ИНФОРМАТИКА, УПРАВЛЕНИЕ И СИСТЕМНЫЙ АНАЛИЗ

УДК 004.89 ***EDN: PMAPDM***

ИДЕНТИФИКАЦИЯ ПРИЛОЖЕНИЙ ПО СЕТЕВОМУ ТРАФИКУ

Г.Д. Кузнецов
ORCID: **0000-0001-5564-045X** e-mail: **gd.smith@yandex.ru**
Нижегородский государственный технический университет им. Р.Е. Алексеева
Нижний Новгород, Россия

В.Е. Гай
ORCID: **0000-0002-3644-5234** e-mail: **iamuser@inbox.ru**
Нижегородский государственный технический университет им. Р.Е. Алексеева
Нижний Новгород, Россия

Исследована возможность идентификации приложений по сетевому трафику с применением классических методов классификаторов. Предложены модель сбора и обработки входных сетевых данных, а также алгоритм формирования признакового описания для классификации сетевых приложений с целью повышения точности идентификации программ. Проведены эксперименты и анализ полученных результатов, которые позволили выявить преимущества и недостатки используемых методов классификации.

Ключевые слова: идентификация, сетевой трафик, сетевые пакеты, сетевые данные, классификация, признаковое описание.

ДЛЯ ЦИТИРОВАНИЯ: Кузнецов, Г.Д. Идентификация приложений по сетевому трафику / Г.Д. Кузнецов, В.Е. Гай // Труды НГТУ им. Р.Е. Алексеева. 2024. № 4. С. 7-16. EDN: PMAPDM

APPLICATION IDENTIFICATION BY NETWORK TRAFFIC

G.D. Kuznetsov
ORCID: **0000-0001-5564-045X** e-mail: **gd.smith@yandex.ru**
Nizhny Novgorod State Technical University n.a. R.E. Alekseev
Nizhny Novgorod, Russia

V.E. Gai
ORCID: **0000-0002-3644-5234** e-mail: **iamuser@inbox.ru**
Nizhny Novgorod State Technical University n.a. R.E. Alekseev
Nizhny Novgorod, Russia

Abstract. The paper presents the possibility of identifying applications by network traffic. Classical classifier methods are applied for identification. The paper proposes a model for collecting and processing input network data, as well as an algorithm for forming a feature description for classifying network applications to improve the accuracy of application identification. The results of the experiments made it possible to identify the advantages and disadvantages of the classification methods used.

Key words: identification, network traffic, network packets, network data, classification, feature description.

FOR CITATION: G.D. Kuznetsov, V.E. Gai. Application identification by network traffic. Transactions of NNSTU n.a. R.E. Alekseev. 2024. № 4. Pp. 7-16. EDN: PMAPDM

Введение

Идентификация приложений по сетевому трафику позволяет определить, какая программа сгенерировала сетевую активность. Сетевые приложения создают уникальный поток данных и обладают характерными поведенческими особенностями.

В рамках данной работы предлагается алгоритм формирования признакового описания. Это необходимо для решения задач классификации приложений, которые выполняют обмен данными по сети. Алгоритм основан на выделении временных, переменных и статистических характеристик поведения программ и призван улучшить точность идентификации приложений на основе анализа их сетевой активности. Анализ сетевого трафика и идентификация сетевых приложений могут быть использованы для мониторинга сетевой активности, определения угроз безопасности, оптимизации работы сетевых приложений, повышения эффективности сетевых систем, контроля пиковых нагрузок на сеть и отслеживания используемых ресурсов сетевыми узлами [1].

Обзор существующих методов идентификацией приложений по сетевому трафику

Для идентификации приложений существует несколько методов анализа.

1. *Анализ статистических характеристик* позволяет классифицировать сетевые приложения за счет выделения количественных свойств. Например, выделение размеров пакетов, отклонение времени между передаваемыми пакетами, их количество [2]. Не все сетевые приложения имеют предсказуемое поведение в процессе обмена информации; поэтому данный метод может не быть эффективным в задачах классификации сетевых приложений.

2. *Анализ протокола, состояний* – более эффективный метод в случае отсутствия шифрования трафика, поскольку идентификация обеспечивается за счет обращения к удаленному серверу и сверкой возвращенного результата [3].

