Возможности и ограничения классификации атак в зашифрованном трафике методами машинного обучения

М.В. Бондарев, О.Ю. Гузев

***Аннотация*—В работе исследуются возможности и ограничения применения методов машинного обучения для классификации компьютерных атак в зашифрованном сетевом трафике, что актуально из-за широкого распространения шифрования, усложняющего работу традиционных методов анализа. В статье предложена комплексная модель данных, учитывающая статистические характеристики потоков, последовательности пакетов, а также распределение байт полезной нагрузки.**

Проведен сравнительный анализ алгоритмов машинного обучения: Random Forest, Extra Trees, Decision Tree, AdaBoost и KNN на трех публичных наборах данных: CIC-IDS2017, TON IoT и USTC-TFC2016. Наилучшие результаты показали алгоритмы Random Forest и Extra Trees, при этом Extra Trees продемонстрировал немного более высокую стабильность. Была проведена оптимизация гиперпараметров модели Extra Trees и доказана его эффективность при применении в рамках одного набора данных.

Ключевым ограничением предложенного подхода является низкая обобщающая способность модели – перекрестное тестирование между различными наборами данных выявило существенное снижение качества классификации, что указывает на зависимость результатов от конкретной сетевой среды. Примененные методы повышения обобщающей способности, включающие фильтрацию признаков, использование нейронной сети и доменную адаптацию, оказались недостаточно эффективными для создания универсального решения.

Сделан вывод о применимости моделей машинного обучения для качественного обнаружения и классификации компьютерных атак в пределах одной сетевой среды. Переносимость обученных моделей между различными сетевыми средами остается ограниченной и требует дальнейших исследований.

***Ключевые слова*—Анализ сетевого трафика, классификация атак, машинное обучение, обобщающая способность, зашифрованный трафик**

I. Введение

Согласно [1] по состоянию на август 2024 года более 95% интернет-трафика защищено с помощью протоколов TLS/SSL. Внедрение шифрования, безусловно, является важным шагом в задаче обеспечения конфиденциальности и безопасности информации. Однако, данный тренд создает значительную проблему для традиционных систем обнаружения вторжений (IDS, Intrusion Detection System).

В условиях, когда существенная доля сетевой активности скрыта шифрованием, эффективность классических методов обнаружения, основанных на сигнатурном анализе, заметно снижается. Принципиальная невозможность инспектирования содержимого зашифрованных пакетов без расшифрования ограничивает возможности сигнатурных IDS в части сопоставления с известными вредоносными шаблонами. Кроме того, сигнатурные методы традиционно характеризуются высоким уровнем ложноположительных срабатываний и ограниченной способностью к обнаружению атак "нулевого дня". В условиях преобладания зашифрованного трафика эти недостатки становятся еще более значимыми.

Таким образом, актуальной является задача разработки новых подходов к обнаружению вредоносной сетевой активности, способных эффективно функционировать в условиях повсеместного шифрования, и использование для этого методов машинного обучения и анализа паттернов сетевого трафика представляется перспективным направлением.

В рамках настоящего исследования была предпринята попытка создания модели машинного обучения для обнаружения и классификации компьютерных атак в зашифрованном трафике. В статье подробно рассматриваются ограничения данного подхода и предлагаются возможные направления для дальнейших исследований.

II. Постановка задачи и анализ релевантных работ

В данном разделе приведен сравнительный анализ работ, относящихся к теме обнаружения компьютерных атак в сетевом трафике.

В работе [2] рассматривалась задача классификации различных семейств вредоносных программ, использующих TLS-шифрование. Для ее решения использовалась модель логистической регрессии с L1-регуляризацией. Авторы предложили признаковое пространство, основанное на статистической информации о потоках пакетов, последовательности длин и времени прибытия пакетов, распределении байт,

Статья получена 7 мая 2025.

М.В. Бондарев, исследователь АО «ИнфоТеКС», студент НИ ТГУ, Томск, Россия (e-mail: Matvey.Bondarev@infotecs.ru).

О.Ю. Гузев, старший исследователь АО «ИнфоТеКС», к.т.н. Москва, Россия (e-mail: Oleg.Guzev@infotecs.ru).

а также незашифрованной информации из TLS-заголовков. Оценка эффективности подхода производилась на одном самостоятельно созданном наборе данных, разделенном на обучающую и тестовую выборки. В задаче бинарной классификации с использованием полного признакового пространства точность составила 99,6%. Модель также показала высокую точность 90,3% в задаче многоклассовой классификации 18 семейств вредоносных программ.

В работе [3], посвященной обнаружению сетевых вторжений с использованием алгоритма XGBoost, была проведена оценка его эффективности в выявлении компьютерных атак на основе анализа сетевого трафика. Для этого использовались специально разработанные наборы данных NSL-KDD и UNSW NB15. Результаты экспериментов продемонстрировали высокую точность обнаружения: 88,6% для набора данных NSL-KDD и 93,3% для UNSW NB15. В исследовании не рассматривался сценарий перекрестной валидации между двумя наборами данных, при котором модель обучается на одном наборе, а тестируется на другом.

В работе [4] описан метод классификации потоков трафика на основе анализа их временных рядов (SFTS, Single Flow Time Series). Авторами предложено признаковое пространство, включающее статистические характеристики (такие как: среднее, дисперсия, медиана) длин и времени прибытия пакетов, параметры распределения данных во временном ряду, частотные характеристики (например, спектральные компоненты) и поведенческие признаки, отражающие динамику трафика. Оценка эффективности подхода была выполнена на 15 общедоступных наборах данных, содержащих “сырой” сетевой трафик. В качестве основной модели для проведения тестирования был выбран алгоритм XGBoost. Результаты показали, что предложенный подход демонстрирует высокую точность, как в бинарной, так и в многоклассовой задачах классификации в рамках любого одного набора данных.

Помимо применения классических алгоритмов машинного обучения в последнее время наблюдается рост интереса к подходам на основе нейронных сетей. Например, в работе [5] был проведен анализ методов глубокого обучения для решения задачи обнаружения вторжений в зашифрованном трафике. Для формирования признакового пространства модели использовался инструмент CICFlowMeter. В ходе исследования рассматривались три различные архитектуры нейронных сетей: MLP (Multi-Layer Perceptron Classifier), 1-D CNN (1-Dimensional Convolutional Neural Network) и LSTM (Long-Short-Term-Memory). Оценка эффективности проводилась на наборе данных CTU-13 в задаче бинарной классификации. Результаты показали, что MLP обеспечивает высокую точность с показателями Accuracy и F1-мера 99,1% и 99,09% соответственно, а также демонстрирует минимальный уровень ложноположительных срабатываний (FPR, False Positive Rate).

