**МНОГОКЛАССОВАЯ КЛАССИФИКАЦИЯ СЕТЕВЫХ АТАК МЕТОДАМИ**

**ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА**

***Кононов Руслан Валерьевич,***

*магистрант МТУСИ, Москва, Россия,*

[*dremming@mail.ru*](mailto:dremming@mail.ru)

***Шелухин Олег Иванович,***

*заведующий кафедрой ИБ МТУСИ, д.т.н., профессор, МТУСИ, Москва, Россия,*

*sheluhin@mail.ru*

**Аннотация**

*В статье рассматривается использование алгоритма интеллектуального анализа в задачах обнаружения и многоклассовой классификации аномалий. Актуальность данной проблемы вызвана развитием информационных и коммуникационных технологий, влекущих за собой рост количества уязвимостей. Приведено описание процесса создания искусственной нейронной сети прямого распространения - многослойный перцептрон. Произведена оценка полученных результатов несколькими алгоритмами. Показано, что созданное программное обеспечение обладает высокой эффективностью в решении поставленных задач***.**

***Ключевые слова:*** *защита информации, информационная безопасность, интеллектуальный анализ, информационные системы, уязвимости, атаки, нейронные сети.*

**Введение**

Сегодня информационные системы повсеместно внедряются на предприятия, что служит увеличению количества возможных уязвимостей. Для противодействия каждому типу уязвимости, обнаруженному в информационной системе на предприятии, требуется выполнение ряда действий со стороны сотрудников службы безопасности. Поэтому, правильная классификация угроз является важным этапом в обеспечении информационной безопасности, в том числе при разработке систем обнаружения вторжений. Наиболее эффективные системы обнаружения вторжений представляют собой сложные аппаратно-программные комплексы. Например, для решения задач обнаружения и последующей классификации аномалий часто используются такие технологии, как накопление статистических данных и *Deep Packet Inspection*, что требует больших вычислительных мощностей [1].

Также, для обнаружения и классификации аномалий используются алгоритмы интеллектуального анализа [2]. Данные алгоритмы могут самообучаться и противостоять новому типу угроз при сравнительно небольших требованиях к вычислительным мощностям.

Целью статьи является исследования эффективности использования методов интеллектуального анализа в задачах обнаружения и классификации аномалий.

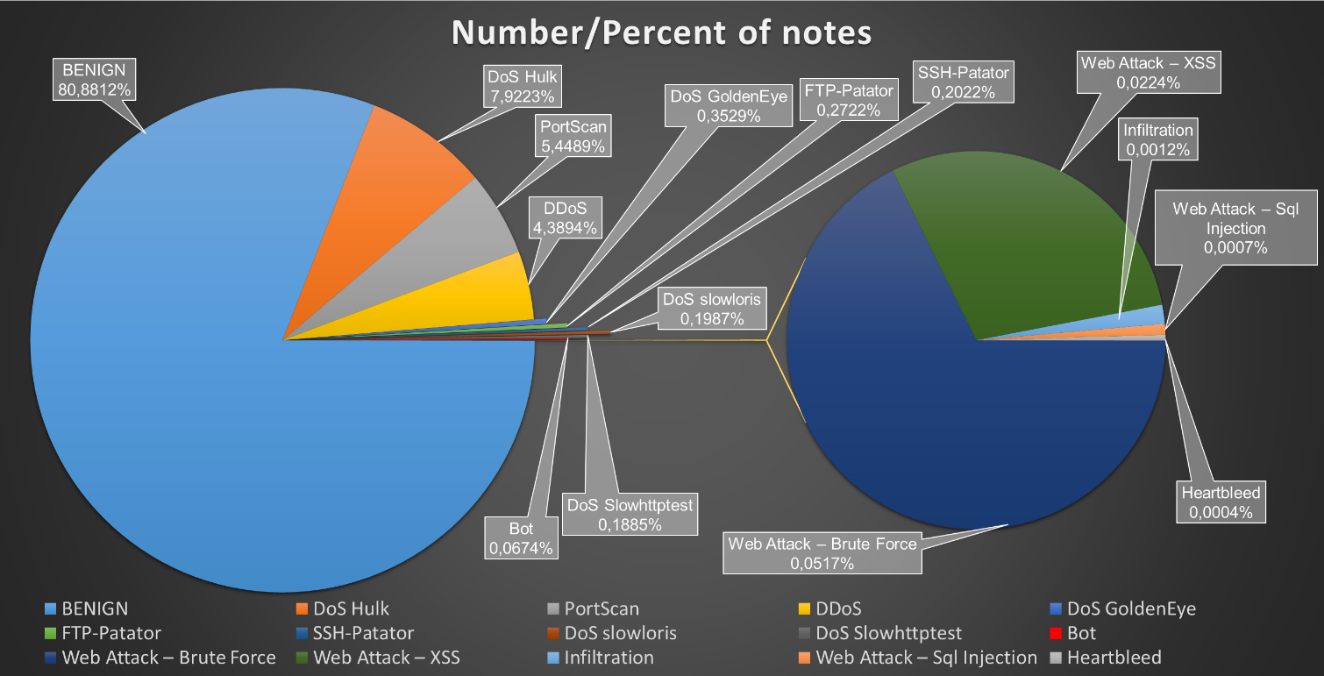
**Описание разрабатываемого программного обеспечения**

