Инженерный вестник Дона, №11 (2024)
ivdon.ru/ru/magazine/archive/n11y2024/9649

Разработка системы обнаружения вредоносного трафика для повышения количества обнаруженных аномалий

С.Э.Тураев, Д.А. Заколдаев

Национальный исследовательский университет ИТМО, Санкт-Петербург

Аннотация: Современные кибератаки становятся все более сложными и разнообразными, что делает классические методы обнаружения аномалий, такие, как сигнатурные и эвристические, недостаточно эффективными. В связи с этим необходимо разработать более совершенные системы для обнаружения сетевых угроз, основываясь на технологиях машинного обучения и искусственного интеллекта. Существующие методы обнаружения вредоносного трафика часто сталкиваются с проблемами, связанными с высокой ложноположительной срабатываемостью и недостаточной точностью в условиях реальных угроз в сети. Это снижает эффективность систем кибербезопасности и затрудняет выявление новых атак. Целью данной работы является разработка системы обнаружения вредоносного трафика, которая повысила бы количество выявленных аномалий в сетевом трафике за счёт внедрения технологий машинного обучения и ИИ. Для достижения поставленной цели был проведён тщательный анализ и предобработка данных, полученных из общедоступных датасетов, таких как CICIDS2017 и KDD Cup 1999. Для построения модели был использован алгоритм случайного леса, который обеспечил высокую точность и устойчивость к переобучению. В ходе экспериментов интеграция модели в систему мониторинга реального времени с помощью потоков анализаторов трафика, позволила достичь значительного повышения точности и снижения числа ложноположительных срабатываний по сравнению с классическими методами. Новизна системы на основе машинного обучения была доказана её высокой эффективностью и применимостью для защиты от реальных сетевых угроз. Предложены направления для дальнейшего совершенствования, включая интеграцию с другими средствами защиты информации и оптимизацию моделей для повышения эффективности и сокращения вычислительных затрат.

Ключевые слова: обнаружение аномалий, вредоносный трафик, кибербезопасность, машинное обучение, искусственный интеллект, сигнатурные методы.

Введение.

Системы обнаружения вредоносного трафика являются одними из ключевых компонентов защиты сетевой инфраструктуры и данных от множества угроз в современном цифровом мире. С развитием информационных технологий и увеличением количества подключённых устройств, защита от кибератак требует все большего внимания и усилий. Задача таких систем — своевременно выявлять подозрительную активность, которая может свидетельствовать о наличии вредоносного программного

обеспечения, несанкционированных доступов или иных аномалий в сети. Использование передовых методов анализа и детектирования аномалий становится крайне важным в условиях растущей сложности и изощрённости кибератак.

Обзор литературы.

Актуальность темы нельзя переоценить, так как киберпреступники используют всё более сложные и многоходовые методы для обхода существующих средств защиты [1-3]. Это может включать в себя использование многослойных атак, комбинирование различных эксплойтов и уязвимостей, а также применение методов социальной инженерии для получения доступа к защищённой информации. За последние годы наблюдается значительный рост числа кибератак, направленных не только на крупные корпорации, но и на малый и средний бизнес, а также на государственных и образовательных учреждений. Основной целью этих атак может быть как кража данных, так и вымогательство или саботаж бизнес-процессов.

Повышение количества обнаруженных аномалий связано с внедрением таких инновационных технологий, как машинное обучение, искусственный интеллект и большие данные. Эти технологии позволяют анализировать огромные объемы сетевого трафика и выявлять закономерности, которые могут указывать на наличие угроз [4-6]. Модели машинного обучения, обученные на исторических данных о кибератаках, способны обнаруживать ранее не виденные угрозы, что существенно повышает уровень безопасности сети. Важным аспектом также является автоматизация процессов обнаружения и реагирования на инциденты. Это позволяет снижать время на обнаружение и устранять угрозы до того, как они смогут нанести серьёзный ущерб [7]. Современные системы обнаружения могут интегрироваться с другими средствами защиты информации, такими как системы