В работе [6] предложен метод, основанный на глубокой сети доверия (DBN, Deep Belief Network). Данная архитектура представляет собой композицию нескольких ограниченных машин Больцмана (RBM, Restricted Boltzmann Machine). Процесс обучения DBN делится на два этапа: на первом этапе каждый слой RBM обучается неконтролируемо для извлечения абстрактных признаков из данных, на втором этапе к сети добавляется линейный слой, и вся архитектура настраивается к решаемой задаче. Для оценки эффективности авторами использовался набор данных CIC-IDS2017 в задаче многоклассовой классификации. Для уменьшения размерности пространства признаков использовался метод главных компонент (PCA, Principal Component Analysis). Предложенная архитектура продемонстрировала высокое значение F1-меры 94%, что превосходит аналогичный показатель классического MLP 87,3%. В своей работе авторы также провели тестирование различных комбинаций методов балансировки классов и продемонстрировали, что наилучший результат достигается при совместном использовании SMOTE (Synthetic Minority Over-sampling Technique) и RandomUnderSampling.

Рассмотренные выше и многие другие исследования подробно описывают использование методов машинного обучения для выявления компьютерных атак в сетевом трафике и анализируют эффективность их обнаружения. Однако, они не предусматривают проверку обобщающей способности разработанных методов с учетом специфики различных сетевых инфраструктур, что затрудняет оценку применимости методов в реальных условиях. Оценки точности в рассмотренных выше работах получены на отдельных наборах данных, разделенных на обучающую и тестовую выборки каждый, что не позволяет учитывать возможные различия в сетевых средах. В результате такие оценки могут быть завышенными и не отражать реальную эффективность моделей при их применении в условиях нескольких гетерогенных сетей.

Целью настоящего исследования является разработка метода обнаружения и классификации компьютерных атак в зашифрованном трафике с применением методов машинного обучения, а также проведение перекрестного тестирования между различными наборами данных для оценки применимости и универсальности метода в реальных сетевых инфраструктурах.

III. Обработка трафика и формирование модели данных

Для создания оцифрованных выборок был разработан комплексный программный инструмент, обеспечивающий обработку сетевого трафика в режиме реального времени, а также сохраненного в формате PCAP. Данный инструмент анализирует сетевой трафик, группируя его в двунаправленные потоки пакетов, где одно направление потока определяется как последовательность пакетов, имеющих одинаковые значения пяти параметров: IP-адрес источника, IP-адрес назначения, порт источника, порт назначения, транспортный протокол. В пакетах другого направления

потока парные параметры меняются местами. После группировки инструмент извлекает из потока необходимые данные. Ниже рассмотрим состав модели данных.

A. Статистические атрибуты потока

Первый блок модели данных представляет собой набор статистических атрибутов (признаков), полученных в результате анализа потока сетевых пакетов. В его основе лежит набор признаков инструмента CICFlowMeter [7], широко используемого в исследованиях для извлечения характеристик сетевого трафика. Стандартный набор CICFlowMeter был дополнен статистическими атрибутами, описывающими последовательности данных, и анализом времени отклика между отправленным и ответным пакетом. Данный блок содержит 114 атрибутов.

B. Последовательности длин и времени прибытия пакетов

Второй блок модели данных основан на представлении потока пакетов в виде Марковской цепи с дискретным временем. Важно отметить, что в данном контексте матрица переходов не является стохастической в классическом смысле, так как сумма элементов в строке может быть равна 0 (при отсутствии переходов из данного состояния) или 1 (при наличии переходов). Для построения набора анализируются первые 50 пакетов потока, при этом пакеты с нулевым размером полезной нагрузки исключаются из рассмотрения.

Подход к формированию признаков рассмотрим на примере последовательности длин пакетов. Матрица размерностью 10x10 отражает вероятности переходов между состояниями, дискретизированными на основе размера пакетов. Выбор шага дискретизации в 150 байт обусловлен стандартом Ethernet II, где максимальный размер пакета ограничен 1500 байтами. Для определения индексов состояний, соответствующих текущему и предыдущему пакету, размеры пакетов делятся на шаг дискретизации и округляются вниз до целого значения. При получении пакета происходит инкремент соответствующего элемента матрицы переходов, отражающего переход из состояния, определяемого предыдущим пакетом, в состояние, определяемое текущим пакетом.

Аналогичный алгоритм применяется для построения матрицы переходов для временных интервалов между пакетами. В данном случае используется шаг дискретизации в 50 миллисекунд. Состояние для первого пакета потока устанавливается равным 0.

Полученные матрицы нормализуются по строкам, формируя матрицу вероятностей переходов между состояниями. С целью учета направлений сетевого трафика (прямого и обратного) и типов анализируемых характеристик (длины пакетов и межпакетные временные интервалы) используются в общей сложности четыре матрицы переходов, что составляет 400 атрибутов. Визуализация графов переходов представлена на рисунке 1.

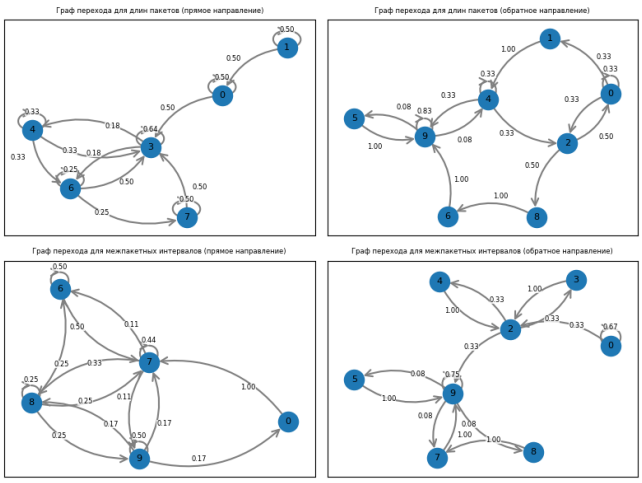

Рис. 1. Визуализация графов переходов для прямого и обратного направлений трафика.

