S. V. Goryainov, Sh. S. Fahmi, S. A. Panov
*Saint Petersburg Electrotechnical University*

DISCRETE OPERATOR INFLUENCE ON THE PROPERTIES OF NONLINEAR DYNAMICAL SYSTEMS

*In this paper, we study the influence of a discrete operator (a numerical integration method) on the time behavior and qualitative properties of nonlinear dynamical systems. Methods of numerical integration of three classes are considered: explicit, implicit and semi-implicit. A methodology for determining the volume of the phase space of nonlinear dynamical systems is presented. A methodology for constructing the implicit midpoint method and the semi-implicit Wehrle method for an arbitrary system of ordinary differential equations is presented. Experimental studies are carried out on two systems: a model of a nonlinear harmonic oscillator and a model that implements the gravitational problem of N bodies. For each model, the results of long-term modeling and estimation of the phase volume of the system using explicit, implicit and semi-implicit methods are presented. The advantages and disadvantages of using each class of numerical integration methods in modeling the considered mathematical systems are presented. Conclusions are drawn about the applicability of the considered methods in modeling nonlinear Hamiltonian systems.*

**Numerical integration methods, phase space volume, N-body problem, discrete operator, harmonic oscillator, Verlet method, implicit midpoint method**

УДК 004.056.5

Т. М. Татарникова, Ф. Бимбетов
*Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)*

П. Ю. Богданов
*Российский государственный гидрометеорологический университет*

# Выявление аномалий сетевого трафика методом глубокого обучения

*Обсуждается применение метода глубокого обучения, основанного на нейронных сетях, в системах обнаружения атак. Обозначены ограничения применения нейронных сетей для классификации трафика на нормальный – не содержащий атаку, и аномальный – содержащий атаку. Ограничения связаны с необходимостью набора данных для обучения нейронной сети, низкой скоростью вычисления нейронной сети при большом количестве входных параметров и влиянием неравномерности распределения выборки примеров из обучающего набора на качество обучения. Предложены способы обхода этих ограничений: выбор значимых информационных признаков, позволяющих классифицировать сетевой трафик, и сохранение значимых примеров обучения, которые представлены малым объемом выборки. Снижать размерность вектора информационных признаков предложено линейным методом с ранжированием признаков по степени важности и в дальнейшем для обучения нейронной сети использовать только «важные» признаки. Сохранение значимых примеров обучения, представленных малым объемом выборки, предложено решить модификацией алгоритма обучения, суть которого сводится к адаптивному присвоению весовых коэффициентов таким примерам. Проведенные эксперименты свидетельствуют об эффективности предложенных метода и алгоритма обучения нейронной сети обнаружению сетевых атак.*

**Сетевая атака, аномалии сетевого трафика, нейронная сеть, глубокое обучение, сокращение параметров, неравномерность количества обучающих примеров, ошибка обучения, точность классификации**

Системы обнаружения атак (СОА) – базовое средство защиты корпоративных информационных ресурсов. Обнаружение атаки подразумевает сначала сбор данных, а затем их анализ средства-

ми СОА. В частности, можно анализировать сведения о передаваемых по сети пакетах данных, производительности программно-аппаратных средств – вычислительной нагрузке на узлы сети, загруженности оперативной памяти, скорости работы прикладного программного обеспечения, доступе к определенным файлам и т. д. [1].

В основе работы СОА лежат специальные методы выявления атак, как правило сигнатурные и/или поведенческие, при этом известно, что использование сигнатурного метода не позволяет обнаруживать новые виды атак и их модификации, а создать точную модель штатного режима функционирования сети таким образом сложно или даже невозможно [2].