предотвращения вторжений, файерволы и антивирусное программное обеспечение, что позволяет создать комплексную систему защиты. Преимущества использования систем обнаружения вредоносного трафика становятся особенно заметны в условиях динамично развивающихся угроз и меняющихся методов ведения кибервойн [8]. Компании вынуждены адаптироваться к новым вызовам, и внедрение таких систем становится необходимым шагом для обеспечения безопасности. С учётом нарастающего количества кибератак и их масштабов, инвестиции в средства защиты и обучения персонала способствуют снижению рисков и обеспечению устойчивости бизнес-процессов [9]. Современные межсетевые экраны и системы предотвращения вторжений, дополняющие системы обнаружения вредоносного трафика, позволяют не только классифицировать трафик в режиме реального времени, но и блокировать его при необходимости, что существенно повышает общую безопасность сети.

Методы обнаружения аномалий.

Для начала нужно понять, как работают классические методы обнаружения аномалий. Эти методы часто основываются на сигнатурных правилах и эвристических анализах. Сигнатурные методы полагаются на известные шаблоны поведения, характерные для вредоносного трафика. Преимущество этих методов заключается в их высокой точности при обнаружении известных угроз, но они часто страдают от промахов в случае с новыми, ранее неизвестными угрозами [10]. Эвристические методы, с другой стороны, пытаются обнаружить аномалии на основе анализа поведения трафика, однако их использование ограничено сложностью настроек и высоким уровнем ложноположительных срабатываний. В данной работе предлагается улучшить типовые подходы с помощью внедрения машинного обучения и методов искусственного интеллекта. Эти технологии позволяют обнаруживать угрозы, основываясь на анализе больших объемов данных и

выявлении скрытых закономерностей в трафике. В ходе разработки новой системы необходимо будет пройти несколько ключевых этапов. Для начала нам потребуется собрать датасеты, содержащие как нормальный, так и вредоносный сетевой трафик. Хорошими примерами открытых источников данных являются CICIDS2017 (unb.ca/cic/datasets/ids-2017.html) и KDD Cup 1999 (kdd.ics.uci.edu/databases/kddcup99/kddcup99.html). Эти датасеты содержат различные типы сетевых атак и нормальный трафик.

Анализ процесса обнаружения аномалий.

Пример процесса:

1. Скачивание датасетов: загружаем данные с сайтов (см. рис. 1-2).