C. Распределение байт

Распределение байт представляет собой вектор размером 256, каждый элемент которого отражает частоту встречаемости соответствующего байта (от 0 до 255) в полезной нагрузке пакетов потока. Вероятность каждого байта вычисляется, как отношение его частоты встречаемости к общему числу байт в полезной нагрузке пакетов. Пример визуализации распределения байт представлен на рисунке 2.

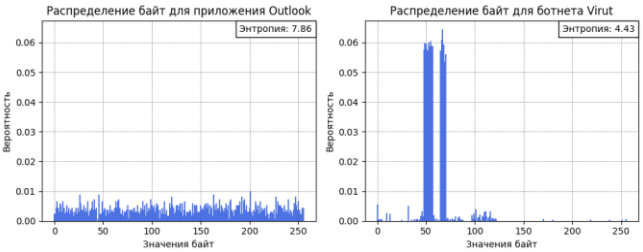

Рис. 2. Визуализация векторов распределения байт для приложения Outlook и ботнета Virut.

Анализ распределения байт может предоставить информацию о способе шифрования, а также выявить наличие заполнений, если в распределении наблюдаются аномальные пики.

Также следует заметить, что в модель данных можно добавить четвертый блок – атрибуты TLS-заголовков. Однако, поскольку в настоящем исследовании использовались наборы данных с различными протоколами защиты данных, с целью унификации модели данных для всех наборов атрибуты TLS-заголовков не использовались.

IV. Наборы данных

Для обучения и тестирования модели были использованы три публично доступных набора данных:

CIC-IDS2017 [8], TON IoT [9] и USTC-TFC16 [10]. Каждый из наборов был оцифрован с помощью разработанного инструмента и размечен в соответствии с доступным описанием.

A. CIC-IDS2017

Набор данных CIC-IDS2017, созданный Канадским институтом кибербезопасности в 2017 году,

представляет собой имитацию сетевого трафика небольшой корпоративной сети на протяжении пяти дней. Основной целью создания данного набора являлась реалистичная имитация, как легитимной пользовательской активности, так и разнообразных сетевых атак. В соответствии с целями исследования

разметка набора была уточнена. В частности, атака типа Infiltration, включающая элементы управления и контроля (C2) и разведки (Reconnaissance), была разделена на две отдельные категории. Атака типа Botnet была перенесена в категорию C2. Атака Heartbleed была исключена из рассмотрения. Помимо упомянутых выше изменений схожие по своей природе атаки были объединены в более общие категории. Количественный состав измененного набора данных представлен в таблице 1.

Табл. 1. Количественный состав измененного набора данных CIC-IDS2017.

Тип трафика	Количество потоков
Benign (легитимный трафик)	1530584
Reconnaissance	229453
DoS/DDoS	212787
Bruteforce	8359
C2	2252
XSS	683
SQL Injection	12

B. TON IoT

Набор данных TON IoT, разработанный Университетом Нового Южного Уэльса (UNSW Canberra) в 2021 году, представляет собой разнородные данные, содержащие, как легитимную, так и вредоносную сетевую активность. Основной целью создания TON IoT являлась разработка и тестирование методов машинного обучения для обнаружения и предотвращения кибератак в современных компьютерных и IoT-сетях. В рамках данного исследования использовалась подвыборка TON IoT, включающая только сетевой трафик. Количественный состав набора представлен в таблице 2.

Табл. 2. Количественный состав подвыборки набора данных TON IoT.

Тип трафика	Количество потоков
Reconnaissance	5796997
DoS/DDoS	4944508
XSS	2189843
Bruteforce	1539031
SQL Injection	441053
Benign (легитимный трафик)	81009
Backdoor	27105
Ransomware	2719
MITM	1041

C. USTC-TFC2016

Набор данных USTC-TFC2016 состоит из двух частей. Первая часть включает 10 типов трафика от вредоносных приложений, собранного исследователями из CTU в период с 2011 по 2016 год. Вторая часть включает 10 типов легитимного трафика, сгенерированного с помощью инструмента IXIA BPS. Количественный состав набора представлен в таблице 3.

Табл. 3. Количественный состав набора данных USTC-TFC2016.

Тип легитимного трафика	Количество потоков	Тип ботнета	Количество потоков
FTP	101037	Cridex	60051
MySQL	86089	Geodo	42808
Weibo	39950	Neris	34846
SMB	38937	Virut	33298
Gmail	8629	Miuref	13551
WorldOfWarcraft	7883	Nsis	7546
Outlook	7524	Htbot	6514
BitTorrent	7517	Zeus	6207
Skype	6321	Shifu	492
Facetime	6000	Tinba	10

Протоколы защиты данных в разных наборах различались: CIC-IDS2017 – SSH, TLS; TON IoT, USTC-TFC2016 – TLS.

V. Выбор модели машинного обучения, оптимизация гиперпараметров, тестирование

В данном разделе представлен процесс выбора и тестирования модели машинного обучения, включая оптимизацию гиперпараметров и оценку обобщающей способности с использованием метода перекрестной проверки между двумя наборами данных.

A. Выбор модели машинного обучения

Для выбора алгоритма классификации был проведен сравнительный анализ нескольких популярных моделей машинного обучения на каждом из используемых наборов данных:

1. Случайный лес (RF, Random Forest).
2. Экстремально случайные деревья (ET, Extra Trees).
3. Решающее дерево (DT, Decision Tree).
4. Адаптивный бустинг над решающими деревьями (AdaBoost).
5. Метод k ближайших соседей (KNN, k-Nearest Neighbors).

Оценка производилась методом стратифицированной кросс-валидации с пятью разбиениями. На каждом этапе применялась предварительная обработка данных с использованием Robust Scaler для нормализации признаков. Для уменьшения дисбаланса классов в обучающей выборке использовалась комбинация методов Random Under Sampling и SMOTE (Synthetic Minority Over-sampling Technique). Оценка качества классификации моделей производилась с использованием метрик Precision, Recall, F1-меры (макро-усреднение) и Accuracy.

Полученные результаты, усредненные по всем разбиениям для каждого набора данных, представлены в таблице 4.

Табл. 4. Результаты стратифицированной кросс-валидации на каждом наборе данных (для F1-меры, Precision и Recall используется макро-усреднение).

