Дубиков Д.Э. Анализ пользовательских запросов на наличие сетевой атаки с использованием технологий больших данных // Международный журнал информационных технологий и энергоэффективности. – 2025. – Т. 10 № 1(51) с. 65–70

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.623

# АНАЛИЗ ПОЛЬЗОВАТЕЛЬСКИХ ЗАПРОСОВ НА НАЛИЧИЕ СЕТЕВОЙ АТАКИ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ БОЛЬШИХ ДАННЫХ

**Дубиков Д.Э.**
*ФГАОУ ВО "МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ" , Москва, Россия (107023, город Москва, Большая Семёновская ул., д. 38 ), e-mail: Orp7ptdQtr@yandex.ru*

**В процессе функционирования сетей, в том числе Интернета, их узлы, принимающие входящие запросы, сталкиваются с проблемой различения в них полезных и вредоносных. Такие запросы могут приводить к несанкционированному доступу к узлу сети и, как следствие, к отказу систему, утечке конфиденциальной информации и иным нежелательным последствиям. В статье рассматривается ситуация, при которой через сеть поступает множество пользовательских запросов, часть из которых представляют собой сетевую атаку. Целью исследования является анализ пользовательских запросов на наличие сетевой атаки и автоматизация этого процесса с использованием технологий больших данных. Для анализа выбран датасет с данными о совершённых сетевых атаках. При помощи языка программирования Python и специальных библиотек pandas и sklearn реализована его автоматизация путём создания нейронной сети. Полученная нейронная сеть имеют высокую точность и может быть использована для анализа пользовательских запросов на практике в любой сфере человеческой деятельности, требующей работы с сетью Интернет. Созданный алгоритм может быть использован для обучения нейронной сети на любых других данных, имеющих свою специфику.**

Ключевые слова: Сетевая атака, сетевой запрос, большие данные, нейронная сеть.

# ANALYSIS OF USER REQUESTS FOR THE PRESENCE OF NETWORK ATTACK USING BIG DATA TECHNOLOGIES

**Dubikov D.E.**
*MOSCOW POLYTECHNIC UNIVERSITY, Moscow, Russia (107023, Moscow, Semyonovskaya str., 38.), e-mail: Orp7ptdQtr@yandex.ru*

**In the process of functioning of networks, including the Internet, their nodes that receive incoming requests face the problem of distinguishing between normal and malicious units. Such requests can lead to unauthorized access to a network node and, as a consequence, to system failure, leakage of confidential information and other undesirable consequences. The article considers a situation in which many user requests are received through the network, some of which represent a network attack. The purpose of the study is to analyze user requests for a network attack and automate this process using big data technologies. For the analysis, a dataset with data on committed network attacks was selected, using the Python programming language and special libraries pandas and sklearn, its automation was implemented by creating a neural network. The resulting neural network has high accuracy and can be used to analyze user requests in practice. The created algorithm can be used to train the neural network on other data with different specifics. The developed neural network can be applied in any area of human activity that requires working with the Internet.**

Keywords: Network attack, network request, big data, neural network.

## Введение

Ещё в недалёком прошлом технологии, позволяющие соединять технические устройства в единую сеть, были доступны только самым крупным компаниям, поскольку стоили больших

денег по причине слабой развитости соответствующих сфер знаний человечества и отсутствия продвинутых технологий серийного производства соответствующего оборудования. Однако в современном мире подавляющее большинство людей имеют множество сложных технических устройств, часто соединённых в единую всемирную сеть — Интернет, что обуславливает актуальность настоящего исследования. Если недавно к таким устройствам относились лишь компьютеры и смартфоны, то сегодня с развитием интернета вещей в эту категорию попадают и иная бытовая техника от холодильников и микроволновок до камер видеонаблюдения [4].

Стремительное развитие сетевых технологий неизбежно влечёт за собой необходимость развития соответствующих протоколов безопасности взаимодействия узлов сети между собой. Эта необходимость обусловлена, в первую очередь, сохранением конфиденциальности информации, которую хранят и которой оперируют технические устройства, но также нужда в ней обуславливается и обеспечением в целом правильной работоспособности сети, то есть её защиты от попыток дестабилизации нормальной работы [4].

Сетевые атаки являются серьёзным препятствием для штатной работы сети, поскольку имитируют обычные запросы, предназначенные для взаимодействия узлов в сети, имея своей реальной целью дестабилизацию работы системы. Различение обычных пользовательских запросов и запросов, представляющих собой элементы сетевой атаки, представляет собой важную часть концепции защиты сети, которая должна быть учтена в протоколах безопасности.