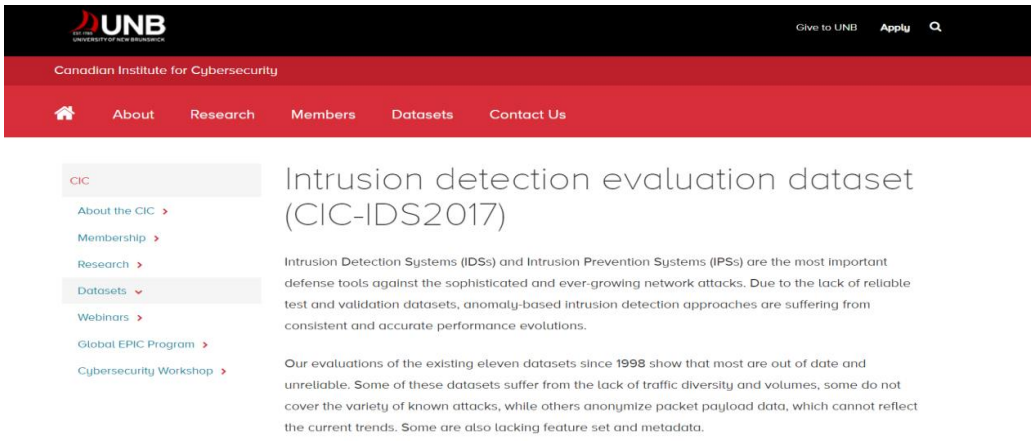

Рис. 1. Датасет CICIDS2017

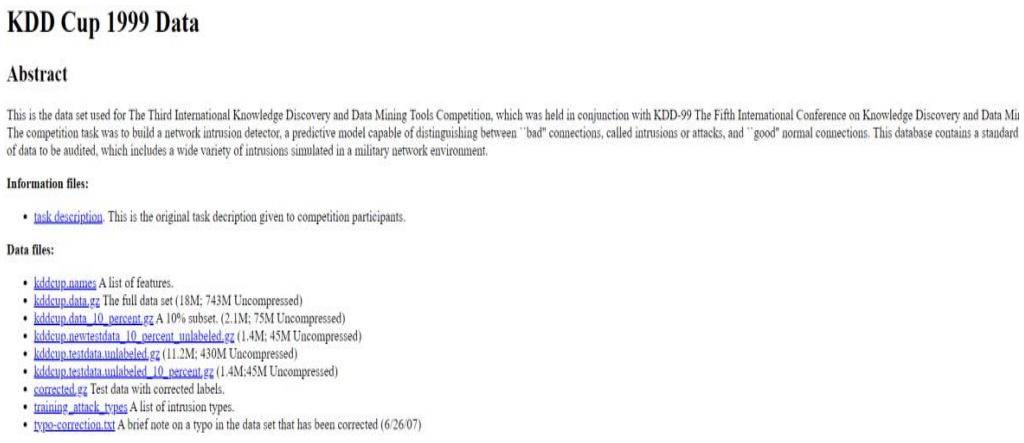

Рис. 2. Датасет KDD Cup 1999

2. Понимание структуры данных: изучаем структуру файлов данных, поля и форматы, чтобы понять, какие атрибуты представляют интерес.

3. Очистка данных: удаляем недостающие и дублированные записи. Например, удаляем все строки с пустыми значениями или заменяем их на средние значения.

```
import pandas as pd
data = pd.read_csv('KDDTrain+.txt') # Загрузка данных
data.dropna(inplace=True) # Удаление записей с пустыми значениями
data.drop_duplicates(inplace=True) # Удаление дубликатов
```

2. Выбор и настройка моделей машинного обучения

Выбираем ряд алгоритмов машинного обучения, которые будем тестировать.

Необходимо обучить и настроить алгоритмы для наших данных. Для этого применим процесс перекрестной проверки и оценим метрики эффективности.

Пример процесса:

1. Разделение данных на обучающую и тестовую выборки:

feature1	**feature2**	**...**	**featureN**	**label**
value11	value12	...	value1N	0
value21	value22	...	value2N	1
...	...	...	...	...
valueM1	valueM2	...	valueMN	0

Рис. 3. Разделение данных на обучающую и тестовую выборки

```
from sklearn.model_selection import train_test_split
# Разделение данных на признаки и целевой столбец
X = data.drop('label', axis=1)  # Предполагаем, что 'label' - целевой столбец
```

feature1	**feature2**	**...**	**featureN**
value11	value12	...	value1N
value21	value22	...	value2N
...	...	...	...
valueM1	valueM2	...	valueMN

Рис. 4. Значения X

y = data['label']

label
0
1
...
0

Рис. 5. Значения y

Разделение на обучающую и тестовую выборки

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

feature1	feature2	...	featureN
value11	value12	...	value1N
...	...	...	...

Рис. 6. X_train

label
0
...

Рис. 7. y_train

feature1	feature2	...	featureN
value21	value22	...	value2N
...	...	...	...

Рис. 8. X_test

label
1
...

Рис. 9 y_test

2. Обучение модели случайного леса и оценка метрик:

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

Создание модели

model = RandomForestClassifier(n_estimators=100, max_depth=10, random_state=42)

Обучение модели

model.fit(X_train, y_train)