Вследствие роста объемов цифровых данных стали актуальны исследования, связанные с применением методов глубокого обучения для обнаружения аномалий сетевого трафика – наличия сетевых атак. В частности, в [3]–[5] предлагается множество параметров, формирующих обучающий набор данных (data set), в [6]–[8] описывается процесс обучения нейронной сети. Привлекательность методов, основанных на нейронных сетях, заключается в возможности их обучения на зарекомендовавшем себя data set, переобучения при появлении дополнительных параметров и самообучения при возникновении ранее не появлявшихся результатов. К известным недостаткам искусственных нейронных сетей относятся необходимость наличия data set и низкая скорость вычисления при большом количестве входных параметров.

**Постановка задачи.** За время применения нейронных сетей в задачах обнаружения аномалий сетевого трафика сформировались несколько наборов data set, например Overview, Aposemat IoT-23, Normal. В статье используется data set NSL-KDD трафика, передаваемого по протоколам TCP, UDP и ICMP. Каждая запись в NSL-KDD представляет собой цепочку сетевых пакетов,

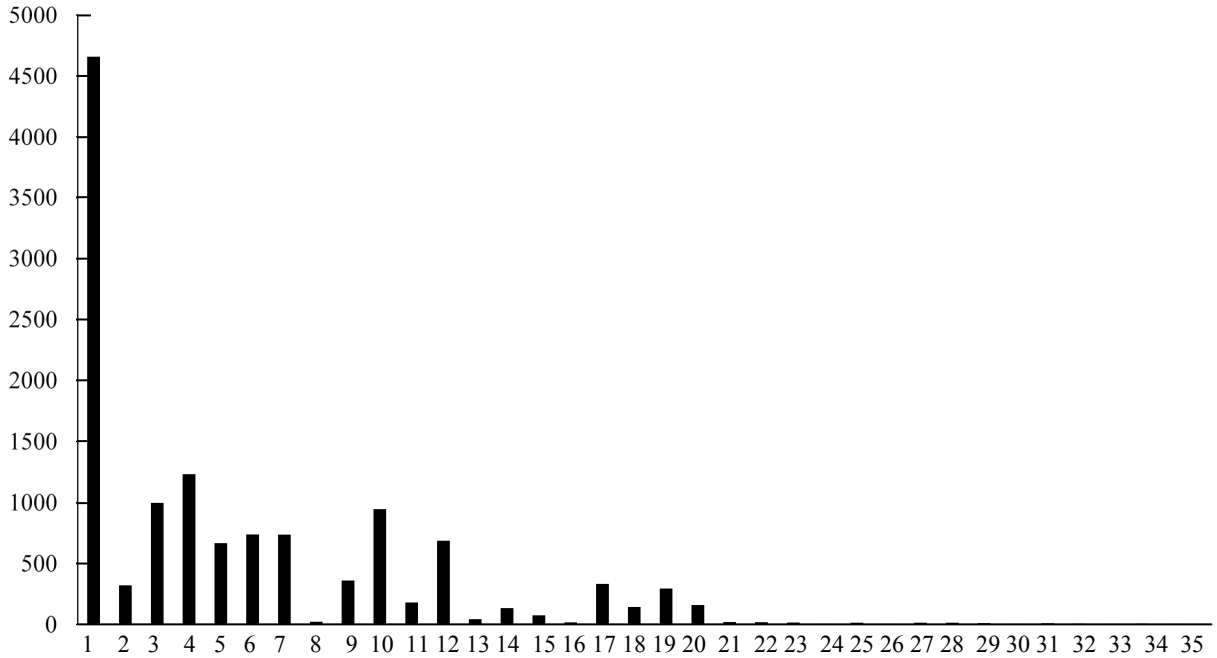


*Рис. 1*

зафиксированных в интервале времени и отправленных от источника к адресату в соответствии с IP-адресами, указанными в заголовке пакета. Записи включают множество параметров $\mathbf{\Pi} = \{x_i\}$, где $i = \overline{1.42}$, в котором $x_1$, …, $x_{41}$ – это информационные признаки, а последний параметр $x_{42}$ – метка класса «атака» или «не атака». Анализ [9], [10] показал, что не все параметры одинаково важны при обнаружении сетевых атак – требуется сокращение размерности $\mathbf{\Pi}$ для повышения скорости работы СОА.