Набор данных	Алгоритм	Accuracy	Precision	Recall	F1-мера
CIC-IDS2017	RF	99,38%	87,79%	92,15%	88,41%
	ET	99,36%	87,82%	95,40%	90,27%
	DT	98,35%	76,98%	87,61%	80,97%
	AdaBoost	95,28%	58,05%	82,52%	62,31%
	KNN	96,02%	68,10%	87,56%	71,07%
TON IoT	RF	99,75%	94,04%	97,72%	95,54%
	ET	99,74%	94,07%	97,55%	95,50%
	DT	99,71%	93,06%	95,94%	94,28%
	AdaBoost	87,89%	57,55%	65,02%	55,84%
	KNN	86,42%	72,79%	89,05%	75,08%
USTC-TFC2016	RF	97,69%	97,38%	96,28%	96,73%
	ET	97,57%	97,18%	96,56%	96,85%
	DT	97,27%	95,94%	96,51%	96,20%
	AdaBoost	41,56%	17,39%	17,44%	13,21%
	KNN	88,3%	82,73%	84,36%	82,47%

По результатам проведенного тестирования наилучшую точность классификации продемонстрировали алгоритмы RF и ET. В частности, алгоритм ET показал более стабильные и высокие значения F1-меры на двух наборах данных, поэтому он был выбран в качестве основного алгоритма для дальнейших экспериментов.

B. Оптимизация гиперпараметров модели

Для оптимизации гиперпараметров модели ET использовался пакет Optuna [11], представляющий собой инструмент для автоматизированного подбора гиперпараметров на основе байесовской оптимизации.

Оптимизация проводилась для следующих гиперпараметров модели ET:

- количество деревьев (n_estimators);
- максимальное количество признаков (max_features);
- максимальная глубина деревьев (max_depth);
- минимальное количество образцов для разбиения (min_samples_split);
- минимальное количество объектов в листе (min_samples_leaf).

Для каждого набора данных выполнялась оптимизация гиперпараметров за 50 итераций. Перед оптимизацией данные были разделены на обучающую (70%), валидационную (10%) и тестовую (20%) выборки с сохранением исходного соотношения классов.

Обучающая выборка была сбалансирована с помощью комбинации методов Random Under Sampling и SMOTE, в качестве метрики оптимизации использовалась F1-мера с макро-усреднением. Оценки качества классификации потоков тестовых выборок с помощью настроенных и обученных моделей представлены в таблице 5.

Оптимизация гиперпараметров с использованием инструмента Optuna продемонстрировала улучшение показателей качества классификации на каждом из используемых наборов данных (табл. 5) по сравнению с использованием стандартного набора гиперпараметров (табл. 4).

C. Оценка обобщающей способности модели

Высокие показатели качества классификации, полученные при обучении и тестировании модели в рамках одного набора данных, соответствуют результатам, представленным в работах [2–6]. Однако, для всесторонней оценки обобщающей способности модели требуется проведение перекрестного тестирования, при котором обучение осуществляется на одном наборе данных, а тестирование – на другом независимом наборе, что позволяет оценить устойчивость модели к изменениям сетевой среды.

Для проведения перекрестного тестирования использовались наборы данных CIC-IDS2017 и TON IoT. Чтобы обеспечить сопоставимость результатов были выделены общие классы сетевой активности в двух наборах: XSS, Benign, Bruteforce, DoS/DDoS, SQL Injection и Reconnaissance.

Обучение проводилось на наборе CIC-IDS2017, а тестирование на TON IoT. Оцифрованные данные были приведены к одинаковому диапазону значений с помощью MinMaxScaler. В качестве алгоритма машинного обучения использовался ET со стандартным набором гиперпараметров. Результаты тестирования, проведенные в рамках одного набора данных и при перекрестном тестировании, представлены в таблице 6.

Анализ результатов перекрестного тестирования, представленных в таблице 6, показывает значительную потерю обобщающей способности модели при ее переносе в отличную от обучающей сетевую среду. Когда обучение и тестирование проводились в рамках одного набора данных, модель демонстрировала высокие значения F1-меры с макро-усреднением – 90,27% для CIC-IDS2017 и 95,39% для TON IoT. Однако в сценарии перекрестного тестирования обобщающая способность модели значительно ухудшилась (F1-мера с макро-усреднением составила 16,25%).

Табл. 5. Оценки качества классификации настроенной модели ET (Weighted – взвешенное усреднение, Macro – макро-усреднение)

	Набор данных		
	CIC-IDS2017	**TON IoT**	**USTC-TFC2016**
Гиперпараметры	n_estimators=100 max_features=None max_depth=21 min_samples_split=2 min_samples_leaf=1	n_estimators=500 max_features=None max_depth=31 min_samples_split=2 min_samples_leaf=1	n_estimators=500 max_features=None max_depth=25 min_samples_split=2 min_samples_leaf=4
Accuracy	99,39%	99,76%	97,80%
Weighted Precision	99,41%	99,77%	97,81%
Weighted Recall	99,38%	99,76%	97,80%
Weighted F1-мера	99,39%	99,77%	97,80%
Macro Precision	88,69%	93,64%	97,54%
Macro Recall	96,48%	98,54%	97,34%
Macro F1-мера	91,16%	95,39%	97,43%

Табл. 6. Результаты классификации модели в сценариях обучения и тестирования в рамках одного набора данных и перекрестного тестирования (для F1-меры, Precision и Recall используется макро-усреднение)

Сценарий	**Accuracy**	**Precision**	**Recall**	**F1-мера**
Тестирование в рамках одного набора данных	99,74%	94,07%	97,55%	95,50%
Перекрестное тестирование на двух наборах	26,46%	33,08%	28,21%	16,25%

VI. Методы повышения обобщающей способности

При переносе модели машинного обучения между различными сетевыми инфраструктурами наблюдается снижение качества классификации, вызванное доменным сдвигом. Это связано с различиями характеристик сетевого трафика, обусловленными спецификой инфраструктуры и бизнес-задач. В данном разделе рассматриваются методы, направленные на снижение влияния доменного сдвига и повышение качества классификации.