Для решения задач обнаружения и многоклассовой классификации аномалий используется размеченный набор данных *CICIDS 2017*, который содержит информацию о 14 различных типах трафика [3]. Каждая запись об атаке характеризуется следующими атрибутами: 60 параметров и одна метка трафика. В выбранном наборе данных атаки делятся на шесть основных профилей, которые были сформированы на основе новейшего списка наиболее распространенных семейств атак:

* *Brute Force Attack* – это один из наиболее популярных типов атак, который может быть использован не только для взлома пароля, но также для обнаружения скрытого материала в веб-приложениях;
* *Heartbleed Attack* – это атака, использующая уязвимость, которая позволяет читать оперативную память небольшими фрагментами, размер которых составляет не более 64 КБ. Проблема заключается в том, что уязвимость двухсторонняя, а значит, пользователь может получить данные с уязвимого сервера, а сервер злоумышленника также может получить данные с оперативной памяти пользователя;
* *Botnet* – группа устройств, подключенных к сети Интернет, может быть использована для выполнения вредоносных задач, таких как кража данных, передача вредоносных программ, спам или запуск атак, предоставляющих злоумышленнику доступ к устройству;
* *DoS* атака – данный тип атак осуществляется следующим образом: несколько компьютеров отправляют огромное количество запросов в сеть-жертву, перегружая ее ресурсы и нарушая ее работу. Сеть, использованная для организации *DoS* атаки, как правило, представляет собой некоторое количество зараженных компьютеров;
* *Web Attack* – количество атак, произведенных таким способом, растет с каждым днем. Именно поэтому, организации и частные лица все сильнее задумываются о полноценной защите своих сетей. Существует огромное количество программ и исполняемых кодов, созданных для сканирования компьютера с целью обнаружения уязвимостей в системе безопасности, позволяющих проникнуть в сеть жертвы.
* *Infiltration Attack* – это атака, которая осуществляется изнутри сети-жертвы. Данный тип атак использует уязвимое программное обеспечение, например, *Adobe Acrobat Reader*. После успешного проникновения в сеть, вирус будет запущен на компьютере жертвы и сможет проводить различные атаки на сеть жертвы.

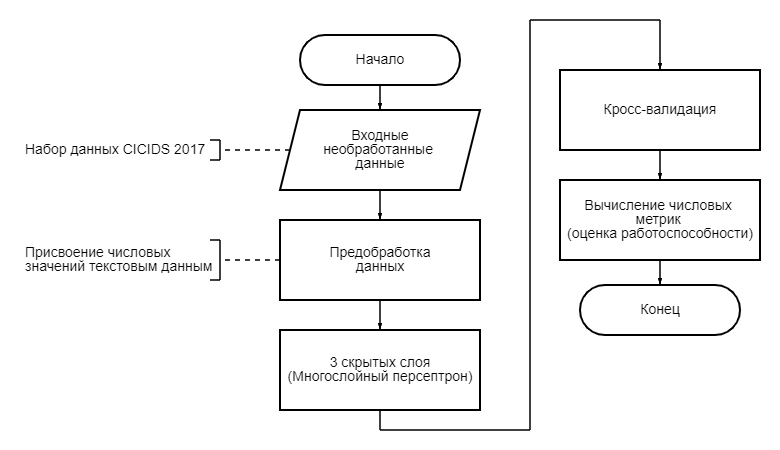
На рисунке 1 представлено соотношение классов трафика в рассматриваемом наборе данных. Для обучения и тестирования ИНС в работе был использован заранее оптимизированный набор данных. Набор готов к использованию в задачах обучения и тестирования нейронных сетей. В качестве основной архитектуры для решения задач обнаружения и многоклассовой классификации аномалий был выбран многослойный перцептрон. Данный тип архитектуры характерен тем, что каждый нейрон предыдущего слоя связан с каждым нейроном следующего слоя [4].

