ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ
INFORMATION TECHNOLOGY AND TELECOMMUNICATIONS

УДК 004.056
DOI: 10.21822/2073-6185-2023-50-1-53 -61 Оригинальная статья/Original Paper

Применение методов машинного обучения для автоматизированного обнаружения сетевых вторжений

М.В. Бабичева, И.А. Третьяков
Донецкий национальный университет,
283001, г. Донецк, ул. Университетская, 24, Россия

Резюме. Цель. Целью исследования является развитие автоматизированных систем обнаружения сетевых атак, способных адаптироваться под постоянно меняющийся характер сетевых атак и новые виды угроз. В основе таких систем должны быть использованы алгоритмы и модели машинного обучения, которые способны выявлять сложные зависимости между данными в процессе обучения **Метод.** Для обучения моделей была подготовлена выборка с признаками нормального и аномального трафика, причем она была прорежена и сбалансирована в результате предварительного статистического анализа. Отобраны пять алгоритмов машинного обучения и протестированы, как на обучающем множестве признаков, так и на реальном тестовом множестве, полученном экспериментально. По результатам экспериментов отобран классификатор случайного леса, показавший наилучшие результаты. **Результат.** Разработана модель для обнаружения сетевых вторжений, которая показала точность обнаружения на реальном трафике 0,99. **Вывод.** Показано, что система обнаружения сетевых вторжений на основе машинного обучения может решить проблему гибкой защиты, которая могла бы адаптироваться под постоянно меняющийся характер сетевых атак, поскольку одним из самых важных преимуществ машинного обучения в выявлении сетевых вторжений является способность обучаться признакам атак и выявлять случаи, которые не характерны для тех, что наблюдались раннее.

Ключевые слова: информационная безопасность, сетевые атаки, сетевой трафик, машинное обучение, модель обнаружения вторжений, автоматизированные системы.

Для цитирования: М.В. Бабичева, И.А. Третьяков. Применение методов машинного обучения для автоматизированного обнаружения сетевых вторжений. Вестник Дагестанского государственного технического университета. Технические науки. 2023; 50(1):53-61. DOI:10.21822/2073-6185-2023-50-1-53-61

Application of machine learning methods for automated detection of network intrusions

M.V. Babicheva, I.A. Tretyakov
Donetsk National University,
24 Universitetskaya Str., Donetsk 283001, Russia

Abstract. Objective. Development of automated network attack detection systems capable of adapting to the ever-changing nature of network attacks and new types of threats. Such systems should be based on machine learning algorithms and models that are able to identify complex dependencies between data in the learning process. **Method.** To train the models, a sample with signs of normal and abnormal traffic was prepared, and it was thinned and balanced as a result of preliminary statistical analysis. Five machine learning algorithms were selected and tested, both on a training set of features and on a real test set obtained experimentally. Based on the results of the experiments, a random forest classifier was selected, which showed the best results. **Result.** A model for detecting network intrusions has been developed, which showed a detection accuracy of 0.99 on real traffic. **Conclusion.** It is shown that a machine learning-based network intrusion detection system can solve the problem of flexible protection that could adapt to the ever-changing nature of network attacks, since one of the most im-

portant advantages of machine learning in detecting network intrusions is the ability to learn the signs of attacks and identify cases that are uncharacteristic of those that were observed earlier.

Keywords: information security, network attacks, network traffic, machine learning, intrusion detection model, automated systems

For citation: M.V. Babicheva, I.A. Tretyakov. Application of machine learning methods for automated detection of network intrusions. Herald of the Daghestan State Technical University. Technical Science. 2023; 50 (1): 53-61. DOI: 10.21822 /2073-6185-2023-50-1-53-61

Введение. В современном мире в связи с быстрыми темпами развития информационных технологий стремительно увеличивается объем сетевого трафика, что приводит к постоянному увеличению угроз сетевой безопасности, увеличению количества нарушений информационной безопасности и созданию новых типов атак. Однако в свою очередь появляются новые и более эффективные методы защиты от сетевых атак [1-6].

