# ИНФОРМАТИКА
# ИНФОРМАЦИОННЫЕ СИСТЕМЫ

УДК 004.032.26 004.7.056.53

# Разработка приложения для анализа сетевого трафика и обнаружения сетевых атак

**А. Д. Иванов, А. А. Кутищев, Е. Ю. Никитина**
Пермский государственный национальный исследовательский университет
Россия, 614990, г. Пермь, ул. Букирева, 15
mr.ivlex@gmail.com; 8-963-870-08-33

Продемонстрировано применение нейронных сетей при разработке сетевых систем обнаружения вторжений, описана структура разработанного приложения для анализа сетевого трафика и обнаружения сетевых атак, приведены результаты работы приложения.

**Ключевые слова:** *информационная безопасность; система обнаружения вторжений; искусственные нейронные сети; сетевой анализатор.*

DOI: 10.17072/1993-0550-2021-2-57-64

## Введение

На сегодняшний день задача обнаружения сетевых атак является одной из самых актуальных в области информационной безопасности. Ее значимость возрастает с каждым днем благодаря постоянному увеличению объемов передаваемой информации посредством компьютерных сетей, количества пользователей и усложнения методов атак злоумышленников.

Одним из способов обеспечения защиты от сетевых атак является использование сетевых систем обнаружения вторжений, предназначенных для обнаружения факта проведения сетевых атак на защищаемые ресурсы. Кроме того, в зависимости от конкретной реализации, в функции системы обнаружения вторжений может входить применение мер по предотвращению обнаруженной атаки. Эффективность таких систем зависит от используемых методов анализа имеющейся информации о сетевых атаках.

Одним из таких методов является использование нейронных сетей для анализа данных. С помощью данного решения становится возможным создать систему, которая будет способна эффективно определять, как существующие, так и неизвестные ранее атаки, распознавать аномальный трафик и совершенствоваться в процессе своей работы, не требуя при этом вмешательства человека.

Основная идея работы заключается в разработке самостоятельного приложения, достаточного для сбора информации обо всем проходящем трафике через конкретный узел сети, анализа и классификации данного сетевого трафика. Одним из наиболее эффективных средств классификации собранной информации являются искусственные нейронные сети [1].

© Иванов А.Д., Кутищев А.А., Никитина Е.Ю., 2021

## 1. Разработка приложения для анализа сетевого трафика и обнаружения вторжений

Разрабатываемое приложение будет состоять из следующих модулей: сетевой анализатор, нейросетевой модуль, хранилище информации о сетевом трафике и API.

Сетевой анализатор необходим для получения данных обо всем трафике, проходящем через узел сети.

Полученная информация будет преобразована к единому формату и записана в хранилище информации о сетевом трафике.

Нейросетевой модуль выполняет функцию анализа данных о сетевом трафике, определения вида трафика (корректный трафик или сетевая атака) и типа сетевой атаки при помощи методов машинного обучения. Результат анализа будет также записан в хранилище информации о сетевом трафике.

Каждая запись в хранилище будет хранить информацию о параметрах сессии, результат обработки данных нейросетевым модулем и результат проверки эксперта. После получения результатов анализа данных нейросетевым модулем и экспертом, полученные записи могут быть использованы для дообучения нейронной сети (рис. 1).

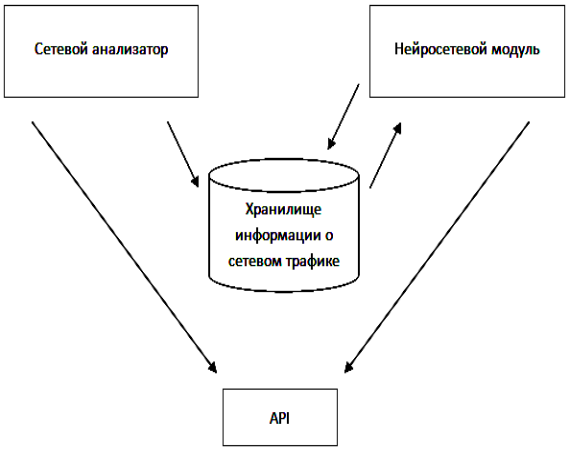


Рис. 1. *Структура разрабатываемого приложения*

Нейронные сети для обнаружения аномалий обучаются в течение некоторого периода времени, когда все наблюдаемое поведение считается нормальным. После обучения нейронная сеть запускается в режиме распознавания.