Полный цикл разработки нейронной сети начинается с определения схемы алгоритма. На вход нейронной сети подается набор данных *CICIDS 2017.* Учитывая, что данный набор содержит текстовую информацию, необходимо предварительно обработать его с помощью кодировщика *LabelEncoder* [5]*.* Далее, используется алгоритм кросс-валидации для проверки работоспособности модели. Следующим шагом работы алгоритма является оценка качества построенной ИНС путем расчета числовых метрик.



**Рис 1.** Соотношение классов атак в наборе данных *CICIDS 2017*

На рисунке 2 представлена блок-схема алгоритма обнаружения и многоклассовой классификации аномалий.



**Рис 2.** Блок-схема алгоритма выявления сетевых атак

Программная реализация осуществлялась средствами языка программирования Python и дополнительных библиотек, таких как *Scikit Learn, LabelEncoder, Pandas, Numpy, Copy, PyTorch, Logging*.

*Scikit Learn* – это один из наиболее широко используемых пакетов *Python* для *Data Science* и *Machine Learning* [6]. Данная библиотека использовалась для разбиения набора данных на две части:

* *train* – данная выборка использовалась для обучения искусственной нейронной сети. Размер выборки составляет 80% от общего значения;
* *test* – данная выборка нужна для конечной оценки работоспособности нейронной сети.

Для решения проблемы невозможности распознавания текстовой информации алгоритмами интеллектуального анализа, было принято решение об использовании библиотеки *LabelEncoder*. С помощью данной библиотеки был создан кодировщик, способный преобразовывать текстовые значения в числовые. В наборе данных *CICIDS* *2017* присутствует текстовый параметр ‘*Label*’, который идентифицирует тип трафика. Таким образом, каждому типу трафика было присвоено собственное числовое значение.

*Pandas* — это специальная библиотека, которая позволяет проводить различные действия с временными рядами и числовыми таблицами [7]. Данная библиотека была использована в задачах чтения и объединения нескольких *csv* файлов набора данных *CICIDS* *2017*.

*Numpy* – это специальный модуль для *Python*, позволяющий использовать математические и числовые операции в виде пре-скомпилированных функций, которые объединяются в пакеты более высокого уровня [8]. Данная библиотека была использована для простейших операций, таких как формирование случайного значения и обозначение простейших констант.

Модуль *Copy* позволяет осуществлять два типа операций копирования: глубокое копирование и поверхностное копирование [9]. Данный модуль был использован в задачах глубокого копирования, которое применялось в случае, если только что обученная модель показывала лучший результат в сравнении с предыдущей моделью. Использование модуля *Copy* позволило получить наиболее правильно обученную искусственную нейронную сеть.

*PyTorch* – это библиотека, которая облегчает создание моделей глубокого обучения [10]. При создании искусственной нейронной сети модуль *PyTorch* был использован в следующих задачах:

* использование функционала параллельных вычислений;
* использование таких утилит, как: *DataLoader* - для загрузки набора данных в ИНС, *RandomSampler* для получения содержания выборок в случайном порядке и *SequentialSampler* для получения содержания выборок в последовательном порядке [10,11];
* использование модуля *torch.nn* для получения доступа к удобному инструменту создания полносвязных нейронных сетей;
* использование метода «отсева» для решения проблемы переобучения ИНС;
* оптимизация нейронной сети с помощью изменения параметров скорости обучения.