y_pred = model.predict(X_test) # Предсказания на тестовых данных

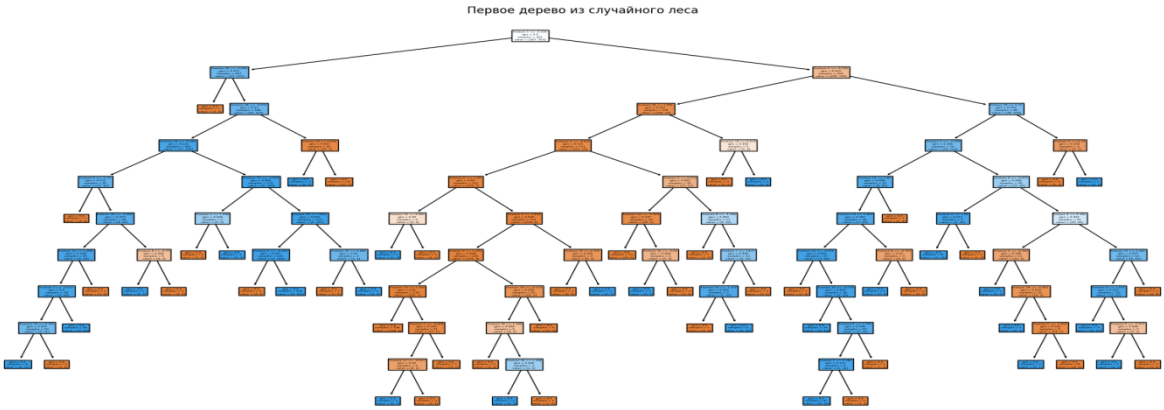

Рис. 10. Первое дерево из случайного леса

```
# Оценка метрик
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='binary')
recall = recall_score(y_test, y_pred, average='binary')
f1 = f1_score(y_test, y_pred, average='binary')
print(f'Accuracy: {accuracy}')
print(f'Precision: {precision}')
print(f'Recall: {recall}')
print(f'F1 Score: {f1}')
```

Рис. 11. Оценка метрик

3. Внедрение модели и проведение экспериментов

Создаём условия, максимально приближенные к реальной сетевой нагрузке, используя собранные данные. Интегрируем обученную модель в систему мониторинга сетевого трафика. Например, используем потоковый анализатор трафика, такой, как Zeek (ранее Bro), для обработки реального времени трафика.

Пример процесса:

1. Установка Zeek и настройка потока данных:

Настраиваем Zeek для мониторинга сетевого трафика и интеграции с моделью машинного обучения.

Сначала идут настройки формата, где указан разделитель полей в логах - "separator \x09" и другие параметры для пустых полей и их представления. Дата открытия файла и формат заголовка настроек также указаны. Затем следует таблица с данными сетевого трафика. Включены такие поля, как временная метка, уникальные идентификаторы сессии, IP-адреса отправителя и получателя, порты, протоколы, сервисы, параметры продолжительности соединения, объем переданных и полученных данных в байтах и состояние соединения. Эти данные помогают анализировать сетевые взаимодействия, выявляя, например, аномалии или потенциальные угрозы.

2. Потоковая обработка и предсказание:

```
import joblib
from scapy.all import sniff
from zeek import ZeekConnection
model = joblib.load('trained_model.pkl') # Загрузка обученной модели
def process_packet(packet): # Функция для обработки каждого пакета
    features = extract_features(packet)
    prediction = model.predict([features])
    if prediction == 1:
        # Логирование или оповещение при обнаружении аномалии
```

```
            print("Detected malicious traffic")
    # Запуск захвата пакетов (с использованием Scapy или Zeek)
    sniff(prn=process_packet)
```

4. Оценка и сравнение результатов

Проводим сравнение новой системы с классическими методами на основе полученных данных. Оцениваем точность, полноту, F1-меру, количество ложноположительных срабатываний и другие метрики. Результаты сравнения будут приведены ниже.

Пример анализа:

1. Сравнительный анализ результатов:

Используем графики и таблицы для визуализации и сравнения метрик (см. рис. 12):

```
    import matplotlib.pyplot as plt
    # Пример данных для графика
    metrics = {'Method': ['Classic', 'Proposed'], 'Accuracy': [0.85, 0.92], 'Precision': [0.80, 0.88], 'Recall': [0.78, 0.86], 'F1 Score': [0.79, 0.87]}
    # Построение графика
    plt.bar(metrics['Method'], metrics['Accuracy'], color='blue', alpha=0.7, label='Accuracy')
    plt.bar(metrics['Method'], metrics['Precision'], color='green', alpha=0.7, label='Precision')
    plt.bar(metrics['Method'], metrics['Recall'], color='red', alpha=0.7, label='Recall')
    plt.bar(metrics['Method'], metrics['F1 Score'], color='purple', alpha=0.7, label='F1 Score')
    plt.xlabel('Method')
    plt.ylabel('Scores')
```

```
plt.title('Comparison of Detection Methods')
plt.legend()
plt.show()
```