В ситуации, когда во входном потоке не удается распознать корректное (нормальное) поведение, фиксируется факт атаки.

В случае использования репрезентативной обучающей выборки нейронные сети дают хорошую устойчивость в пределах заданной системы; но составление подобной выборки является серьезной и сложной задачей [2].

Для обнаружения действий нарушителей, нейронную сеть необходимо обучить на выборке, содержащей основные виды сетевых атак. После обучения нейронная сеть запускается в режиме распознавания. При наличии во входном потоке сетевых пакетов с параметрами, схожими с параметрами одного из видов сетевых атак из обучающей выборки, фиксируется факт и тип атаки.

В данном случае, составление репрезентативной выборки является еще более сложной задачей, чем в случае обнаружения аномалий. Поэтому в качестве выборки в данной ситуации необходимо взять одну из хорошо известных выборок для решения задачи обнаружения сетевых атак с помощью нейронных сетей.

Разработка сетевой системы обнаружения вторжений с использованием методов машинного обучения, обычно включает следующие три основных этапа [3]:

- этап предварительной обработки данных;
- этап обучения;
- этап тестирования.

Сначала сетевой трафик, захваченный при помощи специального программного средства – анализатора трафика, предварительно обрабатывается, производится преобразование к единому формату, который можно использовать для обучения нейронной сети. Происходит нормализация данных, кодирование символьных параметров, удаление неполных и/или некорректных наборов данных.

На втором этапе происходит разделение полученного набора данных на две части: обучающее множество и тестовое множество. Обычно исходный набор данных разделяется в соотношении 80 % к 20 % или 85 % к 15 % соответственно.

После этого происходит обучение нейронной сети с помощью обучающего сформированного множества.

На завершающем этапе происходит тестирование полученной модели с использованием тестового множества и определение точности определения атак данной моделью.

## 2. Проектирование нейросетевого модуля

Для обнаружения трафика атак в разрабатываемом приложении будет использоваться искусственная нейронная сеть прямого распространения.

Для разработки нейросетевого модуля необходимо два основных компонента: набор данных, который будет использован для обучения и тестирования нейронной сети, и инструмент для создания и обучения нейронной сети.

В качестве набора данных для обучения и тестирования нейронной сети был выбран набор данных о сетевых вторжениях CSE-CIC-IDS2018 [4].

В качестве инструмента для создания и обучения нейронной сети была выбрана библиотека PyTorch для языка Python.

### 2.1. Выбор набора данных

При выборе набора данных для обучения нейронной сети были рассмотрены следующие варианты:

- Knowledge Discovery in Database (KDD) Cup 1999 Data [5];
- UNSW-NB-15 [6];
- CSE-CIC-IDS2018.

Основным фактором выбора набора данных для обучения нейронной сети являлась его актуальность, так как устаревшие наборы данных могут содержать невоспроизводимые на настоящий момент и неиспользуемые на сегодняшний день примеры сетевых атак.

Также важную роль играет количество представленных классов сетевых атак, количество примеров по каждому из данных классов и количество выделенных признаков сетевых атак.

Для выбора оптимального набора данных была составлена сравнительная характеристика, представленная в табл. 1.

Таблица 1. *Сравнительная характеристика наборов данных*

| Набор данных | KDD Cup 1999 | UNSW-NB-15 | CSE-CIC-IDS-2018 |
|---|---|---|---|
| Год создания | 1999 | 2015 | 2018 |
| Количество классов сетевых атак | 22 | 9 | 14 |
| Количество признаков сетевых атак | 41 | 47 | 79 |
| Общее количество примеров | Около 5 млн. | Около 2,5 млн | Около 16 млрд |
| Общее количество примеров сетевых атак | Около 4 млн. | Около 300 тыс. | Около 2,7 млн |
| Количество классов атак, количество примеров по которым составляет менее 5% от общего количества примеров атак | 20 | 5 | 6 |

По результатам сравнения был выбран набор данных CSE-CIC-IDS2018, так как является наиболее актуальным, содержит наибольшее количество признаков сетевых атак и большее количество классов атак, представленных достаточным количеством примеров.

Общедоступный набор данных CSE-CIC-IDS2018, созданный совместно Организацией по обеспечению безопасности связи (Communications Security Establishment, CSE) и Канадским институтом кибербезопасности (Canadian Institute for Cybersecurity, CIC), содержит данные о легальных сетевых соединениях и 14 классах сетевых атак. Каждая запись набора содержит 79 параметров о сетевом соединении.