Модуль *Logging —* это специальный инструмент, используемый в языке *Python* и предназначенный для логгирования хода работы программы [12]. Данный модуль был использован для ведения журнала о ходе обучения искусственной нейронной сети.

После этапа предобработки данных используется полносвязный слой нейронов, имеющий функцию активации *ReLu*. В качестве метода оптимизации используется алгоритм *AdamW*. Внешний цикл обучения нейронной сети состоит из 6 повторений. Внутренний тренировочный цикл проходит через все тренировочные данные в партиях, размер которых равен 64. Коэффициент скорости обучения нейронной сети был выбран равным 0,00005.

На рисунке 3 представлена структура ИНС с 71 признаком входных данных, 3 скрытыми слоями и 15 результирующими выходами.

В процессе обучения и тестирования производилась настройка модели нейронной сети путем изменения количества скрытых слоев размера *batch-size* и количества нейронов на каждом из слоев. Результаты данных тестов представлены в таблице 1.

Для тестирования работоспособности нейронной сети использовалось четыре показателя: *F1 Score, Precision, Recall* [13]*.* Исходя из полученных данных, можно сделать вывод о том, что наилучшая конфигурация нейронной сети состоит из 5 эпох обучения и 3 скрытых слоев.

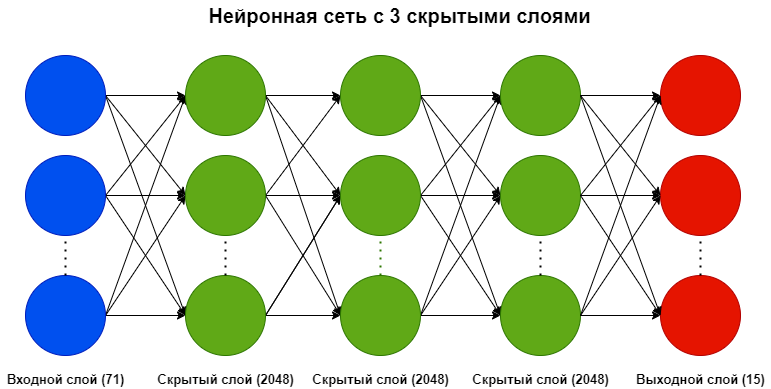


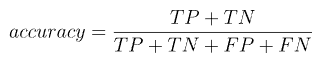
Рис 3. Структура НС с 3 скрытыми слоями

Таблица 1

**Результаты тестирования различных конфигураций ИНС**

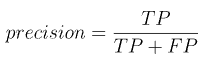
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Количество эпох | Количество скрытых слоев | *Accuracy* | *Precision* | *Recall* | *F1 score* |
| 2 | 2 | 0.9188 | 0.9217 | 0.9043 | 0.9114 |
| 3 | 2 | 0.9532 | 0.9252 | 0.9654 | 0.9364 |
| 4 | 2 | 0.9076 | 0.9171 | 0.8932 | 0.9079 |
| 5 | 2 | 0.9261 | 0.9113 | 0.9258 | 0.9149 |
| 6 | 2 | 0.9153 | 0.9216 | 0.9211 | 0.9112 |
| 7 | 2 | 0.9330 | 0.9148 | 0.9332 | 0.9165 |
| 8 | 2 | 0.9353 | 0.9295 | 0.9143 | 0.9229 |
| 3 | 3 | 0.9204 | 0.9214 | 0.9112 | 0.9215 |
| 4 | 3 | 0.9264 | 0.9062 | 0.9304 | 0.9077 |
| 5 | 3 | 0.9396 | 0.9311 | 0.9521 | 0.9428 |

Метрика *Accuracy(1.1)* необходима для определения процентного соотношения правильных ответов, которые дала ИНС [14]. Данная метрика высчитывается с помощью отношения суммы *True Positive* и *True Negative* к сумме всех значений матрицы ошибок.



(1)

Метрика *Precision(1.2)* представляет собой отношение числа верно предсказанных ответов с меткой *Positive,* вне зависимости от достоверности предсказанного значения [15].