Dataset NSL-KDD содержит 36 типов атак, представляющих 4 категории [11]:

1) Denial of Service (Dos) – атаки, ограничивающие доступ верифицированным пользователям к конкретному сервису через определенный протокол (Back, Land, Neptune, Pod, Smurf, Teardrop, Apache2, Udpstorm, Processtable, Worm);

2) Remote to Local (r2l) – атаки, направленные на получение доступа к локальной машине пользователя из внешней среды (Guess_Password, Ftp_write, Imap, Phf, Multihop, Warezmaster, Warezclient, Spy, Xlock, Xsnoop, Snmpguess, Snmpgetattack, Httptunnel, Sendmail, Named);

3) User to Root (u2r) – атаки, направленные на получение привилегированных прав доступа к машине жертвы (Buffer_overflow, Loadmodule, Rootkit, Perl, Sqlattack, Xterm, Ps);

4) Probe – атаки, направленные на получение сведений об инфраструктуре пользователя (Satan, Ipsweep, Nmap, Portsweep, Mscan, Saint).

В целом NSL-KDD содержит 125 973 записи, предназначенные для обучения, и 22 544 записи для тестирования. Гистограмма распределения типов сетевых атак в dataset NSL-KDD приведена на рис. 1, из которого видно, что количество обучающих примеров по типам атак распределено неравномерно, что влияет на точность обнаружения и, соответственно, на качество работы СОА. Поэтому еще одной актуальной задачей становится сведение к минимуму влияния неравномерности распределения выборки примеров data set на качество обучения [12].

**Пути решения поставленных задач.** Задача *сокращения размерности вектора* $\mathbf{\Pi}$ решена линейным методом, позволившим проранжировать параметры по степени важности на «полезные», «второстепенные» и «бесполезные». Таким образом, из вектора $\mathbf{\Pi} = \{x_1, x_2, ..., x_{41}\}^{\mathrm{T}}$ необходимо получить новый вектор $\mathbf{\Pi}' = \{k_1, k_2, ..., k_c\}^{\mathrm{T}}$. Здесь $c < 41$ и

$$k_i = w_{i,1}x_1 + ... + w_{i,41}x_{41},\ i = \overline{1,c},$$

$$\mathbf{K} = W\mathbf{\Pi},$$

где $W$ – матрица весов линейных преобразований.

Очевидно, что сокращение размерности вектора $\mathbf{\Pi}$ за счет устранения бесполезных параметров позволит ускорить вычисления, производимые нейронной сетью, поскольку сокращается число нейронов входного слоя, и тем самым повысить точность обнаружения сетевых атак благодаря концентрации обучения нейронной сети только на значимых параметрах.

Важность параметров оценивалась эмпирически: сначала за один раз удалялся один параметр, и на полученном data set обучалась и тестировалась нейронная сеть. Во время тестирования фиксировались показатели качества нейронной сети по следующим метрикам:

– точность классификации P (precission):

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}},$$

где TP – количество истинно положительных записей; FP – количество ложноположительных записей;

– скорость обнаружения DR (detection rate):

$$\mathrm{DR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}},$$

где FN – количество ложноотрицательных записей.

Таким образом, в оценке важности каждого параметра участвовали три критерия эффективности: точность классификации – P, время обучения $T_{\text{об}}$ и время тестирования $T_{\text{тест}}$. Ранжирование параметров выполнялось в соответствии с деревом решений, приведенным на рис. 2.

Далее процесс обучения проводился только на «полезных» параметрах, которых в результате ранжирования осталось 10, т. е. количество входных параметров удалось сократить в 4 раза.