Таким образом, целью исследования является построение анализатора пользовательских запросов на наличие в них сетевой атаки с использованием технологий больших данных.

## Материалы и методы

### Материалы для анализа

Для анализа пользовательских запросов на наличие в них сетевой атаки используем набор данных, содержащий различную информацию о пользовательских запросах, а также имеющий поле, обозначающее, являлся ли фактически запрос элементом сетевой атаки.

Датасет содержит 257674 строки и 49 столбцов, из которых 1 обозначает наличие атаки, 1 — тип атаки, 1 — идентификатор записи в датасете и 46 — параметры запроса, предположительно являющегося сетевой атакой.

Исходные сетевые пакеты набора данных были созданы с помощью инструмента IXIA PerfectStorm в лаборатории Cyber Range Австралийского центра кибербезопасности (ACCS) для создания гибрида реальных современных повседневных действий и синтетических современных атак. В этом наборе данных есть девять типов атак: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode и Worms.

### Метод анализа

Для анализа данных пользовательских запросов на наличие в них сетевой атаки используем нейронную сеть. Этот мощный инструмент, получающий всё более широкое распространение за счёт распараллеливания обработки информации и способности к самостоятельному обучению, то есть созданию обобщений, где под «обобщением» понимается получение результата, основываясь на данных, не встречавшихся во время обучения [1].

Использование нейронных сетей обеспечивает следующие полезные свойства системы [1]:

1. Нелинейность. Является особенно важным свойством, когда механизм формирования входного сигнала также не считается линейным, что позволяет значительно расширить области применения нейросетевых технологий.
2. Отображение входной информации в выходную.
3. Адаптивность. Это свойство означает, что нейронные сети имеют способность адаптировать синаптические веса к происходящим при анализе изменениям. Так, например, нейронная сеть, обученная в определённой среде, может быть переобучена для новых условий.
4. Очевидность ответа. Нейронная сеть даёт однозначный ответ на запрос, взвешивая все вероятности по ходу своей работы.
5. Контекстная информация.
6. Отказоустойчивость. Неблагоприятные условия не оказывают значительного влияния на производительность системы. Так, например, если какой-либо нейрон оказывается повреждён, то извлечь усвоенную им информацию трудно, однако с учётом распределённого характера хранения информации можно утверждать, что существенного влияния на работу нейронной сети в целом это не оказывает.
7. Масштабируемость.

**Инструменты анализа**

Анализ данных пользовательских запросов на наличие в них сетевой атаки автоматизируем при помощи соответствующих инструментов. В качестве такого инструмента автоматизации выберем язык программирования Python, имеющий ряд преимуществ [5]:

1. Качество программного обеспечения. Python разработан специально для того, чтобы отличаться от остальных языков программирования читабельностью и согласованностью.
2. Продуктивность труда разработчиков. За счёт, в том числе, динамической типизации переменных Python значительно снижает трудоёмкость процесса разработки.
3. Переносимость программ. Код, написанный на Python, практически всегда работает одинаково на всех платформах компьютеров, в том числе при переносе на другую операционную систему.
4. Поддерживающие библиотеки. Так называемая «стандартная библиотека» Python включает в себя множество инструментов самого разного рода для решения наиболее часто требуемых для разработчиков задач, в том числе, узкоспециализированного профиля.
5. Интеграция компонентов. Python позволяет работать с библиотеками, написанными для других языков программирования, таких как C++. Это в значительной степени расширяет его возможности, в том числе, производительность.

В рамках использования Python используем специальные библиотеки для анализа данных. Для преобразования исходного датасета в переменную, пригодную для работы, используем встроенные инструменты библиотеки pandas, являющейся центром обширной экосистемы исследования данных. Преимуществом pandas является её универсальность по части сочетаемости с другими библиотеками, представляющими более широкие возможности для анализа данных [2]. Для собственно работы с датасетом выберем библиотеку sklearn. Scikit-Learn очень проста в использовании, но при этом эффективно реализует множество

алгоритмов машинного обучения, поэтому является отличной отправной точкой для работы с инструментами машинного обучения [3].

Для обучения нейронной сети используем 75% исходного датасета, для тестирования — оставшиеся 25%.

