DOI: 10.15514/ISPRAS–2020–32(6)–11

Обзор методов классификации сетевого трафика с использованием машинного обучения

[1,2] *А.И. Гетьман, ORCID: 0000-0002-6562-9008 <thorin@ispras.ru>*
[1] *М.К. Иконникова, ORCID: 0000-0003-1530-5133 <mikonnikova@ispras.ru>*
[1] *Институт системного программирования им. В.П. Иванникова РАН, 109004, Россия, г. Москва, ул. А. Солженицына, д. 25*
[2] *Национальный исследовательский университет «Высшая школа экономики», Россия, 101000, г. Москва, ул. Мясницкая, д. 20*

Аннотация. В статье рассматривается задача классификации сетевого трафика с использованием методов машинного обучения. Приводятся различные постановки задачи, описываются ограничения использовавшихся ранее методов и причины использования машинного обучения в данной области. Рассматриваются различные алгоритмы машинного обучения, которые могут использоваться для решения задачи, указываются их преимущества и недостатки. Исследуется вопрос отбора признаков для классификации и проблема получения данных для обучения, основные компромиссы в этом вопросе. Перечисляются часто используемые наборы данных и их характеристики. Завершается обзор описанием актуальных проблем в данной области: обучение и сравнение моделей, защита данных пользователей, изменчивость трафика.

Ключевые слова: анализ сетевого трафика; классификация сетевого трафика; машинное обучение

Для цитирования: Гетьман А.И., Иконникова М.К. Обзор методов классификации сетевого трафика с использованием машинного обучения. Труды ИСП РАН, том 32, вып. 6, 2020 г., стр. 137-154. DOI: 10.15514/ISPRAS–2020–32(6)–11

A survey of Network Traffic Classification Methods Using Machine Learning

[1,2] *A.I. Getman, ORCID: 0000-0002-6562-9008 <thorin@ispras.ru>*
[1] *M.K. Ikonnikova, ORCID: 0000-0003-1530-5133 <mikonnikova@ispras.ru>*
[1] *Ivannikov Institute for System Programming of the Russian Academy of Sciences, 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia*
[2] *National Research University Higher School of Economics, 20, Myasnitskaya st., Moscow, 101000, Russia*

Abstract. This survey is dedicated to the task of network traffic classification, particularly to the use of machine learning algorithms in this task. The survey begins with the description of the task, its variations and possible uses in real-world problems. It then proceeds to the description of the methods used historically to solve this task, their limitations and evolution of traffic making machine learning the main way to solve the problem. Then the most popular machine learning algorithms used in this task are described, with the examples of research papers, providing the insight into their advantages and disadvantages in relation to this field. The task of feature selection is discussed, followed by the more global problem of acquiring the suitable dataset to use in the research; some examples of such popular datasets and their descriptions are provided. The paper concludes with the outline of the current problems in this research area to be solved.

Keywords: network traffic analysis; network traffic classification; machine learning

For citation: Getman A.I., Ikonnikova M.K. A survey of network traffic classification methods using machine learning. Trudy ISP RAN/Proc. ISP RAS, vol. 32, issue 6, 2020, pp. 137-154 (in Russian). DOI: 10.15514/ISPRAS–2020–32(6)–11

1. Введение

Классификация трафика является необходимой в наше время, так как полученные результаты могут применяться в различных приложениях, важных как для администрирования сети, так и для конечного пользователя [1, 2].

С точки зрения провайдера определение протоколов/приложений/типов приложений по потокам данных в сети может использоваться для:

- контроля сети и трафика в ней (например, для блокировки отдельных протоколов, таких как BitTorrent),
- обеспечения высокого качества обслуживания клиентов посредством эффективного выделения наиболее приоритетных потоков и регулирования скорости передачи отдельных пакетов,
- регулирования цен на услуги,
- планирования размещения и использования ресурсов,
- оптимизации предоставляемых сервисов и алгоритмов маршрутизации (например, для изменения приоритетов передачи различных типов данных в случае высокой загруженности сети).

Оценка текущего использования сети пользователями может давать понимание об оптимальном устройстве новых сетей с учётом понимания предпочтений и принципов работы интернет-пользователей и интернет-сервисов, так как появляется возможность получать подробную статистику по всем сервисам.