3. *Анализ образца* – метод, при котором содержимое передаваемых данных в сетевых пакетах может содержать информацию с уникальной последовательностью байтов данных. За счет выделения подобных последовательностей можно классифицировать сетевое приложение. Но зачастую бинарные данные сетевых пакетов передаются зашифрованными, поэтому можно классифицировать только малое количество приложений. Также достаточно сложно определить, к какой программе могут относиться передаваемые аудио- и видеоматериалы. Помимо этого, подобная обработка информации может затрудняться за счет случайных совпадений последовательности байт и объема передаваемых данных.

4. *Анализ сетевого трафика с помощью глубокого исследования пакетов DPI (Deep Package Inspection)* имеет высокую точность в вопросах идентификации трафика. Он анализирует непосредственно содержимое передаваемых данных [4]. Такой метод также требует повышенной производительности ЭВМ, на которой производится классификация трафика. На основе DPI существует пакетный метод классификации трафика. Результат идентификации основывается на анализе содержимого отдельных сетевых пакетов. Также существует метод, основанный на потоках, позволяющий анализировать несколько сетевых посылок в течение некоторого времени. При использовании методов глубокого анализа пакетов могут возникать проблемы конфиденциальности пользовательских данных и программ [5].

5. *Совместное использование поведенческого анализа с эвристическим* позволяет анализировать сетевой трафик за счет выделения характерных особенностей поведения программ. Генерируя сетевую активность, можно определить приложение за счет выделения признаков. Подобный анализ требует значительно меньше вычислительных ресурсов.

Предлагаемая модель и алгоритмы идентификации приложений на основе сетевого трафика

Выполнять идентификацию приложений на основе сетевых пакетов предлагается с помощью статистических методов. Это позволит выполнять операцию быстрее на менее требовательных ЭВМ. Для этого необходимо предварительно проанализировать и обработать входные данные. Идентификацию сетевых приложений рекомендуется реализовать поэтапно: выполнить сбор и обработку данных, сформировать признаки, затем обучить модель (рис. 1). За счет формирования приложениями уникальных поведенческих привычек выпол-няется формирование уникальных признаковых описаний.

Рис. 1. Модель классификации сетевых приложений, этап обучения

Fig. 1. Network application classification model, training stage

Определим обучающую выборку как $X^l = (x_i, y_i)^l, i = \overline{1,l}$, где X^l является обучающей выборкой, x_i – конечная последовательность сетевых дампов данных, y_i принадлежит от 0 до N и обозначают к какому классу относится идентифицируемые сетевые программы.

Дополнительно существует некоторая зависимость (отображение), значения которых известны непосредственно на самих элементах выборки данных участвующих в обучение: $F: X \to D_f$,где D_f является множество значений признака, которые допустимы в решение задачи.

Когда классификатор будет обучен, модель можно будет применять для идентификации сетевых приложений. Для оценки работы модели формируются метрики, которые демонстрируют качество работы обученных моделей.

Рис. 2. Модель классификации сетевых приложений, этап применения

Fig. 2. Network application classification model, application stage

На рис. 2 x является конечной последовательностью сетевых дампов данных. $a(x,w)$ в общем виде является самой моделью классификатора, где x – признаки объекта, а w – неизвестные параметры [6, 7]. В процессе обучения модели для идентификации сетевых приложений выполняется поиск критериев оптимальности модели:

$$\sum_i^l L_i(a(x_i^{'}w)) \to \min_w, \tag{1}$$

где L_i обозначает функцию потерь модели $a(x_i^{'}, w)$ на объекте x_i.

Функционал качества модели можно вычислить по формуле:

$$Q: Q(a, X^l) = \frac{1}{l}\sum_{i=1}^{l} L(a(x_i^{'}, w), y^*(x_i^{'})), \tag{2}$$

где $y^*(x_i^{'})$ обозначает истинный класс объекта x_i.

Обучение модели μ основано на минимизации эмпирического риска, которая рассчитывается по следующей формуле:

$$\mu(X^l) = \arg\min_{a \in A} Q(a, X^l), \quad (3)$$

где A – множество моделей.

Предлагаемый алгоритм сбора и обработки входных данных

Этап сбора и разметки данных. В процессе накопления сетевых пакетов, которые будут использованы для обучения классификатора, ключевым этапом для идентификации является разметка собранных данных. Она необходима для определения совокупности признаков, характеризующих анализируемый объект. Важно, чтобы разметка была однородной для всех типов данных, так как различные объекты должны описываться одними и теми же свойствами. Различия в этих полях будут определять тип идентифицируемого сетевого приложения.