Вторая задача – *сведение к минимуму влияния неравномерности распределения выборки примеров data set на качество обучения* – решена модификацией алгоритма обучения. Суть модификации сводится к адаптивному присвоению весовых коэффициентов обучающим примерам, представленным малыми выборками.

Известно, что в процессе обучения нейронной сети веса синапсов корректируются после подачи всей обучающей выборки по усредненному значению градиента E целевой функции:

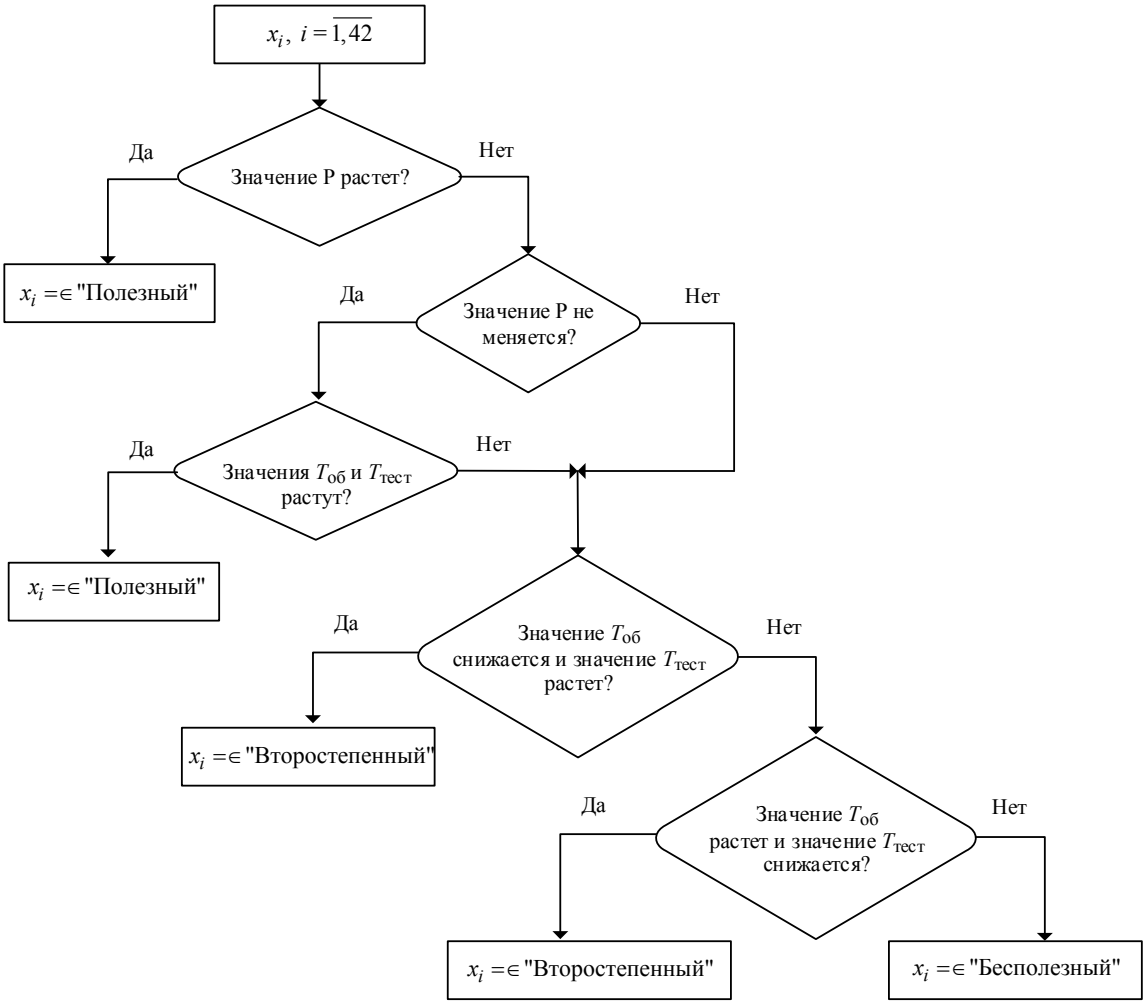


*Рис. 2*

$$\mathrm{E} = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2,$$

где $y_i$ – получаемое значение $i$-го нейрона выходного слоя; $y_i'$ – ожидаемое значение $i$-го нейрона выходного слоя; $n$ – количество нейронов выходного слоя.