Так как потребности пользователей относительно использования сети постоянно меняются, необходимо их знать и модифицировать Сеть в соответствии с актуальными запросами. Для этого нужно как уметь моделировать устройство сети на текущий момент времени, так и понимать направление движения её развития и изменения. Например, на сегодняшний день видна тенденция отказа от превалирующего ранее принципа асимметрии устройства сети в том смысле, что клиенты загружают намного больше информации, чем отправляют её в Сеть. Появление P2P-приложений, VoIP, видеозвонков, потоковой передачи мультимедиа и прочих новшеств должно вызвать у интернет-провайдеров соответствующие ответные действия по переустройству сети под новые запросы клиентов. Кроме того, в настоящее время увеличивается количество так называемых «умных устройств», которые должны в будущем составить Интернет вещей: он также поставит перед интернет-провайдерами ряд задач для обеспечения максимальной эффективности своей работы.

Отдельно следует упомянуть мобильные приложений, чья доля в интернет-трафике неуклонно растёт. Использование смартфонов и мобильных приложений можно считать более персонализированным, поэтому получение данных о такого рода трафике позволяет эффективно составлять сетевой портрет пользователя. Определение интересов пользователей может служить целям маркетинга, позволяя проводить лучше таргетированные рекламные кампании.

С точки зрения безопасности информационных систем, классификация интернет потоков может использоваться как важный признак при выявлении кибер-атак, аномалий в работе Сети, неправомерных или необычных действий пользователя и прочих нарушений, что способно повысить общую безопасность Сети.

Методы, применяемые для классификации интернет-трафика, меняются вместе с глобальными изменениями в устройстве трафика. Внедрение новых технологий начинает негативно влиять на качество работы ранее применимых способов, что приводит к

необходимости создания и развития новых походов. К таким глобальным изменениям, влияющим на решение задачи классификации трафика можно отнести:

- отказ от использования утверждённого списка портов в зависимости от протокола/приложения (намеренный или в связи с устареванием данного списка);
- обфускация протоколов с целью замаскировать те из них, которые блокируются/подавляются провайдером;
- всё более широкое распространение шифрования трафика, не позволяющее использовать для классификации содержимое полезной нагрузки пакета;
- постоянное появление новых протоколов и приложений и т.д.

По приведённым выше причинам, задачу классификации интернет трафика на сегодняшний день нельзя считать решённой, и исследовательские группы продолжают предлагать всё новые решения, позволяющие показывать эффективные результаты в условиях меняющейся реальности.

2. Эволюция методов классификации трафика

Методы классификации сетевого трафика с годами развиваются и модифицируются. Это связано в первую очередь с предъявляемыми сетью требованиями и ограничениями. Изменение устройства сетевого трафика и особенностей его передачи приводит к тому, что старые методы классификации становятся малоэффективными или просто непригодными. С другой стороны, развитие методов классификации и оборудования, на котором может работать система позволяет использовать больше признаков и более развитые способы их применения для принятия решения.

К важным характеристикам методов классификации сетевого трафика относятся:

- *детализация:* с каким уровнем точности система производит классификацию: семейство протоколов/класс приложений или конкретные протоколы, конкретные приложения.
- *скорость реакции:* способна ли система производить классификацию быстро (после нескольких пакетов), что подходит для анализа в реальном времени или для классификации нужны данные о потоке полностью.
- *вычислительная стоимость:* сложность вычислений и затраты по использованию памяти для классификации пакета или потока.

2.1 Классификация по номерам портов

Первые системы классификации трафика основывались на извлечении из пакетов номеров портов и сопоставлении их со списком IANA (Internet Assigned Numbers Authority, «Администрация адресного пространства Интернет»). IANA выделяет и регистрирует номера портов, используемые для конкретных специфических целей, например, под протокол HTTP выделен порт 80. Информацию о протоколе можно уже использовать для примерного определения типа деятельности пользователя. Этот метод классификации работает очень быстро и не требует хранения данных о потоке, вычислительно прост. Это позволяет, например, удобно использовать его в межсетевых экранах для фильтрации трафика. Однако, он обладает рядом существенных недостатков, которые по мере эволюции устройства Сети негативно влияет на результаты его работы.

Номер порта определён не для всех протоколов. В списке IANA уже содержатся категории «известно несколько применений наряду с зарегистрированным» и «порт не зарегистрирован IANA». Некоторые протоколы выбирают порты для обмена данными в ходе своей работы случайным образом (как FTP). Вдобавок, некоторые протоколы могут использовать известные номера портов других протоколов, чтобы замаскироваться под них, если другой протокол является более предпочтительным с точки зрения интернет-провайдера. Например,

протокол BitTorrent может таким образом маскироваться под HTTP, чтобы избегать блокировок или ограничений на скорость передачи данных. Появляющиеся в последнее время протоколы также могут не успевать получить зарезервированный за собой порт.

Этот метод хорошо подходит для определения протоколов, однако не способен хорошо различать приложения. Например, браузинг веб-страниц, VoIP и просмотр видео – все будут использовать 80 (HTTP) или 443 (HTTPS) порт для своей работы. Но у этих приложений абсолютно разные сценарии использования, поэтому на практике нам хотелось бы их различать.