Этап оптимизации для обработки информации. Размеры исходных файлов, содержащих сетевые данные, в таких форматах, как JSON-структуры, могут быть очень большими. Они содержат не только информацию о назначениях посылок и их характеристиках, но и бинарные данные. Поэтому для эффективной обработки требуется использовать подходы, которые позволяют считывать данные постепенно, по мере необходимости, в зависимости от ресурсов вычислительных машин.

Этап очистки данных. Перед обучением классификатора, необходимо применить фильтр для удаления негативно влияющих сетевых пакетов в процессе генерации моделей, а именно:

1) удаление неполных пакетов – в данных могут присутствовать такие выборки, которые могут исказить обучение моделей. Например, при работе сетевого устройства возможно возникновение пакетов, которые содержат ошибки в процессе приема или передачи данных. Наиболее частые проблемы происходят в процессе установки соединения с удаленным узлом по протоколу TCP/IP;

2) удаление избыточных пакетов – некоторые данные в процессе обмена могут быть избыточными по причине дублирования информации. В свою очередь эти данные будут влиять на точность классификации. Также необходимо проверять и удалять служебную информацию, которая не относится к сетевым приложениям, например ARP запросы;

3) удаление нерелевантных признаков – некоторые поля в пакетах могут содержать в себе данные, которые будут мешать процессу классификации или даже нанести вред в точности определения объекта. Передаваемые служебные данные могут отличаться в зависимости от конфигурации сетевого оборудования и самого наблюдаемого узла. Например, могут быть такие признаки, которые хранят в себе адреса узлов, относящиеся к локальной вычислительной сети, а не внешним адресам в глобальной сети.

Предлагаемая схема обработки данных

1. *Поэтапное считывание данных* необходимо для предотвращения переполнения оперативной памяти вычислительной машины. Поскольку объемы обрабатываемых данных достаточно велики, то и их обработку необходимо выполнять по мере заполнения оперативной памяти.

2. *Форматирование пакетов:* после достижения предела загружаемых данных выполняется этап очистки сетевых данных, которые попадают под категории описанных в этапе очистки данных.

3. После разбора загруженных данных в память приложения выполняется *процедура*

последующего чтения входного файла с исходными данными, продолжающаяся до тех пор, пока исходный файл не закончится.

4. *Сохранение результата*: после завершения обработки исходного файла с сетевым дампом данных программа выполняет сохранения результатов в JSON структуру. Размер выходного файла значительно снизится, что упростит дальнейшую обработку вычисления признакового описания.

На рис. 3 описан первичный алгоритм обработки данных, полученных из анализатора пакетов.