A. Фильтрация признаков модели данных

Одним из подходов к снижению влияния доменных различий является фильтрация из модели данных признаков, обладающих высокой специфичностью к сетевой среде. Устранение признаков, позволяющих модели распознавать источник данных, снижает переобучение к специфике инфраструктуры и улучшает переносимость модели. Для выделения таких признаков использовалась вспомогательная модель-дискриминатор, обученная классифицировать записи по их происхождению.

В рамках эксперимента использовались два набора данных: CIC-IDS2017 и TON IoT, записи которых были размечены метками environment=0 и environment=1 соответственно. Объединенные данные были разделены на обучающую (70%) и тестовую (30%) выборки, после чего нормализованы. В качестве модели-дискриминатора был использован алгоритм Random Forest из пакета scikit-learn [12], который включает встроенный механизм оценки значимости признаков (атрибут feature_importance).

Анализ значений важности признаков показал, что модель-дискриминатор способна эффективно определять источник трафика: начальная точность на тестовой выборке составила 99,89%. В качестве порога исключения была выбрана медиана распределения значений feature_importance, поскольку использование более высоких порогов приводило лишь к незначительному снижению точности модели-дискриминатора. После исключения признаков с важностью выше медианы точность дискриминатора снизилась до 76,94%.

1) Сценарий обучения и тестирования в рамках одного набора данных

Для оценки влияния фильтрации на точность классификации в рамках одной сетевой среды был использован набор CIC-IDS2017. Данные были разделены на обучающую (70%) и тестовую (30%) выборки. Для уменьшения дисбаланса классов в обучающей выборке использовались методы Random Under Sampling и SMOTE. В качестве алгоритма классификации использовался ET со стандартным набором гиперпараметров. В таблице 7 представлено сравнение результатов тестирования моделей, обученных на полном и отфильтрованном наборе признаков.

Полученные результаты демонстрируют значительное снижение качества классификации после исключения признаков, ассоциированных с конкретной средой. В частности, F1-мера снизилась более, чем на 30%, что указывает на наличие значимой для классификации информации в зависимых от среды признаках и признаках, определяющих класс трафика.

Табл. 7. Результаты тестирования модели в рамках одного набора данных с полным и отфильтрованным набором признаков (для F1-меры, Precision и Recall используется макро-усреднение).

Набор признаков	Accuracy	Precision	Recall	F1-мера
Полный	86,97%	87,82%	95,40%	90,27%
Отфильтрованный	86,96%	92,3%	54,59%	58,62%

Табл. 8. Сравнение результатов перекрестного тестирования с использованием полного и отфильтрованного наборов признаков (для F1-меры, Precision и Recall используется макро-усреднение).

Набор признаков	Accuracy	Precision	Recall	F1-мера
Полный	26,46%	33,08%	28,21%	16,25%
Отфильтрованный	0,54%	5,94%	16,57%	0,2%

2) Сценарий перекрестного тестирования

Во втором эксперименте была оценена способность модели к обобщению при ее переносе в иную сетевую среду. Обучение проводилось на наборе CIC-IDS2017, тестирование – на наборе TON IoT. Для обеспечения сопоставимости результатов были выделены общие классы легитимной и вредоносной сетевой активности в двух наборах. Оцифрованные данные были приведены к одинаковому диапазону значений с помощью MinMaxScaler. В качестве алгоритма машинного обучения использовался ET со стандартным набором гиперпараметров. В таблице 8 представлено сравнение результатов перекрестного тестирования на полном и отфильтрованном наборе признаков.

Полученные результаты перекрестного тестирования демонстрируют значительное снижение качества классификации при использовании отфильтрованного набора признаков. Эти данные свидетельствуют о том, что фильтрация признаков в данном эксперименте не способствовала улучшению обобщающей способности модели при переносе между наборами данных CIC-IDS2017 и TON IoT. Напротив, удаление части признаков привело к значительному снижению всех метрик, что может быть связано с потерей важной информации, необходимой для разделения классов.

B. Использование более обобщающего алгоритма классификации

Результаты, представленные в предыдущем разделе, свидетельствуют о том, что фильтрация признаков модели данных не способствует повышению универсальности и качества классификации модели. Напротив, как при тестировании в рамках одного набора, так и при перекрестном тестировании наблюдалось значительное ухудшение метрик качества. Одной из возможных причин может быть склонность моделей, основанных на деревьях решений, к формированию жестких границ между классами, что ограничивает их способность к обобщению в условиях изменения сетевой среды.

В исследовании [13] рассматривается проблема переносимости моделей машинного обучения между сетями в контексте задачи обнаружения сетевых атак. Исследование показывает, что алгоритмы, демонстрирующие высокую точность в пределах одной сети, значительно теряют эффективность при переносе в иную сетевую среду (на примере наборов данных CIC-IDS2017 и CIC-IDS2018). В то же время, сохранение однородности сетевой среды, как это наблюдается в наборах LUFlow 2020 и LUFlow 2021, не влияет на показатели качества.

Кроме того, в исследовании демонстрируется, что алгоритмы Random Forest, Decision Tree и Naive Bayes имеют большее падение точности при переносе между различными сетевыми средами. В отличие от них модели DNN (Deep Neural Network), ANN (Artificial Neural Network) и SVM (Support Vector Machine) демонстрируют более высокую способность к обобщению.

Основываясь на представленных результатах, в рамках данного исследования была предпринята попытка использовать нейронную сеть в качестве классификатора, чтобы провести оценку ее обобщающей способности. Архитектура нейронной сети состояла из трех полносвязных слоев с функциями активации ReLU (Rectified Linear Unit) и Log SoftMax на выходе. Для обучения модели использовалась функция потерь Cross Entropy, в которой учитывались веса классов для корректировки их дисбаланса. Для оптимизации параметров модели использовался алгоритм Adam. В процессе обучения применялась техника снижения скорости обучения (Reduce LR on Plateau), которая снижает скорость обучения при отсутствии улучшений функции потерь на валидационной выборке.

Для оценки качества классификации нейронной сети в условиях одной сетевой среды был реализован сценарий обучения и тестирования в рамках одного и того же набора данных. Был взят набор CIC-IDS2017, разделенный на обучающую (70%), валидационную (10%) и тестовую (20%) выборки. Предварительно выборки были нормализованы с помощью Robust Scaler.