Широкое распространение технологий туннелирования, инкапсулирующих протоколы, шифрование на уровне IP, использование NAT (Network Address Translation, преобразование сетевых адресов) и NAPT (Network Address and Port Translation, трансляция сетевых адресов и портов) – всё это влияет на применимость данного метода. Поэтому точность систем, основанных на определении номеров портов, невысока (по разным оценкам, от 30 до 70%) и продолжает ухудшаться. В настоящее время этот признак может служить лишь одним из многих, выступая как источник дополнительной информации при принятии решения, основанного на других критериях.

2.2 Глубокий анализ пакетов

Следующим шагом развития классификаторов интернет трафика стало использование технологии DPI (Deep Packet Inspection, глубокий анализ пакетов). Фильтрация сетевых пакетов в этом случае проводится по их полному содержимому, то есть проводится анализ не только заголовков, но и всего трафика на уровнях модели OSI со второго и выше. Этот метод показывает высокую точность работы, а полученная с его помощью разметка зачастую принимается как эталонная для данных с неизвестными классами. Для классификации с помощью DPI создаётся библиотека сигнатур и шаблонов пакетов, и для каждого пакета производится поиск соответствий в этой библиотеке.

Было замечено, что некоторые проприетарные протоколы передают информацию на уровне битов, что привело к созданию инструментов, работающих и на этом уровне. Генерируемые маски содержат значения 0, 1 и *[3] или вероятность единицы в данном бите [4].

При всех своих достоинствах метод DPI сталкивается с существенными проблемами в своей работе. Среди главных – невозможность работы с зашифрованным трафиком, доля которого в Интернете растёт с каждым годом, и высокие требования к ресурсам. Для хранения данных пакетов и библиотеки сигнатур требуется достаточно большой объём памяти, а при росте количества известных классов растёт размер этой библиотеки и, соответственно, время на поиск соответствий в ней. Поэтому, этот метод плохо подходит для работы в высокоскоростных сетях в режиме реального времени. Кроме того, определённую сложность представляет создание и поддержание в актуальном состоянии библиотеки сигнатур при всё увеличивающемся количестве протоколов и приложений в Сети.

Отдельно стоит вопрос защиты приватности пользователей Сети – проблема, которая актуальна для всех систем, использующих в своей работе полезную нагрузку пакетов. Законодательную сторону этого аспекта нужно учитывать при создании, обучении и работе систем глубокого анализа пакетов. Некоторые методы пытаются ограничить количество используемых данных пакета, например, первыми 40 битами [4], но полностью проблему это не решает.

2.3 Стохастический анализ пакетов

Стохастический анализ пакетов (SPI, Stochastic packet inspection) для классификации пакетов изучает статистические свойств их содержимого. Например, в [5] используется критерий Хи-квадрат Пирсона для изучения случайности распределения первых байтов полезной нагрузки

пакета. Таким образом строится модель синтаксиса протокола, используемого приложением. В [6] потоки определяются как зашифрованные или незашифрованные на основании энтропии первого пакета. В [7] вычисление энтропии первых байтов полезной нагрузки идентифицирует тип содержимого как текст, бинарный файл или шифрованный файл, что позволяет приоритизировать передачу некоторых файлов. Однако, такую классификацию сложно назвать точной или детализированной, так как для одного и того же приложения возможно использование всех видов содержимого. Кроме того, хотя стохастический анализ и использует более простые операции, чем глубокий анализ пакетов, он всё равно использует большой объём памяти для анализа. В связи с этим, данный метод не получил широкого распространения.

2.4 Использование машинного обучения для классификации трафика

Тенденции изменения сетевого трафика, широкое распространение шифрования, рост скорости передачи данных, а соответственно и необходимой скорости их обработки, постоянное появление новых классов трафика - всё это потребовало появления новых способов его классификации. Для этого были предложены методы машинного обучения, которые позволяют во многом упростить работу с созданием наборов различающих характеристик классов, автоматизируя этот процесс на основе анализа большого количества примеров этих классов (собрать который значительно проще, чем проанализировать вручную). Кроме того, многие из предложенных методов работают с общими признаками потоков, а не с полезной нагрузкой пакетов, что решает проблемы, связанные с шифрованием и с защитой данных пользователей. Это же даёт преимущество в скорости классификации и уменьшает необходимый для принятия решения объём памяти.

Далее будут подробно рассмотрены именно методы машинного обучения для классификации сетевого трафика: типы классификации, используемые модели и признаки, а также наборы данных, на которых производится обучение и тестирование моделей.