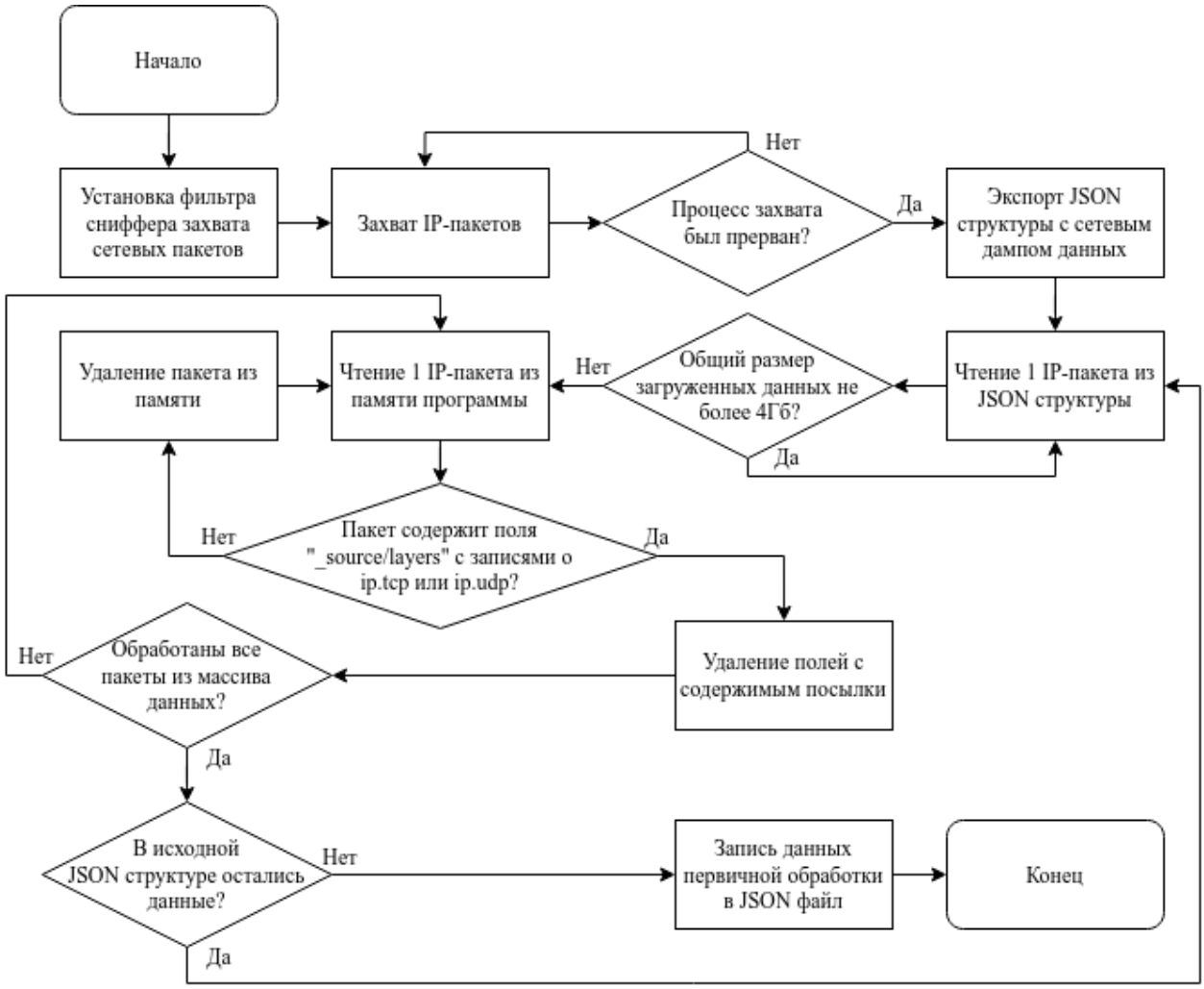

Рис. 3. Алгоритм сбора и предварительной обработки данных

Fig. 3. Algorithm for collecting and pre-processing data

Предлагаемый алгоритм формирования признакового описания сетевого трафика

Повышение точности классификации сетевых приложений напрямую зависит от правильного выделения характерных особенностей их поведения. Необходимо сформировать набор признаков, точно описывающих сетевую активность приложений. Данный процесс можно представить в виде алгоритма (рис. 4).

Этот алгоритм предназначен для выделения уникальных паттернов в сетевых данных, он позволяет отличить одно сетевое приложение от другого. Правильно подобранные признаковые описания позволят классификатору более точно определить, к какому классу относится то или иное приложение.