**Результаты**

Для заявленной автоматизации анализа пользовательских запросов на наличие в них сетевой атаки была написана программа на языке программирования Python с использованием библиотек для анализа данных pandas и sklearn. Код представлен в листинге 1.

Листинг 1 — Код программы, автоматизирующей анализ пользовательских запросов на наличие в них сетевой атаки

```
import pandas as pd
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier


df = pd.read_csv('UNSW_NB15_training-set.csv')

df = df.fillna('')

df = df.drop(['proto', 'service', 'state', 'attack_cat'], axis=1)

X = df.drop('label', axis=1)
y = df['label']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)
clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)
preds = clf.predict(X_test)

print(accuracy_score(y_test, preds))
```

Согласно коду, столбец «label» датасета использован в качестве выходного параметра, а остальные — в качестве входных параметров для нейронной сети, причём четыре столбца — «proto», «service», «state», «attack_cat» — были признаны несущественными и не учитывались при обучении.

Для тестовой выборки использовалась последняя четверть датасета, а для обучения нейронной сети — его первые три четверти. Библиотека pandas была использована для преобразования исходного датасета к виду, доступному для работы с библиотекой sklearn, при помощи которой, в свою очередь, с использованием представляющих её инструментов был проведён собственно анализ исходных данных.

После обучения и тестирования программа рассчитывает и выводит пользователю точность полученной нейронной сети: 0,9821015538893805, то есть приблизительно 98,21%.

Такой показатель означает, что нейронная сеть имеет высокую точность и способна давать в значительной степени достоверный результат при использовании, в том числе, в реальных практике. Такой высокий показатель обеспечен в значительной степени универсальностью использованных средств автоматизации анализа данных, не требующих от пользователя глубоких теоретических знаний, в том числе, по части выбранных специальных библиотек.

**Обсуждение**

Получена нейронная сеть, способная с высокой точностью анализировать пользовательские запросы на наличие в них сетевой атаки. Также разработан соответствующий алгоритм обучения и тестирования полученной нейронной сети, для которого могут быть использованы не только исходные для текущего исследования данные, но и иные, специфические для той или иной области деятельности или, более узко, конкретного предприятия.

Разработанная нейронная сеть может найти широкое применения в самых разных областях человеческой жизнедеятельности — на любых предприятиях и компаниях, использующих Интернет в качестве сети для обмена информацией между внутренними и внешними техническими устройствами. Созданная нейронная сеть способна распознавать сетевые атаки, направленные на дестабилизацию работы предприятия, и уведомлять об этом иные соответствующие системы, разработанные, например, для отсеивания входящего трафика, что, в свою очередь, обеспечивает защиту оборудования от несанкционированного доступа, перегрузки бесполезными запросами и иных способов негативного влияния на часть сети Интернет, локальную для защищаемого предприятия.

Область применения разработанной нейронной сети и соответствующего алгоритма для её обучения столь же широка, сколь широко использование больших объёмов данных с подключением соответствующих технических устройств к Интернету.

**Заключение**

Таким образом, по ходу проведения исследования получена нейронная сеть для анализа пользовательских запросов на наличие в них сетевой атаки. Точность созданной нейросети получилась высокой, что означает, что разработанный алгоритм её обучения является качественным, а полученная нейронная сеть может быть использована на практике. Алгоритм и нейронная сеть могут найти своё применение в любой сфере деятельности, где проводится работа с использованием технологий и устройств, задействующих для своего функционирования Интернет или иные сети.

**Список литературы**

1. Лутц М. Изучаем Python // Диалектика. 2019. С. 40–60.
2. Пасхавер Борис. Pandas в действии // Питер. 2023. С. 30–34.
3. Рашка С. Python и машинное обучение // ДМК Пресс. 2017. С. 68–73.
4. Форшоу Дж. Атака сетей на уровне протоколов // ДМК Пресс. 2021. С. 18–19.
5. Хайкин С. Нейронные сети: полный курс // Вильямс. 2006. С. 31–37.

**References**

1. Lutz M. Learning Python // Dialektika. 2019. P. 40–60.
2. Paskhaver B. Pandas in Action // Piter. 2023. P. 30–34.
3. Rashka S. Python Machine Learning // DMK Press. 2017. P. 68–73.
4. Forshow J. Attacking Network Protocols // DMK Press. 2021. P. 18–19.
5. Haykin S. Neural Networks: A Comprehensive Foundation // Williams. 2006. P. 31–37.