – № 8 (218). – С. 63–67.

7. Chollet F. Deep Learning with Python / F. Chollet. – Manning Publications Co., 2017. – 384 с.

8. CSE-CIC-IDS2018 on AWS. – Режим доступа: https://www.unb.ca/cic/datasets/ids-2018.html, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 10.09.2019).

9. Geron A. Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems / A. Geron. – O'Reilly Media, 2017. – 574 p.

10. Gers F. A. Learning to Forget: Continual Prediction with LSTM / F. A. Gers, J. Schmidhuber, F. Cummins // Neural Computation. – 2000. – № 12 (10). – P. 2451–2471.

11. Goodfellow I. Deep Learning / I. Goodfellow, Y. Bengio, A. Courville. – MIT Press, 2016. – 775 с.

12. Hochreiter S. Long short-term memory / S. Hochreiter, J. Schmidhuber // Neural computation. – 1997. – № 9 (8). – P. 1735–1780.

13. KDD Cup 1999 Data. – Режим доступа: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 25.08.2019).

14. Keras: Deep Learning for humans. – Режим доступа: https://github.com/keras-team/keras, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 21.06.2019).

15. Murphy Kevin P. Machine Learning: A Probabilistic Perspective / Kevin P. Murphy. – MIT Press, 2012. – 1096 с.

16. Ruder S. An overview of gradient descent optimization algorithms / S. Ruder // Cornell University Library. – 2016.

17. Tensorflow. An Open Source Machine Learning Framework for Everyone. – Режим доступа: https://github.com/tensorflow/tensorflow, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 25.06.2019).

18. Tensorboard. TensorFlow's Visualization Toolkit. – Режим доступа: https://github.com /tensorflow/tensorboard, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 01.07.2019).

19. Tomas Mikolov. Distributed Representations of Words and Phrases and their Compositionality / Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean // NIPS'13: Proceedings of the 26th International Conference on Neural Information Processing Systems. – 2013. – Vol. 2. – P. 3111–3119.

20. Tsung-Yi Lin. Focal Loss for Dense Object Detection / Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár // Cornell University Library. – 2017.

References

1. Vlasenko A. V., Dzoban P. I. Razrabotka algoritmov i programm vybora optimalnogo nabora komponent neytralizatsii aktualnykh ugroz na osnove opisaniya modeli i integratsii ikh v WEB-prilozhenie [Development of algorithms and programs to choose the optimal set of components of actual threat neutralization on the basis of model description and their integration in Web appendix]. *Vestnik Adygeyskogo gosudarstvennogo universiteta. Seriya 4: Estestvenno-matematicheskie i tehnicheskie nauki* [Bulletin of Adygea State University. Series 4: Natural-Mathematical and Technical Sciences], 2014, no. 3 (142), pp. 189–193.

2. Chastikova V. A., Volya Ya. I. Analiz effektivnosti raboty algoritma svetlyachkov v zadachakh globalnoy optimizatsii [Firefly algorithm efficiency analisys in case of global optimization problem]. *Trudy Kubanskogo gosudarstvennogo tekhnicheskogo universiteta* [Proceedings of Kuban State Technical University], 2016, no. 15, pp. 105–111.

3. Chastikova V. A., Zherlitsyn S. A. Issledovanie algoritma serykh volkov [Research of the grey wolf algorithm]. *Nauchnye trudy Kubanskogo gosudarstvennogo tekhnicheskogo universiteta* [Proceedings of Kuban State Technical University], 2016, no. 16, pp. 136–142.

4. Chastikova V. A., Druzhinina M. A., Kekalo A. S. Issledovanie effektivnosti algoritma poiska kosyakom ryb v zadache globalnoy optimizatsii [Efficiency research of the fish school search algorithm in the global optimization problem]. *Sovremennye problemy nauki i obrazovaniya* [Modern problems of Science and Education], 2014, no. 4.

5. Chastikova V. A., Malykhyna M. P., Zherlitsyn S. A., Volya Yu. I. Sravnitelnyy analiz nekotorykh algoritmov roevogo intellekta pri obnaruzhenii setevykh atak neyrosetevymi metodami [Comparative analysis of some swarm intelligence algorithms with detection of network attacks using neural network methods]. *Politematicheskiy setevoy elektronnyy nauchnyy zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta* [Poytematic Network Electronic Scientific Journal of Kuban State Agrarian University], 2017, no. 129, pp. 106–115.

6. Chastikova V. A., Zherlitsyn S. A., Volya Yu. I. Neyrosetevoy podkhod k resheniyu zadachi postroeniya fotorobota po slovesnomu opisaniyu [Neural network method of identification by unformalized semantic characteristics]. *Izvestiya Volgogradskogo gosudarstvennogo tekhnicheskogo universiteta* [News of Volgograd State Technical University], 2018, no. 8 (218), pp. 63–67.

7. Chollet F. *Deep Learning with Python.* Manning Publications Co., 2017.

8. *CSE-CIC-IDS2018 on AWS.* Available at: https://www.unb.ca/cic/datasets/ids-2018.html (accessed: 10.09.2019).

9. Geron A. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.* O'Reilly Media, 2017.

10. Gers F. A., Schmidhuber J., Cummins F. Learning to Forget: Continual Prediction with LSTM. *Neural Computation*, 2000, no. 12 (10), pp. 2451–2471.

11. Goodfellow I., Bengio Y., Courville A. *Deep Learning.* MIT Press, 2016.

12. Hochreiter S., Schmidhuber J. Long short-term memory. *Neural computation*, 1997, no. 9 (8), pp. 1735–1780.

13. *KDD Cup 1999 Data.* Available at: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed 25.08.2019).

14. *Keras: Deep Learning for humans.* Available at: https://github.com/keras-team/keras (accessed 21.06.2019).

15. Murphy K. P. *Machine Learning: A Probabilistic Perspective.* MIT Press, 2012.

16. Ruder S. *An overview of gradient descent optimization algorithms.* Cornell University Library, 2016.

17. *Tensorboard. TensorFlow's Visualization Toolkit.* Available at: https://github.com/tensorflow/tensorboard (accessed 01.07.2019).

18. *Tensorflow. An Open Source Machine Learning Framework for Everyone.* Available at: https://github.com/tensorflow/tensorflow (accessed 25.06.2019).

19. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. *NIPS'13: Proceedings of the 26th International Conference on Neural Information Processing Systems*, 2013, vol. 2, pp. 3111–3119.

20. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár. *Focal Loss for Dense Object Detection.* Cornell University Library, 2017.