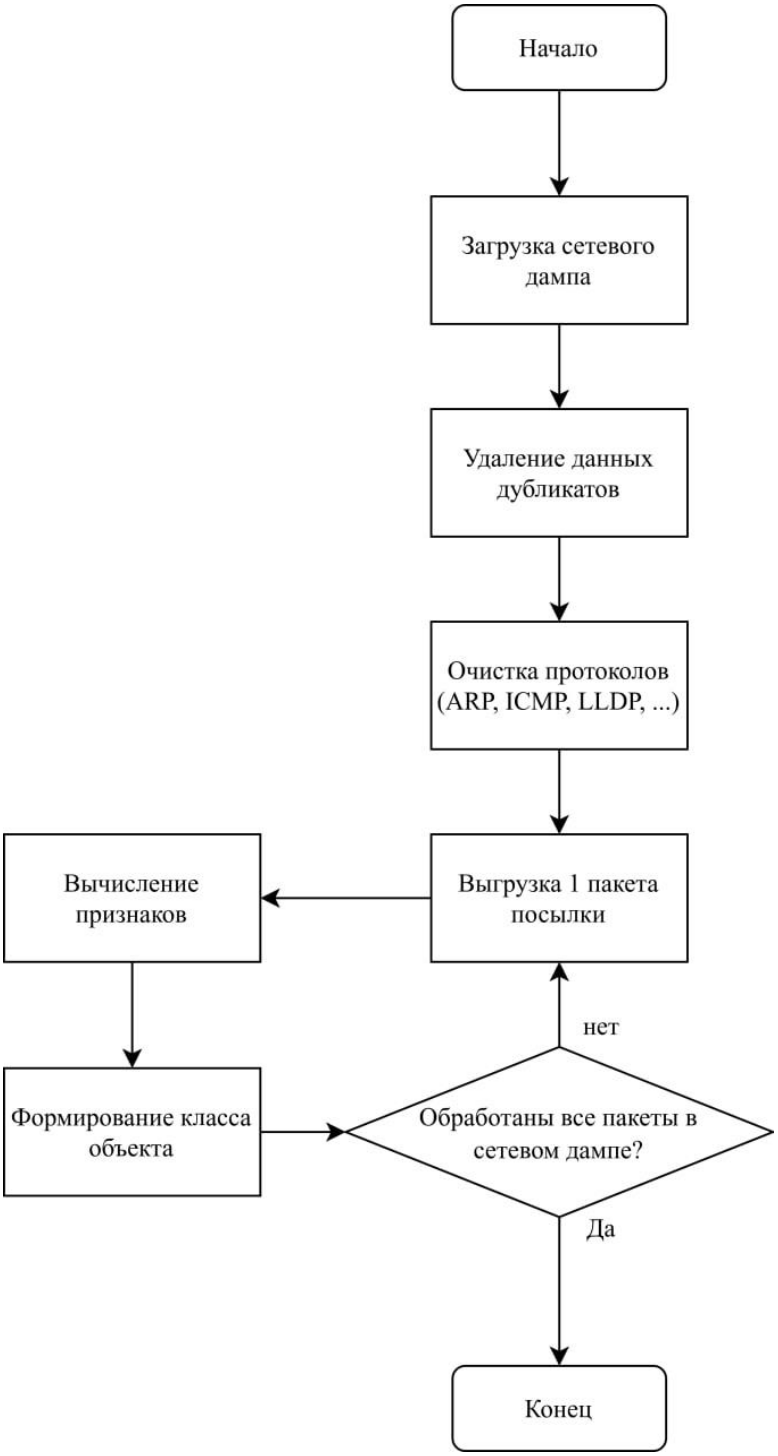

Рис. 4. Алгоритм формирования признакового описания для идентификации приложений по сетевому трафику

Fig. 4. Algorithm for generating a feature description for identifying applications from network traffic

Процесс сбора исходных данных с сетевой активностью приложений

При обучении моделей возникает проблема, связанная с необходимостью генерации обширных баз с данными, так как они имеют отношение к классифицируемым приложениям. Данные должны содержать разнообразные сценарии использования сетевых приложений. Помимо этого, может возникнуть ряд проблем.

1. *Конфиденциальность информации* – сетевые дампы в случае отсутствия шифрования могут содержать приватные данные пользователей (имена, пароли, номера телефонов, личные сообщения и т.д.) Распространение такой информации без согласия пользователей является недопустимым.

2. *Обширное количество сетевых сессий* – трафик, передаваемый по сети, очень разнообразен. Сессии пользователей отличаются по длительности, поведению в используемых сервисах, типу используемых протоколов, объему переданных данных и многим другим параметрам.

3. *Структурирование данных* – собранная информация должна быть структурирована для использования в задачах обучения классификатора. Необходимо создать такой формат данных, который будет удобно обрабатывать.

Формирование базы с трафиком данных осуществляется с использованием специального программного обеспечения, которое позволяет прослушивать все принятые и переданные данные на сетевом интерфейсе без их изменения. Такие программы называются снифферами (например, утилита «Wireshark»). Перед захватом данных в программе можно настроить фильтры, по которым будут отсеивается не интересующие пакеты. После этого возможен экспорт набранных данных в JSON-структуру.

Для правильного обучения моделей, способных определять сетевые приложения, необходимо установить сниффер на узел, в котором планируется наблюдение за сетевой активностью конкретного приложения, а также обеспечить отсутствие посторонней активности, кроме наблюдаемого класса программы. Данные, которые будут использоваться для анализа, можно собирать на устройствах пользователей, где производится доступ к различным сервисам. Такие наборы данных будут участвовать в процессе детектирования приложений.

Признаки сетевых приложений

Предлагается выделить следующие семь признаков для классификации программ.