Ошибка, найденная по примерам малых выборок в процессе обучения, может потеряться в суммарном значении E по всему data set и, как следствие, эти примеры могут быть проигнорированы. Для сведения к минимуму вероятности появления подобной ситуации предложено обучающим примерам, представленным малыми выборками, присваивать весовые коэффициенты, что в физическом смысле усиливает вклад этих примеров в значение ошибки E. Суть модификации алгоритма обучения демонстрирует рис. 3.

Пусть в начале обучения нейронной сети ошибка E по всему data set не превышает значения $E_2$, а $E_1$ является допустимым значением

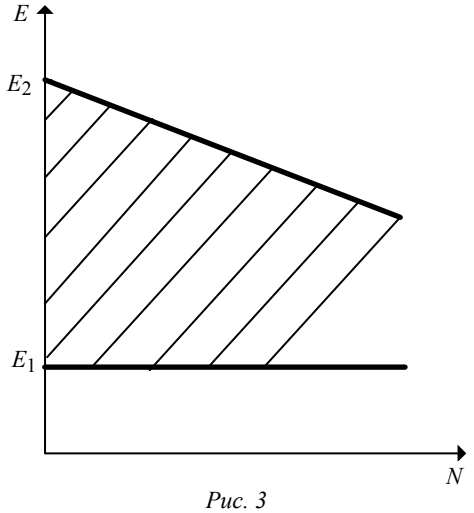


*Рис. 3*

ошибки. В интервале $[E_1, E_2]$ каждый пример на каждой из $N$ итераций процесса обучения вносит в синаптическую карту нейронной сети свои весовые коэффициенты. У большинства примеров значение E начинает снижаться, но у некоторых оно не меняется или меняется незначительно. Когда

ошибка обучающего примера станет $E < E_1$, то пример получит единичный коэффициент, что засвидетельствует его существенный вклад в обучение нейронной сети. Если при дальнейшем обучении для этого же примера $E \geq E_1$, то его весовой коэффициент не изменится, однако при $E > E_2$ пример будет исключен из дальнейшего обучения.

**Эксперименты и оценка результатов.** Структура нейронной сети для решения поставленных задач представляет собой многослойный персептрон с десятью нейронами входного слоя, одним скрытым слоем с двенадцатью нейронами и шестью нейронами выходного слоя, все нейроны – сигмоидального типа. В табл. 1 приведены значения метрик качества нейронной сети.