Таким образом, процесс выявления сетевых вторжений является актуальной научно-технической задачей и ориентирован на противодействие угрозам сетевой безопасности и предотвращение нарушений информационной безопасности. Противодействие актуальным угрозам сетевой безопасности в настоящее время невозможно без развития автоматизированных систем обнаружения атак, способных адаптироваться под постоянно меняющийся характер сетевых атак и новые виды угроз. Следовательно, в основе таких систем должны быть использованы алгоритмы и модели машинного обучения, способные выявлять сложные зависимости между данными в процессе обучения, и являющиеся перспективным инструментом для решения задач защиты информации [7-12].

Постановка задачи. Учитывая вышесказанное, для построения модели автоматизированной системы обнаружения сетевых вторжений на основе машинного обучения, необходимо:

- выбрать наиболее подходящий набор данных для обучения системы обнаружения сетевых вторжений и подходящую для решения поставленной задачи модель;
- подготовить наборы данных для обучения и провести предварительную обработку данных, сформировав признаковое пространство;
- проверить работоспособность настроенной и обученной модели на реальных данных, проанализировать полученные результаты.

Методы исследования. Выбор набора данных и формирование признакового пространства. Для обучения модели обнаружения сетевых атак был использован один из наиболее актуальных наборов данных CIC-IDS2017 [13]. В табл. 1 представлен фрагмент данного набора.

Таблица 1. Фрагмент набора данных для обучения CIC-IDS2017
Table 1. A fragment of the training dataset CIC-IDS2017

№	Название файла File name	Атаки Attacks	Количество записей Number of records
1	Monday-WorkingHours.pcap_ISCX.csv	Benign (обычный трафик)	529918
2	Tuesday-WorkingHours.pcap_ISCX.csv	Benign, FTP-Patator, SSH-Patator	445909
3	Wednesday-workingHours.pcap_ISCX.csv	Benign, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slowloris, Heartbleed	692703
4	Thursday-WorkingHours-Morning-WebAttacks.pcap_ ISCX.csv	Benign, Web Attack – Brute Force, Web Attack – Sql Injection, Web Attack – XSS	170163
5	Thursday-WorkingHours-Afternoon-Infilteration.pcap_ ISCX.csv	Benign, Infiltration	288602

Этот набор данных содержит как реальный фоновый трафик (нормальный трафик), так и трафик, содержащий разнообразные атаки. Он удобен тем, что включает в себя предобработанные файлы в формате CSV, содержащие размеченные сессии с выделенными признаками в разные дни наблюдения.

В рамках данной работы была выбрана только одна обучающая выборка с классом сетевыъ атак – веб-атаки (Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv). Каждая запись в нем представляет собой сетевую сессию и характеризуется 84 признаками. На рис. 1 представлены основные признаки наличия атаки.

```
[ ]   1 model['Label'].value_counts()

BENIGN                        393427
PortScan                      158944
DDoS                          128027
Web Attack - Brute Force        1507
Web Attack - XSS                 652
Web Attack - Sql Injection        21
Name: Label, dtype: int64
```

Рис. 1. Выбранные классы атак

Fig. 1. Selected attack classes

На рис. 1 присутствуют такие атаки, как «Сканирование портов», DDOS, «Брутфорс», XSS и SQL-инъекции. Причем, для каждой атаки представлено различное количество признаков. Выборка была проанализирована и обработана: исключены повторяющиеся признаки (например, «Fwd Header Length»), удалены записи с null значениями идентификатора сессии «Flow ID», нечисловые значения признаков, неопределенные значения и бесконечные значения значениями заменены на -1, строковые значения приведены к числовым, закодированы ответы в обучающей выборке (0 – нет атаки, 1 – есть атака), на выходе модели. Однако такая обработка привела к несбалансированности выборки. А такой дисбаланс может в дальнейшем усложнить обучение модели. На рис. 2 показано расхождение между обучающими и тестовыми данными в несбалансированной выборке.