### 2.2. Проектирование структуры нейронной сети с полным набором параметров

Было рассмотрено два типа структуры нейронной сети: 3-слойный персептрон с одним скрытым слоем и 4-слойный персептрон с двумя скрытыми слоями.

Для обучения нейронной сети из всего набора данных CSE-CIC-IDS2018 было составлено множество, которое затем было разбито на обучающее, тестовое и проверочное множества в отношении 70 %, 15 % и 15 % соответственно. Состав полученных множеств представлен в табл. 2.

Таблица 2. *Состав обучающего, тестового и проверочного множеств*

| Класс атаки | Количество примеров в наборе | Количество примеров в обучающем множестве | Количество примеров в тестовом и проверочном множествах |
|---|---|---|---|
| Benign | 13484708 | 471965 | 101135 |
| Bot | 286191 | 10016 | 2146 |
| Infilteration | 161934 | 5668 | 1214 |
| DoS attack-SlowHTTPTest | 139890 | 4896 | 1049 |
| DoS attack-Hulk | 461912 | 16167 | 3464 |
| DoS attack-GoldenEye | 41508 | 1453 | 311 |
| DoS attack-Slowloris | 10990 | 384 | 83 |
| DDoS attack-LOIC-UDP | 1730 | 1210 | 260 |
| DDoS attacks-LOIC-HTTP | 576191 | 20167 | 4321 |
| DDoS attack-HOIC | 686012 | 24010 | 5145 |
| Brute Force-XSS | 230 | 162 | 34 |
| Brute Force-Web | 611 | 427 | 92 |
| FTP-Brute Force | 193360 | 6768 | 1450 |
| SSH-Brute Force | 187589 | 1210 | 260 |
| SQL Injection | 87 | 61 | 13 |

Подбор количества нейронов на скрытом слое производился экспериментальным путем, так как в настоящее время не существует каких-либо жестких и эффективных правил для выбора их количества. Обычно при решении реальных задач количество нейронов скрытого слоя составляет от N до 3N, где N – количество нейронов в выходном слое сети [7].

Для поиска предположительно оптимальных значений количества нейронов на скрытых слоях было использовано эвристическое правило геометрической пирамиды. Согласно данному правилу, число нейронов единственного скрытого слоя в 3-слойном персептроне вычисляется по следующей формуле [8]:

$$k = \sqrt{nm}, \quad (1)$$

где $k$ – число нейронов в скрытом слое, $n$ – число нейронов во входном слое, $m$ – число нейронов в выходном слое.

Для 4-слойного персептрона используется следующий набор формул:

$$r = \sqrt[3]{\frac{n}{m}}, \quad (2)$$

$$k_1 = mr^2, \quad (3)$$

$$k_2 = mr, \quad (4)$$

где $k_1$ – число нейронов в первом скрытом слое, $k_2$ – число нейронов во втором скрытом слое.

На основании данных формул были рассчитано число нейронов на скрытых слоях для 3- и 4-слойного персептрона. Они представлены в табл. 3.

Таблица 3. *Число нейронов в скрытых слоях, рассчитанное по правилу геометрической пирамиды*

| Количество нейронов во входном слое | Количество нейронов в выходном слое | Количество нейронов в скрытом слое 3-слойного персептрона | Количество нейронов в первом скрытом слое 3-слойного персептрона | Количество нейронов во втором скрытом слое 4-слойного персептрона |
|---|---|---|---|---|
| 76 | 15 | 34 | 44 | 25 |

Далее, относительно каждого полученного значения были составлены интервалы для экспериментального подбора наиболее оптимального количества нейронов на скрытых слоях. Они приведены в табл. 4.

В табл. 5 и табл. 6 представлены 5 лучших результатов обучения 3-слойного персептрона и 5 лучших результатов обучения 4-слойного персептрона соответственно.

В каждом случае в качестве функции ошибки использовалась функция Cross-EntropyLoss, в качестве функции активации скрытого слоя использовалась функция ReLu, в качестве функции активации выходного слоя использовалась функция Softmax.