3. Типы классификации

Трафик в сети можно классифицировать отдельными пакетами, и методы классификации на основе портов и DPI способны решать эту задачу, однако сейчас в большинстве работ классификация производится для потоков. Здесь и далее, поток – это пятёрка значений:

<IP-адрес источника, IP-адрес получателя, порт источника, порт получателя, тип транспортного протокола>.

Существуют подходы, в которых запросы отправителя и ответы получателя интерпретируются как два разных встречных потока, однако более частым решением является объединение этих потоков в единый двунаправленный. Поскольку интернет-поток как правило подразумевает один законченный сеанс взаимодействия между отправителем и получателем (клиентом и сервером), проблем с классификацией всего потока как единого целого в один класс обычно не возникает.

Так как потоки различаются по своей продолжительности и количеству передаваемых данных, среди них иногда особо выделяют самые маленькие («мышиные», mice) потоки и самые большие («слоновьи», elephant). Из-за существенного отличия в объёме этих потоков результаты классификации иногда проверяются отдельно по доле правильно классифицированных потоков и по доле классифицированных байтов. Большое значение имеет также длина потока. Потоки или обрывки потоков, состоящие из малого количества пакетов, могут не нести достаточного количества информации для определения класса, поэтому нуждаются в специальном подходе или игнорируются.

Классификация трафика может проводиться в онлайн режиме, то есть в режиме реального времени, или офлайн, постфактум. Режим классификации определяется решаемой задачей.

Так, например, сбор статистики использования сети для глобального перераспределения ресурсов, получение информации об активности пользователя для выставления ему счетов за интернет-услуги и перерасчёт этих цен - всё это не требует срочного ответа и может обрабатываться в свободном режиме, с любым количеством доступных данных и ресурсов. Другие же задачи, такие как обеспечение качества сервиса для пользователя, выявление атак и угроз, оперативное перераспределение ресурсов, требуют как можно более быстрой реакции. В этих случаях классификатор не имеет возможности ждать окончания потока и оперировать полной информацией о нём, а вынужден ограничиваться лишь частью информации, например, лишь первыми N пакетами потока. Также накладываются ограничения на используемую модель классификатора, так как она должна достаточно экономно расходовать ресурсы системы и максимально оперативно принимать решение о потоке. Кроме того, в этих случаях решение обычно принимается параллельно для нескольких/многих потоков данных сразу, поэтому ограничения накладываются и на количество оперативной памяти, выделяемой для обработки потока.

Выбор набора классов для проведения классификации трафика зависит от решаемой задачи. В качестве примеров можно привести следующее.

1) Классификация по протоколам прикладного уровня (HTTP, SMTP, SSH и т.д.) [8, 9]. Обычно выбираются именно протоколы прикладного уровня, так как такая классификация является наиболее практически ценной.
2) Классификация по приложениям, генерирующим интернет-трафик (Skype, Torrent, браузер и т.д.) [10, 11]. Это определяет активность пользователя и позволяет строить его профиль, ограничивать деятельность конкретных приложений, решать маркетинговые задачи.
3) Классификацию по типам действий пользователя (интернет-браузинг, скачивание файлов, просмотр видео и т.п.) [12, 13]. В данном случае определяется не конкретное используемое приложение, а вид деятельности пользователя. В некотором смысле это обобщение предыдущего типа классификации.

Существуют и другие [14, 15] подходы к определению набора классов для классификации интернет-трафика, но приведённые три являются наиболее популярным, и именно они чаще всего исследуются и описываются в научных статьях.

Отдельно здесь следует упомянуть классификацию мобильного трафика [16] и классификацию трафика устройств интернета вещей [14], которые обладают особенностями, выделяющими их в отдельные задачи. Это и большая популярность новых и/или специализированных протоколов, и другие сценарии работы приложений, и другие масштабы распространения.

4. Методы машинного обучения, используемые для классификации трафика

Задача классификации сетевого трафика, как и другие задачи классификации, обычно рассматривается как задача обучения с учителем, поэтому при её решении используются соответствующие методы машинного обучения. Среди них можно выделить:

- наивный байесовский классификатор;
- метод опорных векторов;
- метод k-ближайших соседей;
- деревья принятия решений (с разными алгоритмами построения дерева: CART, C4.5, C5.0);
- методы бэггинга (случайный лес);
- методы бустинга (Adaboost, XGBoost);

- разные виды нейронных сетей: CNN, CNN+RNN, CNN+LSTM, SAE.

Рассмотрим примеры и результаты применения вышеупомянутых методов в исследованиях по теме классификации сетевого трафика.