(2)

*Recall(1.3)* рассчитывается как отношение числа *Positive* выборок, корректно классифицированных как *Positive*, к общему количеству *Positive* ответов [16]. *Recall* измеряет способность модели обнаруживать выборки, относящиеся к классу *Positive*. Чем выше *Recall*, тем больше *Positive* ответов было найдено*.*



*.*  (3)

Метрика *F1 score(1.4)* необходима для нахождения наилучшего сочетания *Recall* и *Precision* [17]*.*



(4)

Исходя из результатов и учета степени важности метрики *Recall,* наилучшую эффективность показала нейронная сеть в следующей конфигурации: 3 эпохи обучения и 2 скрытых слоя.

Рисунок 4 иллюстрирует полученные результаты тестирования различных конфигураций ИНС.

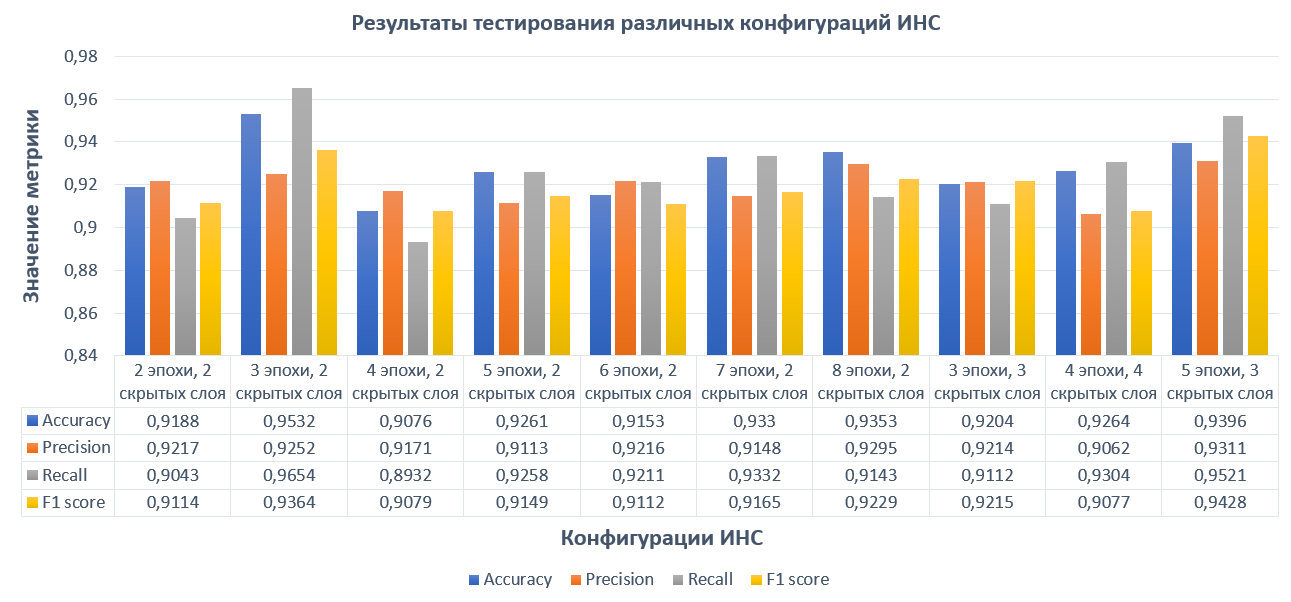


Рис 4. Результаты тестирования различных конфигураций ИНС

**Выводы**

В результате исследований было установлено, что искусственная нейронная сеть прямого распространения многослойный перцептрон показала высокую точность в рассматриваемой задаче обнаружения и многоклассовой классификации аномалий. Полнота классификации трафика составила 96,54%. Полученные результаты свидетельствуют о высокой надежности разработанного алгоритма.