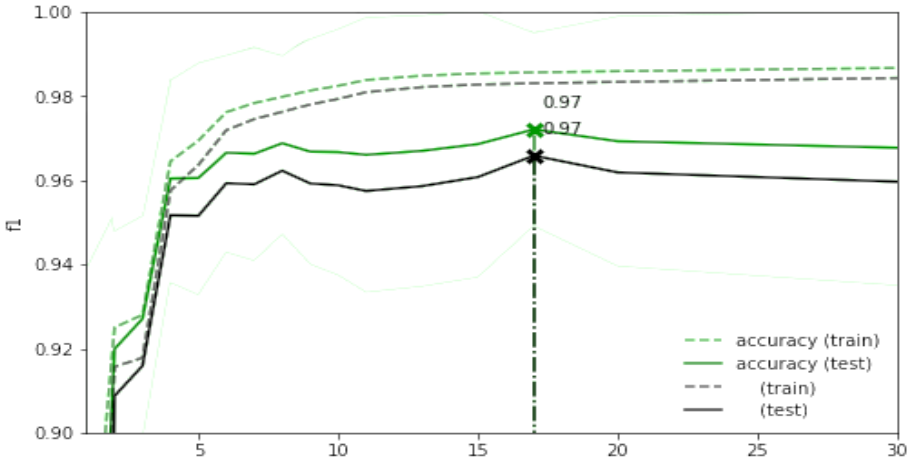


Рис. 2. Расхождение между обучающими и тестовыми данными в несбалансированной выборке

Fig. 2. Discrepancy between training and test data in an unbalanced sample

Из рис. 2 видно, что accuracy для обучающих данных гораздо выше, чем для тестовой. Дисбаланс был устранен методом случайного сэмплирования, а именно субдискретизацией. В результате эмпирически были выбраны коэффициенты 70% и 30% (наличие и отсутствие атаки). Кроме того, были исключены такие признаки, как «Flow ID», «Source IP», «Source Port», «Destination Port», «Protocol», «Timestamp», так как для обнаружения атаки они не важны. После субдискретизации выборка стала намного более сбалансиро-

ванной, и расхождение между обучающими и тестовыми данными устранено, что показано на рис. 3.

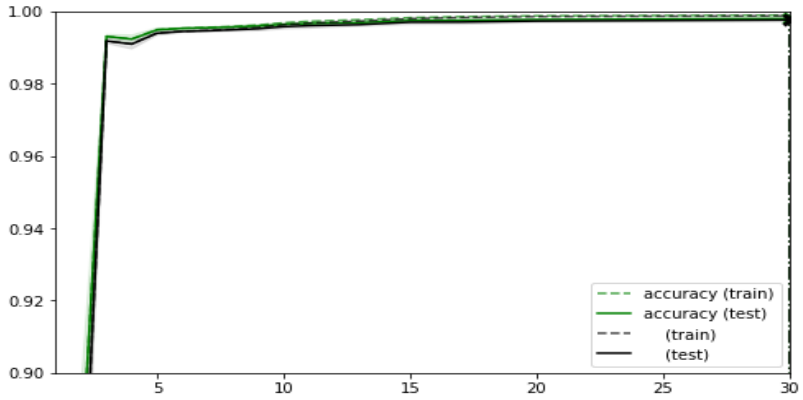


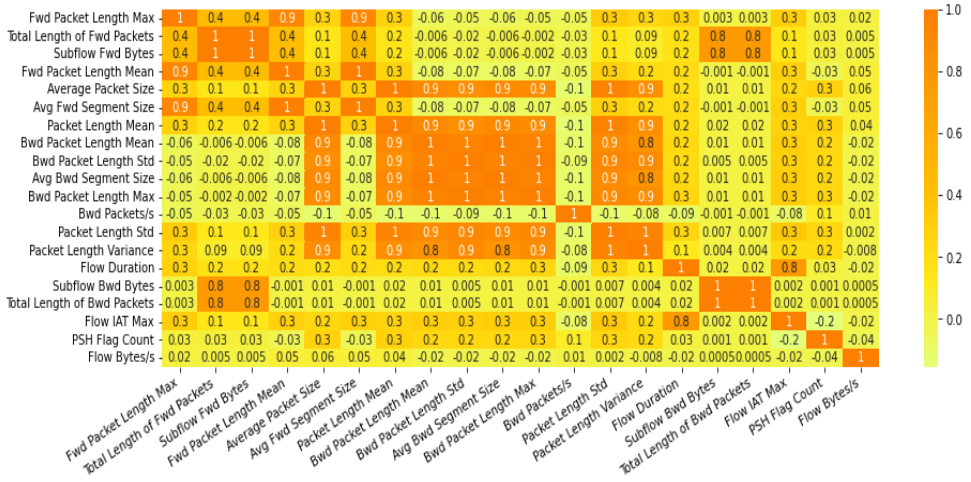
Рис. 3. Расхождение между обучающими и тестовыми данными в сбалансированной выборке
Fig. 3. Discrepancy between training and test data in a balanced sample