Для анализа обобщающей способности нейронной сети при ее переносе в другую сетевую среду был реализован сценарий перекрестного тестирования. В рамках эксперимента обучение проводилось на наборе CIC-IDS2017, разделенном на обучающую (90%) и валидационную (10%) выборки, тестирование – на наборе TON IoT. Для обеспечения сопоставимости результатов были выделены общие классы сетевой активности в двух наборах. Предварительно наборы данных были нормализованы с помощью Robust Scaler. Результаты обоих сценариев представлены в таблице 9.

Анализ представленных в таблице 9 результатов тестирования показывает, что нейронная сеть продемонстрировала более низкие результаты по сравнению с алгоритмом ET в рамках одного набора данных (табл. 4).

Табл. 9. Метрики качества нейронной сети в двух сценариях: при тестировании в рамках одной сетевой среды и при переносе обученной модели в иную сетевую среду (для F1-меры, Precision и Recall используется макро-усреднение).

Сценарий	Accuracy	Precision	Recall	F1-мера
Тестирование в рамках одного набора данных	98,96%	78,74%	75,34%	75,11%
Перекрестное тестирование на двух наборах	39,68%	26,30%	29,39%	19,23%

Табл. 10. Метрики качества архитектуры DANN в двух сценариях: адаптация к целевому домену на полном наборе классов и адаптация к целевому домену с использованием только класса легитимной сетевой активности (для F1-меры, Precision и Recall используется макро-усреднение).

Сценарий	Accuracy	Precision	Recall	F1-мера
Полный набор классов	52,90%	41,99%	32,97%	23,84%
Только класс легитимной активности	11,47%	23,73%	15,51%	7,35%

Однако, в сценарии перекрестного тестирования нейронная сеть показала большую способность к обобщению (табл. 6). Тем не менее, полученные значения метрик качества остаются недостаточно высокими для практического применения модели в сценариях переноса между различными сетевыми инфраструктурами.

C. Применение метода доменной адаптации

Нейронная сеть показала большую способность к обобщению в сравнении с классическим алгоритмом ET, однако достигнутые результаты недостаточны для практического применения подхода в задаче обнаружения сетевых атак. С целью повышения обобщающей способности системы в рамках настоящего исследования был протестирован метод доменно-состязательного обучения DANN (Domain-Adversarial training of Neural Networks) [14]. Данный подход обеспечивает возможность адаптации модели к новой среде при отсутствии соответствующих размеченных данных.

Архитектура DANN состоит из трех основных компонентов: экстрактор (feature extractor), классификатор (label predictor) и дискриминатор (domain classifier). На вход экстрактору поступают записи, как из исходного, так и из целевого домена. Извлеченные из двух доменов признаки передаются дискриминатору, тогда как классификатор получает признаки только исходного домена. Во время обучения классификатор и дискриминатор минимизируют свои функции ошибок, стремясь повысить точность предсказания меток и определения домена. Экстрактор обучается таким образом, чтобы минимизировать ошибку классификатора и максимизировать ошибку дискриминатора. Это достигается при помощи добавления слоя GRL (Gradient Reversal Layer), который умножает градиент дискриминатора на отрицательный параметр λ во время обратного распространения ошибки. В результате экстрактор формирует инвариантные к домену признаки.

Конфигурация компонентов архитектуры DANN:

- экстрактор: два полносвязных слоя с функциями активации ReLU;
- классификатор: два полносвязных слоя с функциями активации ReLU и Log SoftMax на выходе;
- дискриминатор: два полносвязных слоя с функциями активации ReLU и Sigmoid на выходе.

Для классификатора и дискриминатора использовались следующие функции потерь: Cross Entropy, учитывающая веса классов для корректировки их дисбаланса, и Binary Cross Entropy соответственно. Для оптимизации параметров моделей использовался алгоритм Adam. Значения параметра λ и скорости обучения динамически изменялись в процессе обучения в соответствии с подходом, описанным в оригинальной работе [14].

Для тестирования были взяты наборы данных CIC-IDS2017 и TON IoT, и выделены общие классы сетевой активности в двух наборах.

В первом сценарии тестирования метода DANN был использован весь набор классов целевого домена. Исходный домен (CIC-IDS2017) был разделен на обучающую (70%) и тестовую (30%) выборки, целевой домен (TON IoT) был также разделен на обучающую (10%) и тестовую (90%) выборки. Предварительно данные были нормализованы с помощью Robust Scaler.

Во втором сценарии тестовая выборка TON IoT ограничивалась исключительно записями класса легитимной сетевой активности, что моделирует ситуацию, когда данные атак в целевом домене недоступны. Исходный домен (CIC-IDS2017) был разделен на обучающую (70%) и тестовую (30%) выборки. Из целевого домена (TON IoT) было выбрано 70% класса легитимного трафика, оставшаяся часть набора использовалась для тестирования. Предварительно данные были нормализованы с помощью Robust Scaler. В таблице 10 показаны результаты тестирования для обоих сценариев.

Анализ результатов из таблицы 10 показывает, что использование метода доменно-состязательного обучения не обеспечивает достаточного уровня качества для практического применения в задачах обнаружения сетевых атак в условиях переноса модели между различными доменами. В сценарии доменной адаптации с полным набором классов целевого домена модель достигла точности (Accuracy) 52,90% и F1-меры 23,84%. Эти результаты свидетельствуют о некоторой способности модели к обобщению, однако метрики качества остаются недостаточными для эффективного применения в реальных условиях. В сценарии с использованием только легитимного класса сетевого трафика целевого домена точность модели составила 11,47%, а F1-мера 7,35%. Такие низкие показатели указывают на существенные ограничения метода доменно-состязательного обучения в условиях, когда данные об атаках в целевом домене недоступны.

VII. Заключение

В рамках данного исследования была предпринята попытка разработки и оценки системы обнаружения и классификации компьютерных атак в зашифрованном сетевом трафике с применением методов машинного обучения независимо от используемого протокола защиты данных. Была создана комплексная модель данных, включающая статистические характеристики потоков, анализ последовательностей длин пакетов и временных интервалов через цепи Маркова, а также распределение байт в полезной нагрузке пакетов.

Сравнительное тестирование на трех наборах данных (CIC-IDS2017, TON IoT, USTC-TFC2016) показало, что выбранные алгоритмы машинного обучения, в частности Extra Trees, способны достигать высокого качества классификации в рамках одной сетевой среды (F1-мера 90-95% после оптимизации гиперпараметров).