4.1 Наивный байесовский классификатор

Наивный байесовский классификатор – простой вероятностный классификатор, основанный на применении Теоремы Байеса со строгими (наивными) предположениями о независимости. По теореме Байеса,

$$P(A|B) = \frac{P(B|A)*P(A)}{P(B)},$$

где A – класс, B – признак. Для предсказания неизвестного класса вычисляется его апостериорная вероятность. Все признаки считаются независимыми, вероятности напрямую вычисляются из обучающих данных (дискретные значения) или оцениваются через нормальное распределение.

Наивный байесовский классификатор является одним из самых простых методов машинного обучения, к его достоинствам можно отнести высокую скорость обучения и работы, однако он зачастую показывает результаты хуже, чем другие методы, поэтому его применение на практике ограничено. Можно упомянуть пример статьи [11], в котором он проигрывает в сравнении с другими рассмотренными классификаторами при классификации трафика, но показывает хорошие результаты как способ обобщения предсказаний разных типов классификаторов (см. подразд. 4.5, бэггинг).

4.2 Метод опорных векторов

В методе опорных векторов (SVM, support vector machine) каждый объект данных представляет из себя точку в p-мерном пространстве (где p - количество признаков), и алгоритм пытается построить гиперплоскость размерности (p−1), максимально эффективно разделяющую точки, относящиеся к разным классам. Таким образом возможна классификация на два класса, для проведения многоклассовой классификации на N классов посредством метода опорных векторов могут использоваться техники:

- попарного сравнения: строится N*(N-1)/2 классификаторов, каждый из которых учится различать между собой два класса;
- один против всех: строится N классификаторов, каждый из которых учится отличать один класс от всех остальных.

К достоинствам метода опорных векторов можно отнести его эффективность в многомерных пространствах признаков. Однако, этот метод достаточно затратен по памяти и вычислительной сложности и может легко переобучаться, то есть подстраиваться только под параметры конкретного множества примеров, на которых он обучался. В то же время его результаты для задачи классификации сетевого трафика хуже, чем у большинства других классификаторов [11], поэтому в настоящее время он редко используется с этой целью.

4.3 Метод k-ближайших соседей

Метод k-ближайших соседей присваивает каждому классифицируемому примеру то значение класса, которое наиболее распространено среди его k ближайших согласно выбранной функции расстояния соседей, классы которых известны. В качестве функции расстояния часто выбирается евклидова метрика. Такой подход не даёт высокую точность сам по себе, но может применяться в сочетании с кластеризацией, как в [17] для улучшения результатов выделения из трафика отдельных неизвестных классов.

4.4 Дерево принятия решений

Дерево принятия решений (decision tree) представляет собой бинарное дерево, в вершинах которого записаны атрибуты (признаки), по которым различаются различные ситуации, на рёбрах – значения этих атрибутов, а в листьях – значения целевой функции. Чтобы классифицировать новый случай, надо спуститься по дереву до листа и выдать соответствующее значение.

Для построения дерева могут применяться разные алгоритмы, в работах по классификации сетевого трафика можно найти примеры использования:

- C4.5 [8, 18-20] – улучшенная версия одного из базовых алгоритмов построения деревьев ID3 (Iterative Dichotomiser 3), которая может использовать как дискретные, так и непрерывные признаки, позволяет задавать веса для признаков, и производит прореживание (pruning) для построенных деревьев с целью их оптимизации. Для выбора признака для разбиения используется информационный выигрыш, основанный на энтропии.
- C5.0 [10, 21] – оптимизация алгоритма C4.0, дающая преимущество по скорости работы и используемой памяти, строящая деревья, сравнимые по эффективности, но меньшего размера.
- CART [22] (Classification and Regression Trees) строит бинарное дерево решений с использованием критерия Джини.

Деревья принятия решений и решения, построенные на их основе, являются одним из самых популярных способов решения задачи классификации трафика, так как среди их достоинств можно перечислить:

- отсутствие необходимости в специальной подготовке данных (нормализация, добавление фиктивных переменных для приведения примеров к единому размеру и др.) как при работе с нейронными сетями (см. [23]),
- способность работы с разными типами переменных (как категориальными, так и интервальными),
- способность работать с большими объёмами данных,
- хорошие результаты классификации [11, 18],
- высокую скорость работы и низкую вычислительную сложность предсказания результата построенным деревом [18],
- простоту мультиклассовой классификации (в сравнении, например, с SVM).

К недостаткам этого метода можно отнести:

- неоптимальность алгоритмов построения дерева (проблема получения оптимального дерева решений является NP-полной, поэтому оптимальное решение выбирается локально в каждом узле, что влияет на оптимальность дерева в целом),
- риск переобучения (необходимо регулировать глубину дерева),
- сложность выражений некоторых ситуаций посредством дерева (например, XOR),
- неустойчивость деревьев при небольших изменениях входных данных.