Таблица 4. *Предполагаемое оптимальное количество нейронов в скрытых слоях*

| Предполагаемое оптимальное количество нейронов в скрытом слое 3-слойного персептрона | Предполагаемое оптимальное количество нейронов в скрытом слое 3-слойного персептрона | Предполагаемое оптимальное количество нейронов в скрытом слое 4-слойного персептрона |
|---|---|---|
| 24-94 | 34-84 | 15-65 |

Таблица 5. *Результаты обучения 3-слойного персептрона*

| Верность тестирования (%) | Верность проверки (%) | Количество нейронов на скрытом слое |
|---|---|---|
| 95,34 | 94,63 | 88 |
| 95,19 | 94,60 | 81 |
| 95,08 | 94,28 | 94 |
| 94,52 | 94,15 | 32 |
| 95,03 | 94,13 | 72 |

Таблица 6. *Результаты обучения 4-слойного персептрона*

| Верность тестирования (%) | Верность проверки (%) | Количество нейронов на первом скрытом слое | Количество нейронов на втором скрытом слое |
|---|---|---|---|
| 92,91 | 92,00 | 74 | 42 |
| 92,69 | 90,97 | 57 | 44 |
| 92,53 | 90,94 | 41 | 55 |
| 94,32 | 89,51 | 55 | 40 |
| 92,69 | 89,45 | 79 | 55 |

## 2.3. Оптимизация набора входных параметров

Для увеличения точности нейронных сетей и повышения скорости их обучения используется оптимизация входных параметров.

Для начала были определены линейнозависимые группы параметров.

Для этого была построена матрица корреляции и определены параметры, имеющие коэффициент корреляции больше 0,9.

Данные действия были выполнены с помощью библиотек pandas и sklearn для языка программирования Python.

Всего было получено 30 групп линейнозависимых параметров, из каждой группы было выбрано по одному параметру.

Затем при помощи библиотеки sklearn для языка Python была определена значимость каждого из входных параметров.

При анализе полученных результатов из обучающего набора были исключены 18 входных параметров, имеющих наименьшую значимость.

Они представлены в табл. 7.

Таблица 7. *Входные параметры наименьшей значимостью*

| Параметр | Значимость | Параметр | Значимость |
|---|---|---|---|
| Bwd PSH Flags | 0 | TotLen Bwd Pkts | 5,312 |
| Bwd URG Flags | 0 | Tot Bwd Pkts | 9,9507 |
| Fwd Byts/b Avg | 0 | Subflow Bwd Pkts | 9,9507 |
| Fwd Pkts/b Avg | 0 | Bwd Header Len | 1,3293 |
| Fwd Blk Rate Avg | 0 | Active Std | 4,7524 |
| Bwd Byts/b Avg | 0 | FIN Flag Cnt | 4,861 |
| Bwd Pkts/b Avg | 0 | Active Min | 5,9189 |
| Bwd Blk Rate Avg | 0 | Active Mean | 8,3826 |
| Subflow Bwd Byts | 5,3113 | Active Max | 9,4877 |

## 2.4. Проектирование структуры нейронной сети с сокращенным набором параметров

В результате оптимизации набора входных параметров, был получен набор, состоящий из 30 параметров о сетевых соединениях.

На основании формул (1), (2), (3) и (4) было пересчитано количество нейронов на скрытых слоях для 3- и 4-слойного персептрона с использованием сокращенного набора параметров.

Они представлены в табл. 8.

Далее, относительно каждого полученного значения были составлены интервалы для экспериментального подбора наиболее оптимального количества нейронов на скрытых слоях.

Они приведены в табл. 9.

В табл. 10 и табл. 11 представлены 5 лучших результатов обучения 3-слойного персептрона и 5 лучших результатов обучения 4-слойного персептрона соответственно.

В каждом случае в качестве функции ошибки использовалась функция CrossEntropyLoss, в качестве функции активации скрытого слоя использовалась функция ReLu, в качестве функции активации выходного слоя использовалась функция Softmax.