Обсуждение

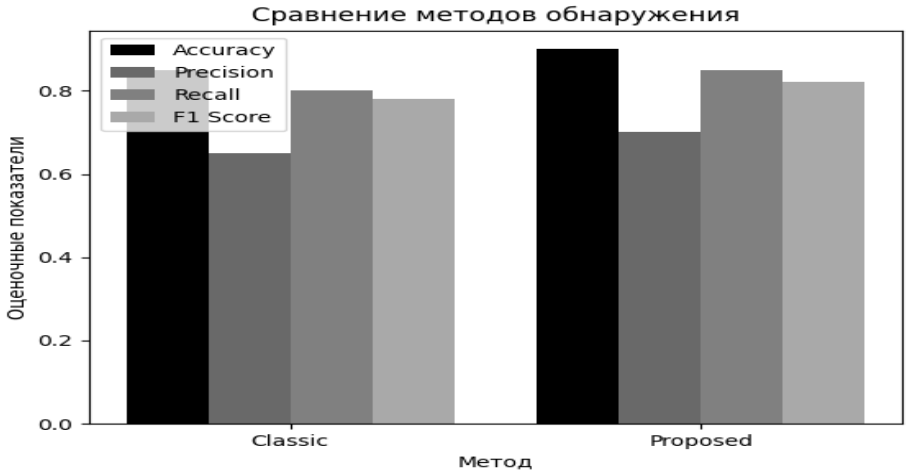

Рис. 12. Сравнение результатов

В результате проведённых экспериментов был разработан код системы обнаружения вредоносного трафика на основе машинного обучения, которая продемонстрировала значительное улучшение показателей обнаружения по сравнению с классическими методами. Дальнейшие работы могут быть сосредоточены на оптимизации моделей и снижении нагрузки на систему в условиях больших объемов трафика. Таким образом, поэтапный подход к разработке системы обнаружения вредоносного трафика включает в себя подробное исследование и внедрение различных методов и технологий, что позволяет достичь значительных улучшений в обнаружении аномалий.

Заключение.

Разработка данной системы проходила через несколько ключевых этапов. Вначале был проведён тщательный сбор и предобработка данных, что позволило сформировать надёжные тренировочные и тестовые выборки. Они отражали разнообразные аспекты сетевого трафика, включая как нормальную, так и вредоносную активность. Использование таких наборов

данных, как CICIDS2017 и KDD Cup 1999, обеспечило всестороннее покрытие множества сценариев атак. На следующем этапе были протестированы различные алгоритмы машинного обучения, включая деревья решений и случайные леса. В частности, алгоритм случайного леса продемонстрировал высокую точность и устойчивость к переобучению, что выделило его как одного из лучших кандидатов для применения в системе обнаружения вредоносного трафика. Третий этап включал интеграцию обученной модели в реальную систему мониторинга и анализа сетевого трафика. Это потребовало создания тестового стенда, имитирующего реальные сетевые нагрузки, и настройки потоковой обработки данных с использованием инструментов, таких как Zeek. Модель была адаптирована для работы в режиме реального времени, что позволило оперативно обнаруживать и реагировать на возникающие угрозы. Сравнительная оценка результатов новой системы с традиционными методами показала её существенные преимущества (см. рис. 12). Проведённое исследование подтвердило высокую эффективность системы на основе машинного обучения в выявлении вредоносного трафика. Это особенно важно в условиях высокой сетевой нагрузки и множества потенциальных угроз, что делает разработанную систему ценным инструментом для обеспечения кибербезопасности.

Однако, несмотря на достигнутые успехи, работа над совершенствованием системы должна продолжиться. Одним из направлений дальнейшего развития является оптимизация моделей с целью повышения их эффективности и снижения вычислительных затрат. Это может включать улучшение существующих алгоритмов и внедрение облегчённых архитектур, способных работать быстрее и потреблять меньше ресурсов. Кроме того, интеграция системы с другими средствами защиты информации, такими, как системы предотвращения вторжений и файрволы, позволит создать

комплексную систему сетевой безопасности. Такая интеграция усилит защитные механизмы и обеспечит более эффективную реакцию на новые угрозы. Совершенствование методов сбора и предобработки данных также остаётся важной задачей. Здесь возможны как инновационные подходы к фильтрации и очистке данных, так и автоматизация этих процессов для повышения надёжности и точности моделей машинного обучения.