С этими недостатками можно бороться применением алгоритмов бэггинга и бустинга, которые рассматриваются далее.

4.5 Бэггинг

Бэггинг (bagging от bootstrap aggregating) – это способ композиции нескольких более простых классификаторов в один с целью повысить его стабильность и точность. В частности, широко используется алгоритм *случайный лес* (Random Forest) [9, 11, 13, 24, 25], заключающийся в использовании комитета (ансамбля) решающих деревьев. Основная идея этого метода в

использовании большого ансамбля деревьев, каждое из которых само по себе даёт очень невысокое качество классификации, но за счёт их большого количества результат получается хорошим. Классификация объектов проводится путём голосования: каждое дерево комитета относит классифицируемый объект к одному из классов, и побеждает класс, за который проголосовало наибольшее число деревьев.

В приведённых статьях метод бэггинга выбирается как основной для классификации или показывает одни из лучших результатов в сравнении с другими методами. Получающиеся модели сложнее, чем для рассмотренных выше классификаторов, а время на классификацию одного примера растёт примерно линейно относительно количества простых классификаторов в ансамбле, однако именно этот метод зачастую даёт наилучший вариант компромисса между качеством и скоростью классификации.

4.6 Бустинг

Бустинг (boosting) – способ организации классификаторов, позволяющий упорядочить и обучить более простые классификаторы (чаще всего это решающие деревья небольшой глубины) таким образом, чтобы результирующий классификатор показывал лучшие результаты.

Примерами алгоритмов бустинга являются:

- AdaBoost (Adaptive Boosting) [13, 18, 26] – регулирует веса в процессе обучения, чтобы примерам, на которых ошибся прошлый классификатор, придавалось большее значение;
- XGBoost [24] – ошибка минимизируется алгоритмом градиентного спуска.

По качеству и скорости работы бустинг примерно сравним с бэггингом (зависит от конкретных данных; один из примеров такого сравнения можно найти в [13]).

Можно видеть, что из рассмотренных на данный момент моделей машинного обучения для классификации сетевого трафика чаще всего и с наибольшим успехом используются деревья принятия решения (с разными алгоритмами построения), а также их комбинации методами бэггинга или бустинга.

4.7 Нейронные сети

Искусственные нейронные сети состоят из нескольких слоёв искусственных нейронов, каждый из которых получает на вход несколько числовых значений и преобразует их в выходное значение в соответствии со своими внутренними правилами (заранее заданной функцией активации и вычисляемыми весами входных параметров). Слой сети может состоять из произвольного числа нейронов, каждый из которых может быть соединён с любыми нейронами из предыдущего и последующего слоёв.

Существуют модели так называемых рекуррентных нейронных сетей с обратной связью, когда сигнал с выходных нейронов или нейронов скрытого слоя частично передается обратно на входы нейронов входного слоя, но чаще всего нейронные сети являются сетями прямого распространения (feedforward). Значения нейронов в каждом слое вычисляются на основании значений предыдущего слоя в ходе процесса, называющегося алгоритмом прямого распространения (forward propagation). В процессе обучения изменение параметров направлено на минимизацию функции штрафа (cost function). Для коррекции весов в процессе обучения применяется алгоритм обратного распространения ошибки (backpropagation).

Для работы с большим количеством параметров во внутренних слоях сети или для определения инвариантных относительно переноса признаков свою эффективность показали свёрточные нейронные сети (CNN), использующие набор небольших ядер для преобразования поступающей информации и уменьшения числа извлекаемых признаков [12, 23, 27-30].

Рекуррентные сети (RNN), как уже упоминалось ранее, могут использоваться для работы с признаками, имеющими зависимости во времени (например, в случае, когда текущий ответ сети должен зависеть не только от текущих признаков, но и от некоторых признаков, использовавшихся при принятии предыдущего решения). Для этого сеть особым образом сохраняет часть извлекаемой информации для дальнейшего использования. Самым часто используемым представителем является LSTM (Долгая краткосрочная память от англ. Long short-term memory) [27, 28].

Автокодировщики (Autoencoders) представляют из себя нейронные сети, состоящие из двух частей: первая половина сети учится кодировать входную информацию в сжатом виде, а вторая учится с максимально возможной точностью воссоздавать эту информацию по сжатому представлению. Поскольку получаемый метод кодирования эффективно работает только для того типа данных, на котором обучалась сеть, другие данные будут воссоздаваться с ошибкой, что позволяет использовать автокодировщики для классификации [12, 31, 32].