Таблица 8. *Число нейронов в скрытых слоях при использовании сокращенного*

*набора параметров, рассчитанное по правилу геометрической пирамиды*

| Количество нейронов во входном слое | Количество нейронов в выходном слое | Количество нейронов в скрытом слое 3-слойного персептрона | Количество нейронов в первом скрытом слое 3-слойного персептрона | Количество нейронов во втором скрытом слое 4-слойного персептрона |
|---|---|---|---|---|
| 30 | 15 | 21 | 24 | 19 |

Таблица 9. *Предполагаемое оптимальное количество нейронов в скрытых слоях при использовании сокращенного набора параметров*

| Предполагаемое оптимальное количество нейронов в скрытом слое 3-слойного персептрона | Предполагаемое оптимальное количество нейронов в скрытом слое 3-слойного персептрона | Предполагаемое оптимальное количество нейронов в скрытом слое 4-слойного персептрона |
|---|---|---|
| 10-50 | 12-48 | 7-43 |

Таблица 10. *Результаты обучения 3-слойного персептрона при использовании сокращенного набора параметров*

| Верность тестирования (%) | Верность проверки (%) | Количество нейронов на скрытом слое |
|---|---|---|
| 95,33 | 94,35 | 50 |
| 95,24 | 94,35 | 34 |
| 96,46 | 93,56 | 43 |
| 94,57 | 92,15 | 28 |
| 94,05 | 92,12 | 32 |

Таблица 11. *Результаты обучения 4-слойного персептрона при использовании сокращенного набора параметров*

| Верность тестирования (%) | Верность проверки (%) | Количество нейронов на первом скрытом слое | Количество нейронов на втором скрытом слое |
|---|---|---|---|
| 93,72 | 92,54 | 44 | 25 |
| 93,87 | 92,07 | 36 | 33 |
| 94,19 | 91,63 | 18 | 31 |
| 93,68 | 90,95 | 41 | 13 |
| 92,97 | 90,03 | 39 | 19 |

## 3. Проектирование сетевого анализатора

Сетевой анализатор необходим для получения данных обо всем трафике, проходящем через узел сети.

В качестве основы при разработке сетевого анализатора был выбран свободно распространяемый сетевой анализатор CICFlowmeter-V4.0 с открытым исходным кодом, разработанный Канадским институтом кибербезопасности [4].

Предыдущая версия данного анализатора была использована для получения набора данных о сетевых вторжениях CSE-CIC-IDS2018.

При помощи данного сетевого анализатора можно получить тот же набор параметров о сессиях, который содержится в наборе CSE-CIC-IDS2018 и при минимальной обработке передавать полученные данные в хранилище информации о сетевом трафике для последующей обработки нейросетевым модулем.

CIC Flowmeter-V4.0 написан на языке программирования Java.

## 4. Проектирование хранилища информации о сетевом трафике

Хранилище информации о сетевом трафике необходимо для хранения данных, поступающих от сетевого анализатора, которые необходимо проверить на предмет наличия сетевых атак.

Для хранения будет использоваться база данных, содержащая одну таблицу – traffic_data. Каждая запись в таблице будет состоять из уникального идентификатора записи, временной метки, параметров сессии, достаточных для анализа при помощи нейросетевого модуля, результат обработки записи нейросетевым модулем и результат проверки эксперта.

## 5. Реализация приложения

На этапе проектирования была получена нейросетевая модель однослойного персептрона с 88 нейронами на скрытом слое, обладающая наиболее высокой точностью обнаружения сетевых атак, представленных в наборе CSE-CIC-IDS2018.

Данная модель была выбрана в качестве основы для разработки нейросетевого модуля. Используемый инструмент для создания нейросетевых моделей PyTorch позволяет сохранить параметры полученной модели в zip архив и использовать обученную модель в дальнейшем.

Нейросетевой модуль был написан на языке программирования Python, модуль сетевого анализатора – на языке Java, в качестве СУБД для реализации хранилища данных была использована PostgreSQL. Все используемые при разработке приложения средства являются кроссплатформенными, что позволяет использовать приложение как на операционных системах семейства Linux, так и Windows.

Разработанное приложение предоставляет пользователю следующие функции:

- выбор сетевого интерфейса для захвата сетевого трафика;
- запуск/остановка захвата сетевого трафика;
- запуск/остановка обработки данных о сетевом трафике нейросетевым модулем;
- сохранение текущей модели нейронной сети в zip-архив с указанным именем в указанной директории;
- загрузка модели нейронной сети из zip-архива с указанным именем в указанной директории;
- дообучение текущей модели нейронной сети с указанием csv-файла с данными для обучения.

Тестирование приложения производилось на компьютере под управлением операционной системы Ubuntu. Приложение было запущено в качестве сервиса при помощи демона systemd и команды systemctl.

Была протестирована работа всех модулей приложения. Для этого был запущен захват трафика на тестовом компьютере. Весь входящий и исходящий трафик за выбранный промежуток времени был получен и записан в хранилище информации о трафике. После появления в хранилище сформированных данных, все они были обработаны нейросетевым модулем.