1. Признаки сетевых протоколов транспортного уровня TCP/IP и UDP/IP наиболее популярны. Они используются для установки соединения клиентских приложений с удаленными серверами. В процессе первичной обработки входного файла подобные запросы не очищаются.

2. Признак с номером порта назначения в протоколах TCP/IP и UDP/IP предназначен для работы приложений на данных протоколах необходимы номера портов, по которым серверные приложения ожидают входящие соединения. В клиентских приложениях исходящие порты игнорируются, так как они могут изменяться. Удаленный сервис не ищет клиента, а клиент сам инициирует обращение.

3. Признаки адреса удаленного сервера в сети и доменного имени DNS (при его наличии) необходимы для определения к какому ресурсу выполняется обращение. Может учитываться как адрес источника, так и адрес назначения (адрес анализируемого узла игнорируется). Имя сервера, или его доменное имя, необходимо для определения конкретного сервиса, размещенного на сервере. На одном сервере, который имеет один или несколько IP-адресов, может быть размещено несколько сервисов. Например, один сервер обрабатывает несколько web-сайтов.

4. Признаки с временными характеристиками сетевого пакета необходимы для определения обращения пакетов клиентского приложения к удаленным ресурсам во времени.

5. Признак установки сетевых сессий во времени учитывается время прошедшее с открытия предыдущей сетевой сессии. Определяется периодичность установки соединений. Данный признак характеризует уникальное поведение программ во времени.

6. Признак частоты передаваемых пакетов позволяет определить количество переданных фрагментов данных за определенный период времени.

7. Длинна сообщений в сетевых пакетах – это средние размеры передаваемых бинарных данных. Размер имеет разную величину в зависимости от используемых сетевых программ.

Для обучения и тестирования моделей машинного обучения был собран объем уникальных данных в 5 гигабайт по каждому из классифицируемых приложений. Данные были разделены на обучающие и тестовые наборы в пропорции 80 к 20 %. Это обеспечивает оптимальный баланс между обучением модели и проведения тестирования модели. Также такое соотношение позволит проверить модель на возможность обобщать результаты на новых наборах сетевых данных. Новые данные также проходят первичную обработку с формированием признаков, затем приложения классифицируются на основе ранее сгенерированных моделей.

Результаты

В эксперименте участвовало несколько подобных программ:

- Telegram – 0 класс;
- Steam – 1 класс;
- Discord – 2 класс.

Предлагаются следующие критерии для оценки адекватности обученных моделей: доля верных ответов (точность), точность (precision), F-мера и полнота [8]. Результаты полученных метрик отражают, что все методы показали высокую точность идентификации сетевых приложений (табл. 1).

Таблица 1.
Метрики качества классификаторов

Table 1.
Classifier quality metrics

Метод	Accuracy	Precision	Recall	F-мера
Model decision tree classifier	0.9999	0.9998	0.9999	0.9999
Model SVM	0.9281	0.9229	0.9285	0.9240
Model logistic regression	0.9998	0.9999	0.9999	0.9999
Model k-nearest neighbors classifier	0.9082	0.9106	0.9082	0.9087
Model multinomial Naive Bayes classifier	0.9263	0.9312	0.9263	0.9274
Model random forest classifier	0.9903	0.9901	0.99	0.9898

С помощью метода кросс-валидации был построен график для оценки качества работы разных моделей, обученных на тестовых наборах данных (рис. 5). При тестировании использовались следующие методы:

- SVC (ovo);
- LinearSVC (ova);
- LogisticRegression;
- DecisionTreeClassifier;
- KneighborsClassifier;
- MultinomialNB;
- RandomForestClassifier.