**Литература**

1. *Шелухин О. И., Ерохин С.Д., Полковников МВ.* Технологии машинного обучения в сетевой безопасности. / Под ред. О. И. Шелухина. – М.: Горячая линия – Телеком, 2021. – 360 с;
2. *Sheluhin O.I., Ivannikova V.P.* (2020) Comparative analysis of informative features quantity and composition selection methods for the computer attacks classification using the UNSW-NB15 dataset. T-Comm, vol. 14, no.10, pр. 53-60. (in Russian);
3. *Todor Balabanov, liyan Zankinski, Kolyu Kolev,* "Multilayer Perceptron Training Randomized by Second Instance of Multilayer Perceptron", 2018, [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/329680882\_Multilayer\_Perceptron\_Training\_Randomized\_by\_Second\_Instance\_of\_Multilayer\_Perceptron, свободный – (16.03.2022);
4. Confusion Matrix, Accuracy, Precision, Recall F1 Score [Электронный ресурс] // medium.com // URL: https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63c, свободный – (16.03.2022);
5. Sklearn.preprocessing.LabelEncoder [Электронный ресурс] URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html, свободный – (16.03.2022);
6. Что такое Scikit Learn - гайд по популярной библиотеке Python для начинающих [Электронный ресурс] URL: https://datastart.ru/blog/read/chto-takoe-scikit-learn-gayd-po-populyarnoy-biblioteke-python-dlya-nachinayuschih, свободный – (16.03.2022);
7. Введение в pandas: анализ данных на Python [Электронный ресурс] URL: https://khashtamov.com/ru/pandas-introduction/, свободный – (16.03.2022);
8. NumPy в Python [Электронный ресурс] URL: https://habr.com/ru/post/352678/, свободный – (16.03.2022);
9. Модуль copy - поверхностное и глубокое копирование объектов [Электронный ресурс] URL: https://pythonworld.ru/moduli/modul-copy.html, свободный – (16.03.2022);
10. A Comprehensive Guide to the DataLoader Class and Abstractions in PyTorch [Электронный ресурс] URL: https://blog.paperspace.com/dataloaders-abstractions-pytorch/, свободный – (16.03.2022);
11. PyTorch — ваш новый фреймворк глубокого обучения [Электронный ресурс] URL: https://habr.com/ru/post/334380/, свободный – (16.03.2022);
12. Модуль Logging в Python [Электронный ресурс] URL: https://dev-gang.ru/article/modul-logging-v-python-sugk5e4d8u/, свободный – (16.03.2022);
13. *Шелухин О.И., Симонян А.Г., Ванюшина А.В.* Влияние структуры обучающей выборки на эффективность классификации приложений трафика методами машинного обучения // T-Comm: Телекоммуникации и транспорт. 2017. Т. 11. № 2. С. 25–31;
14. Основные метрики задач классификации в машинном обучении [Электронный ресурс] URL: https://webiomed.ru/blog/osnovnye-metriki-zadach-klassifikatsii-v-mashinnom-obuchenii/, свободный – (16.03.2022);
15. Multi-Class Metrics Made Simple, Part I: Precision and Recall [Электронный ресурс] URL: https://towardsdatascience.com/multi-class-metrics-made-simple-part-i-precision-and-recall-9250280bddc2 , свободный – (16.03.2022);
16. Precision and recall [Электронный ресурс] URL: https://en.wikipedia.org/wiki/Precision\_and\_recall, свободный – (16.03.2022);
17. Micro, Macro & Weighted Averages of F1 Score, Clearly Explained [Электронный ресурс] URL: https://towardsdatascience.com/micro-macro-weighted-averages-of-f1-score-clearly-explained-b603420b292f, свободный – (16.03.2022);

**MULTICLASS CLASSIFICATION OF NETWORK ATTACKS BY METHODS OF INTELLECTUAL ANALYSIS**

***Ruslan V. Kononov,***

*Graduate MTUCI, Moscow, Russia,*

*[dremming@mail.ru](mailto:dremming@mail.ru)*

***Oleg I. Sheluhin***

*Head of the Department of IS of MTUCI, Doctor of Technical Sciences,*

*Professor, MTUCI, Moscow, Russia,*

*sheluhin@mail.ru*

**Abstract**

*The article discusses the use of the algorithm of intelligent analysis in the problems of detection and multiclass classification of anomalies. The urgency of this problem is caused by the development of information and communication technologies, which entail an increase in the number of vulnerabilities. The process of creating an artificial neural network of direct propagation - a multilayer perceptron is described. The results obtained were evaluated by several algorithms. It is shown that the created software has high efficiency in solving the tasks.*

***Keywords:*** *information security, information security, intelligent analysis, information systems, vulnerabilities, attacks, neural networks.*