Кроме того, нейросетевой модуль разработанного приложения был дополнительно протестирован на множестве размером 30000 записей из набора CSE-CIC-IDS2018.

Данное множество не использовалось при обучении нейросетевого модуля.

Результаты тестирования представлены в табл. 12.

## Выводы

Как видно из табл. 12, нейросетевой модуль с высокой точностью определяет соединения, не содержащие атаку (тип Benign), и распознает 8 из 14 представленных видов атак (с вероятностью более 90 %).

Присутствует высокий процент ошибок при распознавании следующих видов атак: Brute Force-XSS, Brute Force-Web, SQL Injection, Infilteration, DoS attack-GoldenEye, FTP-Brute Force.

Таблица 12. *Результаты тестирования нейросетевого модуля*

| Вид атаки | Общее количество атак | Количество правильно распознанных атак | Количество неверно распознанных атак | Вероятность обнаружения (%) |
|---|---|---|---|---|
| Benign | 10866 | 10703 | 163 | 98,5 |
| Bot | 2146 | 2141 | 5 | 99,77 |
| DoS attack-Slow-HTTPTest | 1049 | 1040 | 9 | 99,14 |
| Dos attack-Hulk | 3118 | 3029 | 89 | 97,14 |
| Brute Force-XSS | 34 | 21 | 13 | 64,76 |
| Brute Force-Web | 92 | 37 | 55 | 40,22 |
| SQL Injection | 13 | 3 | 10 | 23,08 |
| DDoS attack-LOIC-HTTP | 3396 | 3119 | 277 | 91,84 |
| Infilteration | 1214 | 7 | 1207 | 0,58 |
| DoS attack-GoldenEye | 311 | 143 | 168 | 45,98 |
| Dos attack-Slowloris | 83 | 83 | 0 | 100 |
| FTP-Brute Force | 1450 | 515 | 935 | 35,52 |
| SSH-Brute Force | 1407 | 1403 | 4 | 99,72 |
| DDoS attack-LOIC-UDP | 260 | 258 | 2 | 99,23 |
| DDoS-attack-HOIC | 1456 | 4338 | 223 | 95,11 |

## Список литературы

1. *Ясницкий Л.Н.* Интеллектуальные системы: учебник. М.: Лаборатория знаний, 2016. 221 с.

2. *Гамаюнов Д.Ю.* Обнаружение компьютерных атак на основе анализа поведения сетевых объектов: дис… канд. физ.-мат. наук / Московский государственный университет имени М.В. Ломоносова, 2007.
3. *Zeeshan Ahmad, Andan Shahid Khan, Cheah Wai Shiang, Johari Abdullah, Farhan Ahmad.* Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Wiley Online Library. 2020. DOI: 10.1002/ett.4150.
4. *CSE-CIC-IDS2018* on AWS. // Canadian Institute for Cybersecurity. URL: https://www.unb.ca/ cic/datasets/ids-2018.html (дата обращения: 11.12.2020).
5. *KDD Cup 1999:* Computer network intrusion detection // KDD. URL: https://www.kdd.org/kdd-cup/view/kdd-cup-1999/Data (дата обращения: 10.12.2020).
6. *The UNSW-NB15* Dataset Desctiption // UNSW. URL: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/ (дата обращения: 10.11.2020).
7. *Ясницкий Л.Н.* Интеллектуальные информационные технологии и системы: учеб.-метод. пособие / Перм. ун-т. Пермь, 2007. 271 с.
8. *Grabusts P., Zorins A.* The Influence of Hidden Neurons Factor on Neural Network Training Quality Assurance // Proceedings of the 10th International Scientific and Practical Conference. Vol. III. 2015. Vol. 76. P. 81. doi: 10.17770/etr2015vol3.213.

# Development of software application for network traffic analysis and intrusion detection

**A. D. Ivanov, A. A. Kutishchev, E. Yu. Nikitina**
Perm State University; 15, Bukireva st., Perm, 614990, Russia
mr.ivlex@gmail.com; 8-963-870-08-33

This paper demonstrated the use of neural networks in the development of network intrusion detection systems, described the structure of the developed software application for network traffic analysis and network attacks detection, and presented the software application results.

**Keywords:** *information security; intrusion detection system; artificial neural networks; network traffic analyser.1*