Для анализа наиболее значимых признаков использовался классификатор случайного леса Random Forest Classifier [14]. В процессе исследования и настройки модели было выбрано оптимальное количество деревьев 250, увеличение количества деревьев приводило к переобучению. Последующий анализ выделенных признаков показал, что некоторые признаки не имеют практической значимости, и их количество сократилось до 20. На рис. 4 представлены отобранные признаки в порядке убывания значимости.

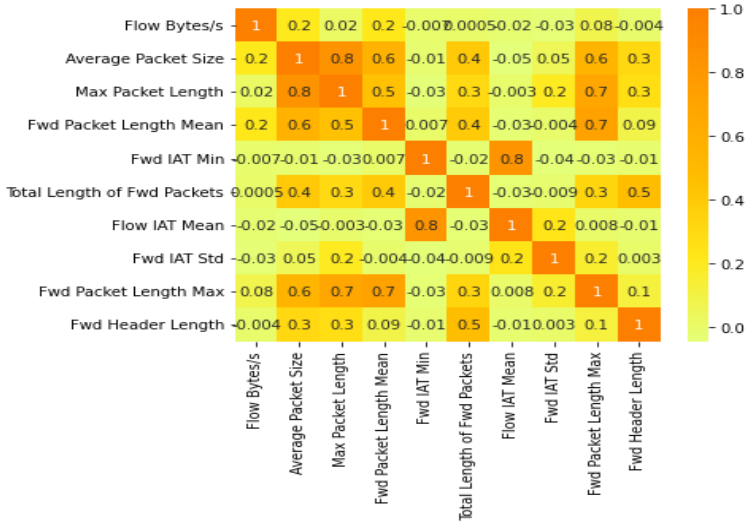
Рис. 4. Наиболее значимые признаки, отобранные классификатором с последующей корректировкой
Fig. 4. The most significant features selected by the classifier with subsequent adjustment

Для сокращения признакового пространства были рассчитаны коэффициенты корреляции Пирсона, которые позволили определить наличие зависимости между парами признаков и удалить 1 признак из пары. Согласно результатам корреляционного анализа, из пространства признаков были исключены следующие признаки: Avg Fwd Segment Size, Subflow Fwd Bytes, Avg Bwd Segment Size, PSH Flag Count, Subflow Bwd Bytes, ACK Flag Count, Packet Length Std, Packet Length Mean. В итоге осталось 10 наиболее значимых признаков. Матрицы корреляции до и после удаления признаков, с обнаруженной зависимостью представлена на рис. 5.

Проверка работоспособности модели на реальных данных. Для проверки корректности работы модели на реальном сетевом трафике был использован сетевой анализатор Wireshark и сниффер (язык программирования C#, среда разработки Microsoft Visual Studio), который может выделять признаки TCP сессии и сразу формировать набор данных, который далее используется в модели. По итогу работы сниффера формируются три папки dataset, pcap, session. В папке dataset находятся выделенные признаки сессии (то есть необходимый набор данных для модели).



а)



б)

Рис. 5. Корреляционные матрицы коэффициентов Пирсона до (а) и после (б) удаления признаков с наибольшими коэффициентами

Fig. 5. Correlation matrices of Pearson coefficients before (a) and after (b) removal of features with the highest coefficients

В папке pcap находятся файлы с трафиками из Wireshark, а в папке sessions соответственно отдельные сессии. Сниффер можно запустить вместо Wireshark и сразу же выделять из сетевого трафика сессии и признаки TCP сессии.