Однако, ключевым ограничением, выявленным в ходе исследования, стала низкая обобщающая способность моделей при переносе между различными сетевыми средами (различными наборами данных). Перекрестное тестирование показало резкое падение качества (F1-мера до 16,25%), указывая тем самым на значительную зависимость моделей от специфики среды обучения. Исследованные подходы к повышению переносимости (фильтрация признаков, использование нейронной сети, доменная адаптация) не привели к созданию достаточно устойчивого и универсального решения (лучший результат в перекрестном тестировании показал метод DANN с F1-мерой 23,84% при условии наличия разметки целевого домена).

Таким образом, несмотря на возможности использования методов машинного обучения для обнаружения атак в зашифрованном трафике в известной среде, разработка универсальной модели, эффективно работающей в гетерогенных сетевых инфраструктурах, остается нерешенной задачей. Ее решение требует дальнейших исследований, перспективными направлениями которых представляются:

- развитие и применение более совершенных методов доменной адаптации, способных лучше справляться с доменным сдвигом в статистических характеристиках сетевого трафика;
- развитие модели данных с целью формирования инвариантных к конкретной сетевой среде признаков, сохраняющих информативность для модели машинного обучения.

Библиография

[1] Отчет о доступности сервисов и данных: [Электронный ресурс]. – Режим доступа: https://transparencyreport.google.com/archive/https/overview (дата обращения: 12.03.2025).

[2] Anderson B., Paul S., McGrew D. A. Deciphering malware's use of TLS (without decryption) // Journal of Computer Virology and Hacking Techniques. – 2016. – Vol. 14. – P. 195–211.

[3] Gouveia A., Correia M. P. Network intrusion detection with XGBoost // Recent Advances in Security, Privacy, and Trust for Internet of Things (IoT) and Cyber-Physical Systems (CPS). – 2020.

[4] Koumar J., Hynek K., Čejka T. Network traffic classification based on single flow time series analysis // Proc. 2023 19th Int. Conf. Network and Service Management (CNSM). – 2023. – P. 1–7. – DOI: 10.23919/CNSM59352.2023.10327876.

[5] Singh A. P., Singh M., Bhatia K. [и др.] Encrypted malware detection methodology without decryption using deep learning-based approaches // Turkish Journal of Engineering. – 2024. – Vol. 8. – P. 498–509. – DOI: 10.31127/tuje.1416933.

[6] Belarbi O., Khan A., Carnelli P. E. [и др.] An intrusion detection system based on deep belief networks // Proc. Int. Conf. Science of Cyber Security. – 2022.

[7] CICFlowMeter / UNB CIC : [Электронный ресурс]. – Режим доступа: https://www.unb.ca/cic/research/applications.html (дата обращения: 12.03.2025).

[8] Sharafaldin I., Lashkari A. H., Ghorbani A. A. Toward generating a new intrusion detection dataset and intrusion traffic characterization // Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP), Portugal, Jan. 2018. – Funchal, Madeira, Portugal : SciTePress, 2018.

[9] Moustafa N. A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets // Sustainable Cities and Society. – 2021. – Vol. 72. – Art. no. 102994. – DOI: https://doi.org/10.1016/j.scs.2021.102994.

[10] Wang W., Zhu M., Zeng X. [и др.] Malware traffic classification using convolutional neural network for representation learning // Proc. 2017 Int. Conf. Information Networking (ICOIN). – Da Nang, Vietnam, 2017. – P. 712–717. – DOI: 10.1109/ICOIN.2017.7899588.

[11] Optuna - A hyperparameter optimization framework / Optuna : [Электронный ресурс]. – Режим доступа: https://optuna.org (дата обращения: 12.03.2025).

[12] Scikit-Learn / Scikit-learn : [Электронный ресурс]. – Режим доступа: https://scikit-learn.org/stable/index.html (дата обращения: 12.03.2025).

[13] Chua T.-H., Salam I. Evaluation of machine learning algorithms in network-based intrusion detection system : arXiv preprint arXiv:2203.05232 : [Электронный ресурс]. – 2022. – Режим доступа: https://arxiv.org/abs/2203.05232 (дата обращения: 12.03.2025).

[14] Ganin Y., Ustinova E., Ajakan H. [и др.] Domain-adversarial training of neural networks // Journal of machine learning research. – 2016. – Vol. 17, no. 59. – P. 1–35.

[15] Booij T. M., Chiscop I., Meeuwissen E. [и др.] ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets // IEEE Internet Things J. – 2022. – Vol. 9, no. 1. – P. 485–496. – DOI: 10.1109/JIOT.2021.3085194.

[16] Alsaedi A., Moustafa N., Tari Z. [и др.] TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems // IEEE Access. – 2020. – Vol. 8. – P. 165130–165150. – DOI: 10.1109/ACCESS.2020.3022862.

[17] Moustafa N., Keshky M., Debiez E. [и др.] Federated TON_IoT windows datasets for evaluating AI-based security applications // Proc. 2020 IEEE 19th Int. Conf. Trust, Security and Privacy Comput. Commun. (TrustCom). – 2020. – P. 848–855. – DOI: 10.1109/TrustCom50675.2020.00114.

[18] Moustafa N., Ahmed M., Ahmed S. Data analytics-enabled intrusion detection: Evaluations of ToN_IoT Linux datasets // Proc. 2020 IEEE 19th Int. Conf. Trust, Security and Privacy Comput. Commun. (TrustCom). – 2020. – P. 727–735. – DOI: 10.1109/TrustCom50675.2020.00100.

[19] Moustafa N. New generations of Internet of Things datasets for cybersecurity applications based machine learning : Presented at the eResearch Australasia Conf., Brisbane, Australia, 2019 : [Электронный ресурс]. – Режим доступа: https://conference.eresearch.edu.au/wp-content/uploads/2019/08/2019_eResearch_59_New-Generations-of-Internet-of-Things-Datasets-for-Cybersecurity.pdf (дата обращения: 12.03.2025).

[20] Moustafa N. A systemic IoT-fog-cloud architecture for big-data analytics and cyber security systems: A review of fog computing : arXiv preprint arXiv:1906.01055 : [Электронный ресурс]. – 2019. – Режим доступа: https://arxiv.org/abs/1906.01055 (дата обращения: 12.03.2025).