*Таблица 1*

| Номер метрики | Тип атаки | DR, % | P, % |
|---|---|---|---|
| 1 | neptune | 99.4 | 99.9 |
| 2 | saint | 98.5 | 100.0 |
| 3 | mscan | 92.7 | 98.1 |
| 4 | guess_passwd | 66.4 | 97.0 |
| 5 | smurf | 95.2 | 99.5 |
| 6 | apacher2 | 97.8 | 99.7 |
| 7 | satan | 90.7 | 81.8 |
| 8 | buffer_overflow | 0.0 | 0.0 |
| 9 | back | 96.1 | 97.7 |
| 10 | warezmaster | 16.1 | 98.1 |
| 11 | snmpgetattack | 88.7 | 99.9 |
| 12 | processtable | 85.8 | 98.4 |
| 13 | pod | 82.9 | 70.8 |
| 14 | httptunnel | 98.5 | 100.0 |
| 15 | nmap | 79.45 | 90.6 |
| 16 | ps | 0.0 | 0.0 |
| 17 | snmpguess | 96.3 | 97.9 |
| 18 | ipsweep | 97.9 | 79.3 |
| 19 | mailbomb | 78.7 | 94.9 |
| 20 | portsweep | 89.2 | 61.7 |
| 21 | multihop | 0.0 | 0.0 |
| 22 | named | 0.0 | 0.0 |
| 23 | sendmail | 32.8 | 40.2 |
| 24 | loadmodule | 78.5 | 50.2 |
| 25 | xterm | 89.5 | 45.3 |
| 26 | worm | 0.0 | 0.0 |
| 27 | teardrop | 75.0 | 60.75 |
| 28 | rootkit | 82.4 | 30.4 |
| 29 | xlock | 98.1 | 55.2 |
| 30 | perl | 100.0 | 50.0 |
| 31 | land | 88.1 | 42.15 |
| 32 | xsnoop | 92.0 | 50.2 |
| 33 | sqlattack | 0.0 | 0.0 |
| 34 | ftp_write | 99.9 | 33.4 |
| 35 | udpstorm | 76.7 | 50.0 |
| 36 | phf | 92.7 | 0.0 |

Проверка качества работы нейронной сети для обнаружения атак, представленных в data set NSL-KDD, показало среднюю точность классификации P, равную 59.2 %. После обучения модифицированным алгоритмом среднее значение P стало равным 92.5 %. Значения P для модели нейронной сети, обученной модифицированным алгоритмом, приведены в табл. 2.

*Таблица 2*

| Тип атаки | Количество примеров обучения | *P*, % |
|---|---|---|
| buffer_overflow | 4 | 100.0 |
| pod | 17 | 100.0 |
| ps | 2 | 100.0 |
| multihop | 2 | 100.0 |
| named | 2 | 100.0 |
| sendmail | 14 | 71.1 |
| loadmodule | 2 | 100.0 |
| xterm | 13 | 77.5 |
| worm | 2 | 100.0 |
| teardrop | 12 | 83.3 |
| rootkit | 13 | 84.2 |
| xlock | 9 | 89.1 |
| perl | 2 | 100.0 |
| land | 7 | 86.17 |
| xsnoop | 4 | 75.3 |
| sqlattack | 2 | 100.0 |
| ftp_write | 3 | 67.1 |
| udpstorm | 2 | 100.0 |
| phf | 2 | 100.0 |

Вследствие роста объемов цифровых данных модели глубокого обучения, основанные на нейронных сетях, получили популярность в решении задачи обнаружения сетевых атак. К известным недостаткам модели нейронной сети в сравнении с сигнатурной относятся возможность ложных срабатываний и пропусков реальных атак, а также необходимость наличия качественного data set для обучения. Устранение этих недостатков или сведение их к минимуму связано, во-первых, с выбором значимых информационных признаков, позволяющих классифицировать сетевой трафик на нормальный – не содержащий атаки, и аномальный – содержащий атаку, и, во-вторых, сохранение значимых примеров обучения с малым объемом выборки.

Предложено снизить размерность вектора информационных признаков линейным методом с ранжированием признаков по степени важности и в дальнейшем для обучения нейронной сети использовать только «важные» признаки.

Сохранение значимых примеров обучения, представленных малыми объемом выборки, предложено решить модификацией алгоритма обучения, суть которой сводится к адаптивному присвоению весовых коэффициентов таким примерам.

Проведенные эксперименты свидетельствуют об эффективности предложенных метода и алгоритма обучения нейронной сети обнаружению сетевых атак – точность классификации выросла с 59.2 до 92.5 %. Использование нейросетевой модели совместно с сигнатурным подходом очевидно повысит эффективность СОА, в том числе при обнаружении новых сетевых атак и модификаций существующих.

## СПИСОК ЛИТЕРАТУРЫ

1. Сафронова Е. О., Жук Г. А. Применение искусственных нейронных сетей для прогнозирования DoS атак // Молодой ученый. 2019. № 23. С. 27–30.