В качестве атакуемого устройства был использован компьютер Intel Pentium ® CPU G630, 2,70 GHz, на котором был запущен сетевой анализатор Wireshark. В этом же момент с другого такого же устройства проводились атаки. Проверка корректности работы модели проводилась в пять этапов:

1. Проверка распознавания сканирования портов. Для осуществления проверки распознавания сканирования портов на другом устройстве использовалось приложение для сканирования портов Port Scanner. После файл сетевого трафика направлялся на сниффер для выделения признаков TCP сессии, а затем в созданную модель обнаружения сетевых вторжений.
2. Проверка распознавания DDoS атаки. Для проведения DDoS атаки использовалось приложение Termux, в котором был загружен скрипт, написанный на Python, способный осуществить DDoS атаку. DDoS атаки проводились с разной протяженностью по времени. На рис. 6 можно увидеть работу данного приложения.

Рис. 6. Работа приложения Termux
Fig. 6. Operation of the Termux application

3. Проверка распознавания нескольких атак одновременно. Чтобы проверить распознавание нескольких разных атак на компьютер, программы, использовавшиеся ранее, запускались с другого устройства одна за другой.
4. Проверка распознавания веб-атак (Brute Force, Sql-инъекция, XSS). Для проверки распознавания веб-атак из открытых источников был взят файл с сетевым трафиком, который содержит такие атаки, как Brute Force, Sql-инъекция, XSS. Основываясь на исследованиях, уже проведенных другими авторами, можно сказать, что именно веб-атаки являются одними из наиболее сложных для распознавания системами обнаружения атак на основе нейронных сетей. С помощью сниффера из этого файла были выделены признаки TCP сессии, и полученный набор данных был использован в созданной модели обнаружения. С помощью сниффера из этого файла были выделены признаки TCP сессии, и полученный набор данных был использован в созданной модели обнаружения. Количественный состав файла с веб-атаками и результат работы модели представлены на рис. 7.

а)

```
Общее время работы: 0.0095977783203125 seconds
{'BENIGN': 5282,
 'PortScan': 4,
 'Web Attack - Brute Force': 1322,
 'Web Attack - Sql Injection': 11,
 'Web Attack - XSS': 648}
```

б)

```
BENIGN                        5087
Web Attack - Brute Force      1507
Web Attack - XSS               652
Web Attack - Sql Injection      21
Name: Label, dtype: int64
```

Рис. 7. Количественный состав файла с веб-атаками (а) и результат работы модели (б)
Fig. 7. The quantitative composition of the file with web attacks (a) and the result of the model (b)

5. Проверка распознавания сетевого трафика без атак для обнаружения ложных срабатываний. Поскольку созданная модель обнаружения сетевых вторжений не является идеально верной, необходимо проверить будут ли ложные срабатывания в сетевом трафике без наличия атак. Для осуществления проверки был взят нормальный сетевой трафик за определенное количество времени. По результатам проверок можно сделать вывод, что модель не имеет ложных срабатываний при нормальном сетевом трафике.

Выбор алгоритма классификатора. Для решения поставленной задачи рассматривались алгоритмы, представленные в табл. 2. Это основные алгоритмы, которые используются в современном машинном обучении для классификации. А задача обнаружения сетевых вторжений в данном исследовании сводится к разбиению признаков на классы, которые соответствуют тем или иным атакам. Причем, целесообразно чтобы количество классов определялось самими алгоритмом, поскольку в реальных условиях невозможно предугадать, сколько различных признаков атак содержит подозрительный трафик. Качество ответов классификаторов определялось по доле правильных ответов модели обнаружения сетевых вторжений (точности) на выборке для обучения и собственной сбалансированной выборке. Также учитывалось время работы алгоритма, поскольку время

обучения является весьма существенным фактором для работы модели. Для исследования модели была выбрана библиотека sklearn. Полученные значения точности и времени обучения для обучающей выборки приведены в табл.2.