[21] Ashraf J., Moustafa N., Khurshid H. [и др.] IoTBoT-IDS: A novel statistical learning-enabled botnet detection framework for protecting networks of smart cities // Sustainable Cities and Society. – 2021. – Vol. 72. – Art. no. 103041. – DOI: https://doi.org/10.1016/j.scs.2021.103041.

Capabilities and limitations of attack classification in encrypted traffic by machine learning methods

M. Bondarev, O. Guzev

***Abstract*—This paper investigates the capabilities and limitations of applying machine learning methods for computer attacks classifying within encrypted network traffic. This research is relevant due to the widespread adoption of encryption, which complicates the operation of traditional traffic analysis methods. The article proposes a comprehensive data model incorporating statistical flow characteristics, packet sequences, and payload byte distribution.**

A comparative analysis of machine learning algorithms: Random Forest, Extra Trees, Decision Tree, AdaBoost, and KNN was conducted on three public datasets: CIC-IDS2017, TON IoT, and USTC-TFC2016. Random Forest and Extra Trees algorithms demonstrated the best results, Extra Trees showing slightly higher stability. Hyperparameter optimization for the Extra Trees model was performed, proving its effectiveness when applied within a single dataset.

A key limitation of the proposed approach is the model's low generalization capability. Cross-dataset testing revealed a significant decrease in classification performance, indicating a dependency of the results on the specific network environment. Applied methods to improve generalization, including feature filtering, neural network implementation, and domain adaptation, did not yield a sufficiently robust or universally applicable solution.

The study concludes that machine learning models are applicable for effective detection and classification of computer attacks within a single network environment. However, the portability of trained models between different network environments remains limited and requires further research.

***Keywords*—Network traffic analysis, attack classification, machine learning, generalization capability, encrypted traffic**

REFERENCES

[1] Google Transparency Report. [Online]. Available: https://transparencyreport.google.com/archive/https/overview.

[2] B. Anderson, S. Paul, and D. A. McGrew, "Deciphering malware's use of TLS (without decryption)," Journal of Computer Virology and Hacking Techniques, vol. 14, pp. 195–211, 2016.

[3] A. Gouveia and M. P. Correia, "Network intrusion detection with XGBoost," Recent Advances in Security, Privacy, and Trust for Internet of Things (IoT) and Cyber-Physical Systems (CPS), 2020.

[4] J. Koumar, K. Hynek, and T. Čejka, "Network traffic classification based on single flow time series analysis," in Proc. 2023 19th Int. Conf. Network and Service Management (CNSM), 2023, pp. 1–7, doi: 10.23919/CNSM59352.2023.10327876.

[5] A. P. Singh, M. Singh, K. Bhatia, and H. Pathak, "Encrypted malware detection methodology without decryption using deep learning-based approaches," Turkish Journal of Engineering, vol. 8, pp. 498–509, 2024, doi: 10.31127/tuje.1416933.

[6] O. Belarbi, A. Khan, P. E. Carnelli, and T. Spyridopoulos, "An intrusion detection system based on deep belief networks," in Proc. Int. Conf. Science of Cyber Security, 2022.

[7] UNB CIC, "CICFlowMeter." [Online]. Available: https://www.unb.ca/cic/research/applications.html.

[8] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP), Portugal, Jan. 2018.

[9] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets," Sustainable Cities and Society, vol. 72, Art. no. 102994, 2021, doi: https://doi.org/10.1016/j.scs.2021.102994.

[10] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in Proc. 2017 Int. Conf. Information Networking (ICOIN), 2017, pp. 712–717, doi: 10.1109/ICOIN.2017.7899588.

[11] Optuna, "Optuna - A hyperparameter optimization framework." [Online]. Available: https://optuna.org/.

[12] Scikit-learn, "Scikit-Learn." [Online]. Available: https://scikit-learn.org/stable/index.html.

[13] T.-H. Chua and I. Salam, "Evaluation of machine learning algorithms in network-based intrusion detection system," arXiv preprint arXiv:2203.05232, 2022. [Online]. Available: https://arxiv.org/abs/2203.05232.

[14] Y. Ganin et al., "Domain-adversarial training of neural networks," Journal of machine learning research, vol. 17, no. 59, pp. 1–35, 2016.

[15] T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. H. den Hartog, "ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets," IEEE Internet Things J., vol. 9, no. 1, pp. 485–496, Jan. 2022, doi: 10.1109/JIOT.2021.3085194.

[16] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," IEEE Access, vol. 8, pp. 165130–165150, 2020, doi: 10.1109/ACCESS.2020.3022862.

[17] N. Moustafa, M. Keshky, E. Debiez, and H. Janicke, "Federated TON_IoT windows datasets for evaluating AI-based security applications," in Proc. 2020 IEEE 19th Int. Conf. Trust, Security and Privacy Comput. Commun. (TrustCom), 2020, pp. 848–855, doi: 10.1109/TrustCom50675.2020.00114.

[18] N. Moustafa, M. Ahmed, and S. Ahmed, "Data analytics-enabled intrusion detection: Evaluations of ToN_IoT Linux datasets," in Proc. 2020 IEEE 19th Int. Conf. Trust, Security and Privacy Comput. Commun. (TrustCom), 2020, pp. 727–735, doi: 10.1109/TrustCom50675.2020.00100.

[19] N. Moustafa, "New generations of Internet of Things datasets for cybersecurity applications based machine learning: TON_IoT datasets," presented at the eResearch Australasia Conf., Brisbane, Australia, 2019. [Online]. Available: https://conference.eresearch.edu.au/wp-content/uploads/2019/08/2019_eResearch_59_New-Generations-of-Internet-of-Things-Datasets-for-Cybersecurity.pdf.

[20] N. Moustafa, "A systemic IoT-fog-cloud architecture for big-data analytics and cyber security systems: A review of fog computing," arXiv preprint arXiv:1906.01055, 2019. [Online]. Available: https://arxiv.org/abs/1906.01055.

[21] J. Ashraf et al., "IoTBoT-IDS: A novel statistical learning-enabled botnet detection framework for protecting networks of smart cities," Sustainable Cities and Society, vol. 72, Art. no. 103041, 2021, doi: https://doi.org/10.1016/j.scs.2021.103041.