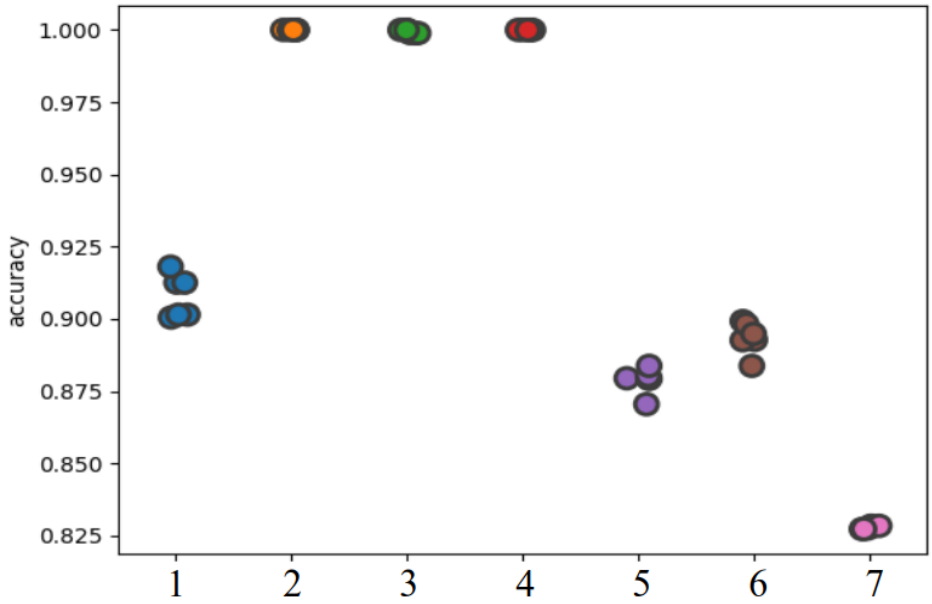

Рис. 5. Результаты точности классификации сетевых приложений различными моделями

Fig. 5. Classification accuracy results of network applications by different models

В процессе проведения вычислительного эксперимента лучшие результаты по точности классификации приложений показали:

1) метод опорных векторов;
2) метод логистической регрессии;
3) метод К ближайших соседей.

Данные результаты демонстрируют высокую эффективность использования методов машинного обучения для идентификации сетевых приложений на основе анализа сетевого трафика. Все используемые методы дают высокую точность идентификации приложений, за исключением «RandomForestClassifier».

Заключение

Рассмотрены методы классификаторов для задач идентификации приложений по сетевому трафику. При выполнении аналитического обзора существующих методов классификации сетевой активности был выявлен ряд недостатков. Предложена новая модель, которая способна идентифицировать сетевые приложения за счет поверхностного анализа сетевого трафика. В ходе проведения вычислительного эксперимента выявлено, что предложенные алгоритмы для идентификации приложений дают сопоставимые результаты с аналогичными системами классификации. Полученные модели могут быть использованы в различных сферах, включая сетевую безопасность, анализ трафика и контроль доступа к сетевым ресурсам.

В перспективе планируется разработка собственного анализатора сетевых дампов данных. Это позволит упростить интеграцию с существующими системами и повысит качество обучаемых моделей классификаторов, а также за счет автоматизации процессов ускорит процесс сбора датасетов. Ключевым направлением станет дополнительное исследование признакового описания сетевого трафика с целью увеличения точности идентификации программ.

Библиографический список

1. **Юхимук, Р.А.** Анализ протоколов сетевого взаимодействия для повышения надежности, быстродействия и безопасности сети организации / Р.А. Юхимук, С.А. Веревкин // Известия Тульского государственного университета. Технические науки. 2023. № 8. С. 286-296.
2. **Веселков, Е.Н.** Проблемы передачи информации в защищенных сетях // Современные материалы, техника и технология. 2019. С. 83-86.
3. **Пальчевский, Е.В.** Анализ и фильтрация протоколов в UNIX-подобных системах, посредством IPTABLES / Е.В. Пальчевский, А.Р. Халиков // Приоритетные задачи и стратегии развития технических наук. 2016. С. 6-9.
4. **Wang, Z.** The applications of deep learning on traffic identification. BlackHat USA. 2015. Т. 24. №. 11. С. 1-10.
5. **Svoboda, J.** Network traffic analysis with deep packet inspection method. Fac. Informatics Masaryk Univ., no. Master's Thesis. 2014.
6. **Утробин, В.А.** Элементы теории активного восприятия изображений // Труды НГТУ им. Р.Е. Алексеева. 2010. Т. 81. № 2. С. 61-69.
7. **Воронцов, К.В.** Обзор постановок оптимизационных задач машинного обучения // Общероссийский семинар по оптимизации 3 июня 2020 г. [Электронный ресурс] Режим доступа: https://www.mathnet.ru/PresentFiles/27231/voron2020_06_03_opt.pdf
8. **Powers, D.M**.W. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation //arXiv preprint arXiv:2010.16061. – 2020.

Дата поступления
в редакцию: 13.09.2024

Дата принятия
к публикации: 07.10.2024