Таблица 2. Результаты оценки качества классификаторов
Table 2. Results of classifier quality assessment

Алгоритм Algorithm	Точность Accuracy	Время выполнения, с Lead time
Дерево решений (CART, sklearn.tree.DecisionTreeClassifier)	0.997	22
Метод k ближайших соседей (KNN, sklearn.neighbors.KNeighborsClassifier)	0.976	30
Случайный лес (RF, sklearn.ensemble.RandomForestClassifier)	0.998	120
Байесовский классификатор (NB, sklearn.naive_bayes.GaussianNB)	0.867	46
Метод опорных векторов (SVM, sklearn.svm.SVC)	0.789	480

Наилучшие результаты на выборке для обучения показали дерево решений и случайный лес, однако на реальном сетевом трафике, в ходе описанных выше экспериментов, случайный лес дал лучший результат, чем дерево решений, хотя и уступал дереву решений в скорости. Поэтому для построения модели использовался алгоритм Random Forest Classifier. Число деревьев подбиралось экспериментально, в цикле, с проверкой попадания признака атаки в класс из тестовой выборки.

Обсуждение результатов. После формирования признакового пространства для обучения модели использовался Random Forest Classifier, который показал хорошие результаты на предварительных испытаниях, с числом деревьев 50. Точность на выходе модели на обучающем множестве признаков составила 0.98. В качестве эксперимента, для оценки работоспособности модели был взят нормальный трафик (без атак). На рис. 8 представлены результаты обнаружения атак для нормального трафика (атак не обнаружено) и аномального трафика (обнаружено 2170 признаков атак).

```
Total operation time: 0.0146853923797607 42 seconds
Benign records detected (0), attacks detected (1):
{0: 59}
```

а)

```
Total operation time: 0.040031909942626 95 seconds
Benign records detected (0), attacks detected (1):
{0: 5097, 1: 2170}
```

б)

Рис. 8. Результат работы модели на нормальном (а) и аномальном (б) трафике
Fig. 8. The result of the model's operation on normal (a) and abnormal (b) traffic

Для сравнения точности обнаружения атак использовался также метод k-ближайших соседей (KNN) [15]. Для этого алгоритма точность на выходе модели составляла 0.97, что ненамного отличалось от результатов случайного леса. Но при эксперименте с нормальным трафиком модель показала не существующие 5 атак. Этот результат предоставлен на рис. 9.

```
Total operation time: 0.0053243637084960 94 seconds
Benign records detected (0), attacks detected (1):
{0: 54, 1: 5}
```

Рис. 9. Результат работы модели к-ближайших соседей
Fig. 9. The result of the k-nearest neighbor model

Вывод. В ходе работы была разработана модель для обнаружения сетевых вторжений, которая показала точность обнаружения на реальном трафике 0,99. Для обучения моделей была подготовлена выборка с признаками нормального и аномального трафика, причем она была прорежена и сбалансирована в результате предварительного статистического анализа. Были отобраны пять алгоритмов машинного обучения и протестированы,

как на обучающем множестве признаков, так и на реальном тестовом множестве, полученном экспериментально. По результатам экспериментов был отобран классификатор случайного леса Random Forest Classifier, как показавший наилучшие результаты.

Качество проверки корректности работы модели на реальном сетевом трафике проводилось в пять этапов: проверка распознавания сканирования портов, DDoS атак, веб-атак нескольких атак одновременно и сетевого трафика без атак. Распознавание сканирования портов происходит с высокой точностью (0,99), как при быстрой DDoS атаке, так и при медленной модель может их распознать, но количество записей не соответствует реальному объему на 17%, веб-атаки являются наиболее сложными для распознавания, но, тем не менее, большинство записей (70%) с ними были найдены правильно, модель не имеет ложных срабатываний при нормальном сетевом трафике.

В работе показано, что система обнаружения сетевых вторжений на основе машинного обучения может решить проблему гибкой защиты, которая могла бы адаптироваться под постоянно меняющийся характер сетевых атак, поскольку одним из самых важных преимуществ машинного обучения в выявлении сетевых вторжений является способность обучаться признакам атак и выявлять случаи, которые не характерны для тех, что наблюдались ранее. К проблеме, которая требует дальнейшего рассмотрения, относится время, затраченное на обучение модели, потому что для такого рода систем важно, чтобы они постоянно доучивались, поскольку в реальном мире постоянно появляются новые виды угроз. Результаты исследования могут быть интересны разработчикам DLP-систем, системным администраторам и DevOPs инженерам наряду с уже существующими средствами предотвращения сетевых вторжений.