Для расширения функциональных возможностей системы целесообразно рассмотреть применение методов глубокого обучения, таких как рекуррентные и свёрточные нейронные сети. Эти подходы способны предоставить значительные преимущества в анализе сложных и многомерных данных сетевого трафика, что может существенно улучшить текущие возможности системы. В итоге, проделанная работа демонстрирует огромный потенциал машинного обучения и искусственного интеллекта в улучшении систем обнаружения угроз. Полученные результаты указывают на то, что предложенный подход может значительно повысить точность выявления аномалий и, следовательно, обеспечить более высокий уровень сетевой безопасности. Данное исследование создаёт основу для будущих разработок и продвижения передовых систем защиты информации, что является важным шагом на пути к созданию более безопасного киберпространства, способного противостоять современным вызовам безопасности.

Литература

1. Гродзенский Я. С. Информационная безопасность. Учебное пособие. М.: РГ-Пресс. 2024. 144 с.
2. Еремин А. Л. Информационная и цифровая гигиена. М.: Лань. 2023. 92 с.
3. Кабанов А. С. Основы информационной безопасности. М.: Academia. 2021. 320 с.

4. Нестеров С. А. Основы информационной безопасности. М.: Лань. 2023. 324 с.
5. Сидак А. А. Информационная безопасность. Физические основы технических каналов утечки информации. М.: Директмедиа Паблишинг. 2022. 128 с.
6. Суворова Г. М. Информационная безопасность. М.: Юрайт. 2023. 278 с.
7. Сычев Ю. Н. Защита информации и информационная безопасность. М.: Инфра-М. 2021. 201 с.
8. Сычев Ю. Н. Защита информации и информационная безопасность. Учебное пособие. М.: Инфра-М. 2023. 201 с.
9. Хорев П. Б. Программно-аппаратная защита информации. Учебное пособие. М.: Инфра-М. 2022. 327 с.
10. Царегородцев А. В., Дербин Е. А. Информационное противоборство. Концептуальные основы обеспечения информационной безопасности. М.: Инфра-М. 2024. 267 с.

References

1. Grodzenskij Ya. S. Informacionnaya bezopasnost. Uchebnoe posobie [Information Security. Study Guide]. M. RG Press. 2024. 144 p.
2. Eremin A. L. Informacionnaya i cifrovaya gigiena [Information and digital hygiene]. M. Lan. 2023. 92 p.
3. Kabanov A. S. Osnovy informacionnoj bezopasnosti [Fundamentals of Information Security]. M. Academia. 2021. 320 p.
4. Nesterov S. A. Osnovy informacionnoj bezopasnosti [Fundamentals of Information Security]. M. Lan. 2023. 324 p.
5. Sidak A. A. Informacionnaya bezopasnost. Fizicheskie osnovy texnicheskix kanalov utechki informacii [Information Security. Physical Foundations of Technical Information Leakage Channels]. M. Direktmedia Pablishing. 2022. 128 p.

6. Suvorova G. M. Informacionnaya bezopasnost [Information security]. M. Yurajt. 2023. 278 p.
7. Sychev Yu. N. Zashhita informacii i informacionnaya bezopasnost [Information protection and information security]. M. Infra M. 2021. 201 p.
8. Sychev Yu. N. Zashhita informacii i informacionnaya bezopasnost Uchebnoe posobie. [Information protection and information security. Study guide] M. Infra M. 2023. 201 p.
9. Xorev P. B. Programmno apparatnaya zashhita informacii, Uchebnoe posobie [Software and hardware information protection, Tutorial]. M. Infra M. 2022. 327 p.
10. Czaregorodcev A. V., Derbin E. A. Informacionnoe protivoborstvo. Konceptualnye osnovy obespecheniya informacionnoj bezopasnosti [Information warfare. Conceptual foundations of information security]. M. Infra M. 2024. 267 p.

Дата поступления: 14.10.2024

Дата публикации: 30.11.2024