Генеративно-состязательная сеть (GAN, Generative adversarial network) - комбинация из двух нейронных сетей, одна из которых учится генерировать образцы данных, похожие на реальные, а вторая - отличать эти образцы. В процессе совместного обучения качество работы обеих этих сетей растёт, что позволяет генерировать искусственные данные, почти неотличимые от реальных. Такой подход может использоваться для генерации дополнительных примеров при невозможности получить их достаточное количество другим путём [9].

Также существуют примеры совместного использования нескольких типов нейронных сетей для получения лучших результатов, в частности, CNN+RNN [14, 27, 33].

5. Признаки сетевого трафика, используемые для его классификации

Алгоритмы машинного обучения для своей работы нуждаются в получении признаков классифицируемых примеров, на основе которых будет приниматься решение. Для задачи классификации сетевого трафика можно выделить два основных способа их получения:

- признаки выделяются на специальном этапе подготовки данных на основе некоторой внешней информации (например, знания эксперта в области); такие признаки могут включать в себя как содержимое пакетов, так и дополнительную информацию о потоке (такую, как общее количество пакетов, размер переданных данных, время получения пакетов и т.п.);
- признаки выделяются из данных (обычно это содержимое пакетов) самой моделью в процессе так называемого глубокого обучения.

Первый подход требует от экспериментатора дополнительных усилий, но позволяет лучше контролировать процесс и использовать любую доступную информацию и её производные (например, посчитанные на основе имеющихся данных статистики). Второй подход требует минимальных действий при подготовке данных (нормализовать данные, унифицировать длину примеров, переупорядочить данные, если это необходимо) и может позволить находить неочевидные на первый взгляд зависимости, но извлекаемый им набор признаков не всегда является лучшим или минимальным для решения поставленной задачи. Кроме того, второй подход обычно требует для обучения большего количества данных.

По содержанию признаков их можно несколько условно разбить на следующие классы:

- данные пакета;
- метаинформация о пакете;
- временные характеристики;
- информация о потоке.

Первая категория включает в себя содержимое полезной нагрузки пакета или все байты пакета без из разделения на заголовок и данные. Именно она чаще всего используется при

глубоком обучении модели классификации [12, 23, 27]. Выделяемые из содержимого пакетов закономерности очень информативны (такой подход является в некотором смысле автоматизированным аналогом DPI). В [12] показано, что такой метод может применяться и для классификации зашифрованного трафика, однако количество информации о пакете, используемой для его классификации достаточно велико (1480 байтов). В [27] поднимается вопрос о возможности переобучения сети под искусственно сгенерированный набор данных: при таком наборе признаков модель способна запоминать IP-адреса или номера портов, которые используются для разных классов. В таком случае показанные ей в экспериментах результаты будут высокими, но такая модель не будет применима для реальных сетей. Эта возможность не учтена в [23], где используются 784 первых байтов пакета, включая все уровни заголовка.

Под метаинформацией о пакете можно понимать такие признаки, как размер пакета, размер его полезной нагрузки, направление движения, а также тип сервиса, указанный тип протокола, установленные флаги и размер окна (для TCP). Различные комбинации этих признаков используются во многих работах по теме.

Временные характеристики включают в себя интервалы времен между прибытием пакетов (этот признак весьма показателен для классификации трафика по разным сценариям использования), общее время сессии. Также, в статье [21] предложена идея использования временных вспышек трафика. Временная вспышка (burst) трафика – это группа последовательных пакетов с интервалами прибытия между ними меньше, чем между разными вспышками. Характеристики вспышек в потоке трафика (burstiness) являются мерой изменчивости распределения времён прибытия пакетов. В новом варианте исследований [10] от тех же авторов был добавлен ещё такой признак, как время бездействия потока.

Информация о потоке объединяет в себе метаинформацию всех или части пакетов (в целом или в каждом направлении отдельно) в потоке в виде некоторых её статистических характеристик: сумма по всех пакетам, минимум, максимум, среднее арифметическое, медиана, дисперсия и т.д. Также, могут вычисляться аналогичные характеристики для временного распределения пакетов. Эти признаки дополняют и расширяют информацию, получаемую из двух предыдущих категорий признаков, но требуют для своего вычисления завершения потока, что затрудняет их использование в режиме классификации в реальном времени. Для решений этой проблемы можно использовать не все, а только N первых пакетов в потоке (например, [23]. Значение N подлежит экспериментальному вычислению или выбирается на основе экспертной оценки.

Итак, разные исследования задачи классификации трафика используют разные категории признаков, одну или несколько из перечисленных выше. Задача выбора и отбора признаков весьма актуальна и решается практически в каждом новом исследовании. Существуют статьи [34] и инструменты [35], посвящённые описанию и получения максимально возможного количества доступных признаков для пакетов в потоке данных.