Библиографический список:

1. Sheluhin, O. I. Comparative analysis of informative features quantity and composition selection methods for the computer attacks classification using the unsw-nb15 dataset / O. I. Sheluhin, V. P. Ivannikova // T-Comm. – 2020. – Vol. 14. – No. 10. – P. 53-60. – DOI: 10.36724/2072-8735-2020-14-10-53-60.
2. Yang, W. Security detection of network intrusion: application of cluster analysis method / W. Yang // Computer Optics. – 2020. –Vol. 44. – No. 4. – P. 660-664. – DOI: 10.18287/2412-6179-CO-657.
3. Третьяков И. А. Оптимизация SQL-запросов / И. А. Третьяков, Е. Н. Кожекина, И. В. Журавлёв // Вестник Донецкого национального университета. Серия Г: Технические науки. – 2021. – № 2. – С. 39-49. – EDN: RPSKQQ.
4. Третьяков И. А. Безопасность облачных технологий на тестируемом WEB сервере / И. А. Третьяков, Е. Н. Кожекина, Б. В. Гайван // Вестник Донецкого национального университета. Серия Г: Технические науки. – 2021. – № 3. – С. 49-62. – EDN: IVEBAS.
5. Safonov L. Unsupervised anomaly detection in network traffic using deep autoencoding gaussian mixture model / L. Safonov // International Journal of Open Information Technologies. – 2021. –Vol. 9. – No. 9. – P. 109-112.
6. Третьяков И. А. Выявление проблем безопасности веб-сайтов посредством DoS-атаки / И. А. Третьяков, Е. Н. Кожекина, К. Е. Лебедев // Вестник Донецкого национального университета. Серия Г: Технические науки. – 2022. – № 1. – С. 19-32. – EDN: UOSGYS.
7. Черникова Е. И. Анализ сетевого трафика с использованием метода машинного обучения / Е. И. Черникова // Аллея науки. – 2019. – Т. 1. – № 6(33). – С. 921-925. – EDN BSYXMW.
8. Кажемский М. А. Многоклассовая классификация сетевых атак на информационные ресурсы методами машинного обучения / М. А. Кажемский, О. И. Шелухин // Труды учебных заведений связи. – 2019. – Т. 5. – № 1. – С. 107-115. – DOI: 10.31854/1813-324X-2019-5-1-107-115.
9. Liu, H. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey / H. Liu, B. Lang // Applied Sciences. – 2019. – Vol. 9. – No. 20:4396. – DOI: 10.3390/app9204396.
10. Utkin, L. V. A deep forest classifier with weights of class probability distribution subsets / L. V. Utkin, M. S. Kovalev, A. A. Meldo // Knowledge-Based Systems. – 2019. – Vol. 173. – P. 15-27. – DOI: 10.1016/j.knosys.2019.02.022.
11. Машинное обучение для анализа и классификации шифрованного сетевого трафика / В. А. Мулюха, Л. Ю. Лабошин, А. А. Лукашин, Н. В. Нашивочников // Международная конференция по мягким вычислениям и измерениям. – 2020. – Т. 1. – С. 238-241. – EDN XYQCZP.
12. Ahmed, H. A. Network intrusion detection using oversampling technique and machine learning algorithms / H. A. Ahmed, A. Hameed, N. Z. Bawany // PeerJ Computer science. – 2022. – Vol. 8. – No. e820. – DOI: 10.7717/peerj-cs.820.
13. Intrusion Detection Evaluation Dataset (CIC-IDS2017). [Электронный ресурс] – URL: https://www.unb.ca/cic/datasets/ids-2017.html (Дата обращения 22.12.2022)

14. Scikit-learn. sklearn.ensemble.RandomForestClassifier. [Электронный ресурс] – URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (Дата обращения 22.12.2022)
15. Метод k-ближайших соседей (k-nearest neighbour). [Электронный ресурс] – URL: https://proglib.io/p/metod-k-blizhayshih-sosedey-k-nearest-neighbour-2021-07-19 (Дата обращения 22.12.2022)

References:

1. Sheluhin O. I. Comparative analysis of informative features quantity and composition selection methods for the computer attacks classification using the unsw-nb15 dataset. O. I. Sheluhin, V. P. Ivannikova. *T-Comm.* 2020; 14(10): 53-60. – DOI: 10.36724/2072-8735-2020-14-10-53-60.
2. Yang, W. Security detection of network intrusion: application of cluster analysis method. *Computer Optics.* 2020; 44(4): 660-664. – DOI: 10.18287/2412-6179-CO-657.
3. Tretiakov I. A. Optimization of SQL queries / I. A. Tretiakov, E. N. Kozhokina, I. V. Zhuravlev. *Bulletin of Donetsk National University. Series G: Technical Science*s. 2021;2: 39-49. – EDN: RPSKQQ. (In Russ)
4. Tretiakov, I. A. Security of cloud technologies on the tested WEB server / I. A. Tretiakov, E. N. Kozhikina, B. V. Gaivan. *Bulletin of Donetsk National University. Series G: Technical Sciences*. 2021;3: 49-62. – EDN: IVEBAS. (In Russ)
5. Safonov L. Unsupervised anomaly detection in network traffic using deep autoencoding gaussian mixture model / L. Safonov. *International Journal of Open Information Technologies.* 2021; 9(9): 109-112.
6. Tretiakov, I. A. Identification of website security problems through DoS attacks / I. A. Tretiakov, E. N. Kozhekina, K. E. Lebedev. *Bulletin of Donetsk National University. Series G: Technical Sciences*. 2022;1: 19-32. – EDN: UOSGYS. (In Russ)
7. Chernikova E. I. Network traffic analysis using machine learning method. *Alley of Science*. 2019; 1(6(33)): 921-925. – EDN BSYXMW. (In Russ)
8. Kezhemsky, M. A. Multiclass classification of network attacks on information resources by machine learning methods / M. A. Kazhemsky, O. I. Shelukhin. *Proceedings of educational institutions of communication.* 2019; 5(1):107-115. – DOI: 10.31854/1813-324X-2019-5-1-107-115. (In Russ)
9. Liu, H. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey / H. Liu, B. Lang. *Applied Sciences*. 2019; 9(20):4396. – DOI: 10.3390/app9204396.
10. Utkin, L. V. A deep forest classifier with weights of class probability distribution subsets / L. V. Utkin, M. S.Kovalev, A.A. Meldo.*Knowledge-Based Systems.* 2019;173:15-27. – DOI: 10.1016/j.knosys.2019.02.022.
11. Machine learning for analysis and classification of encrypted network traffic / V. A. Mulukha, L. Yu. Lapshin, A. A. Lukashin, N. V. Nashivochnikov. *International Conference on Soft Computing and Measurements.* 2020;1: 238-241. – EDN XYQCZP. (In Russ)
12. Ahmed H. A. Network intrusion detection using oversampling technique and machine learning algorithms / H.A. Ahmed, A. Hameed, N. Z. Bawany. *PeerJ Computer science.* 2022; 8(8):20. DOI: 10.7717/peerj-cs.820.
13. https://www.unb.ca/cic/datasets/ids-2017.html (accessed 22.12.2022)
14. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (accessed 22.12.2022)
15. https://proglib.io/p/metod-k-blizhayshih-sosedey-k-nearest-neighbour-2021-07-19 (accessed 22.12.2022)

Сведения об авторах:
Бабичева Маргарита Вадимовна, старший преподаватель кафедры радиофизики и инфокоммуникационных технологий; mv.babicheva60@gmail.com
Третьяков Игорь Александрович, заместитель декана по научной работе, доцент кафедры радиофизики и инфокоммуникационных технологий; i.tretiakov@mail.ru ORCID 0000-0002-7816-1563
Information about authors:
Margarita V. Babicheva, Senior Lecturer, Department of Radiophysics and Information and Communication Technologies; mv.babicheva60@gmail.com
Igor A. Tretyakov, Deputy Dean for Research, Associate Professor of the Department of Radiophysics and Infocommunication Technologies; i.tretiakov@mail.ru ORCID 0000-0002-7816-1563

Конфликт интересов/Conflict of interest.
Авторы заявляют об отсутствии конфликта интересов/The authors declare no conflict of interest.
Поступила в редакцию/Received 16.01.2023.
Одобрена после рецензирования/ Reviced 12.02.2023.
Принята в печать/Accepted for publication 12.02.2023.