2. Бескид П. П., Татарникова Т. М. О некоторых подходах к решению проблемы авторского права в сети интернет // Уч. зап. РГГМУ. 2010. № 15. С. 199–210.

3. Sadek R. A., Soliman M. S., Elsayed H. S. Effective anomaly intrusion detection system based on neural network with indicator variable and rough set reduction // Intern. J. of Computer Science Issues. 2013. Vol. 10, № 6 (2). P. 227–233.

4. Нейросетевая технология обнаружения сетевых атак на информационные ресурсы / Ю. Г. Емельянова, А. А. Талалаев, И. П. Тищенко, В. П. Фроленко // Программные системы: теория и приложения. URL: http://www.mathnet.ru/links/d89cff55f120a931a8d756c39ac40d57/ps38.pdf (дата обращения 27.02.2021).

5. Разработка метода защиты геоинформационных систем и пространственных данных на основе нейронной сети / Т. М. Татарникова, С. Ю. Степанов, Я. А. Петров, А. Ю. Сидоренко // Программные продукты и системы. 2020. № 2. С. 229–235.

6. Татарникова Т. М., Журавлев А. М. Нейросетевой метод обнаружения вредоносных программ на платформе Android // Программные продукты и системы. 2018. № 3. С. 543–547.

7. Guojie L., Jianbiao Z. Research of network intrusion detection based on convolutional neural network // Discrete Dynamics in Nature and Society. 2020. Vol. 2. P. 1–11.

8. Bhattacharjee P., Fujail A., Begum S. Intrusion detection system for NSL-KDD data set using vectorised fitness function in genetic algorithm // Advances in Computational Sciences and Technology. 2017. Vol. 10. P. 235–246.

9. Ingre B., Yadav A., Soni A. K. Decision tree based intrusion detection system for NSL-KDD dataset // Proc. of the Intern. Conf. on Information and Communication Technology for Intelligent Systems. Cham: Springer, 2017. Vol. 2. P. 207–218.

10. NSL-KDD dataset. URL: https://www.unb.ca/cic/datasets/nsl.html (дата обращения 21.02.2021).

11. Chockwanich N., Visoottiviseth V. Intrusion detection by deep learning with TensorFlow // 21st Intern. Conf. on Advanced Communication Technology (ICACT), 2019. P. 654–659.

12. Зуев В., Кемайкин В. Модифицированный алгоритм обучения нейронных сетей // Программные продукты и системы. 2019. Т. 32, № 2. С. 258–262.

T. M. Tatarnikova, F. Bimbetov
*Saint Petersburg Electrotechnical University*

P. Yu. Bogdanov
*Russian State Hydrometeorological University*

## IDENTIFYING NETWORK TRAFFIC ANOMALIES BY DEEP LEARNING

*The article discusses the application of a deep learning method based on neural networks for implementation in intrusion detection systems. The restrictions on the use of neural networks for classifying traffic into normal - not containing an attack and anomalous - containing an attack are indicated. The limitations are related to the need for a data set for training a neural network, a low computation speed of a neural network with a large number of input parameters, and the influence of the uneven distribution of the sample data set on the quality of training. Methods for circumventing these restrictions are proposed: the choice of significant information features that allow the classification of network traffic and the preservation of significant training examples, which are represented by a small sample size. It is proposed to solve the reduction of the dimension of the vector of information features by a linear method with ranking the features according to the degree of importance and in the future to use only «important» features to train the neural network. It is proposed to solve the preservation of significant learning examples represented by a small sample size by modifying the learning algorithm, the essence of which is reduced to adaptive assignment of weighting coefficients to such examples. The experiments carried out indicate the effectiveness of the proposed method and algorithm for training a neural network in detecting network attacks.*

**Network attack, network traffic anomalies, neural network, deep learning, parameter reduction, uneven number of training examples, training error, classification accuracy**