6. Выбор и подготовка набора данных, их сравнение

Препятствием для прямого сравнения различных подходов в работах исследовательских групп по классификации трафика является не только наличие нескольких разных принципов классификации (по протоколам/приложениям/типам приложений/...), но и отсутствие единого общепризнанного набора данных, на которых бы проводилось тестирование предлагаемых методов. На настоящий момент отсутствие такого стандартного набора означает, что каждая исследовательская группа самостоятельно находит/собирает данные для обучения и тестирования моделей, размечает их по своей системе своими силами и способами и публикует полученные на них результаты. Это не позволяет сопоставлять эти результаты между собой напрямую, как делается при решении многих типовых задач, что является существенным недостатком в данной области.

Получение большой и репрезентативной выборки данных для обучения и тестирования моделей – первая проблема, которая возникает при решении задачи классификации данных. Каждая исследовательская группа должна найти для себя подходящий источник данных, убедиться в корректности его разметки соответственно поставленной задаче и оценить его полноту относительно возможных вариантов организации трафика в сети. Количество существующих приложений/протоколов огромно, поэтому производить полную классификацию вряд ли представляется возможным. Обычно исследователи выбирают лишь некоторый набор из возможных вариантов (наиболее частые/характерные/представляющие наибольший интерес) и работают только с ним. Однако в таком случае возникает проблема обработки данных, не входящих в данный набор классов. В некоторых работах эти данные просто не рассматриваются, но этот подход неизбежно сталкивается с проблемами при работе с данными реального мира. Ещё одной проблемой является частое появление новых приложений/протоколов, которые либо должны своевременно отражаться в данных, либо обрабатываться особым образом. Также, стоит учитывать, что от метода сбора данных зависит не только их репрезентативность, но и распределение вероятностей классов, что также может влиять на конечный результат.

6.1 Основные аспекты выбора набора данных

При выборе или получении данных особое внимание следует уделять следующим аспектам.

6.1.1 Получение правильной разметки данных

При применении систем, использующих методы машинного обучения для классификации сетевого трафика, помимо тестовой выборки для оценки результатов нужна достаточно большая тренировочная выборка с правильно размеченными ответами. Соответственно, возникает проблема получения этой разметки. Источников здесь может быть несколько.

a) *Сторонние системы классификации.* Например, можно использовать методы, разбирающие и анализирующие всю информацию, содержащуюся в пакете, для построения более легковесных и оперативных классификаторов. Под это определение подходят системы DPI, такие как Wireshark [36], nDPI [37] и др. Сравнение некоторых доступных для использования систем DPI произведено в [38, 39]. Для проверки правильности разметки инструментов предлагается протестировать их работу на множестве данных, для которых специальная программа регистрирует генерирующие их приложения, что позволяет достичь заведомо высокого качества разметки. Исследование показывает, что использование инструментов DPI даёт высокую, но не идеальную точность при определении приложений.

   Соответственно, при этом подходе, следует учитывать, что точность такой разметки, а соответственно, и обучаемых на ней алгоритмов, ограничена точностью этих сторонних методов. Кроме того, этот способ может быть неприменим при работе с зашифрованным трафиком.

b) *Получение данных в контролируемых условиях.* В этом случае нужно организовать сбор информации как можно ближе к пользователю и поставить перед ним задачу генерировать только заранее определённый трафик. Самая распространённая проблема в этом случае – фильтрация фонового трафика, доля которого может доходить до 70%.

   Для контроля за получением трафика можно либо собирать его в процессе исполнения специальных скриптов, генерирующих только определённые его классы (ISCX, NIMS), либо запускать параллельно со сбором трафика специальные программы, позволяющие определить его источник (UPC dataset).

c) *Генерация искусственных данных на основе существующих.* Ещё один способ получения дополнительных данных для тех классов, в которых этих данных недостаточно для обучения модели, – это дополнение тренировочного множества искусственными

данными. Например, в [9, 28] для этой цели используется LSTM. Как показано, такой метод позволяет улучшать качество построенной модели, особенно в тех случаях, когда некоторые из классов недостаточно представлены в тренировочном множестве. Однако, для его использования нужен хорошо обученный генератор данных, что является отдельной задачей, которую также надо решать. Кроме того, нужно следить, чтобы получаемые таким образом данные не были излишне однообразными и соответствовали реальному положению дел в сети.

6.1.2 Доступность данных

Из-за шифрования данных или в связи с необходимостью соблюдения конфиденциальности пользователей, часть данных в пакетах может быть искажена или недоступна. Например, при публикации снимков сетевых трасс может производиться специальное кодирование IP адресов или удаление полезной нагрузки пакетов. Подобные действия могут создавать трудности, особенно при разметке данных, однако этический вопрос в данном случае должен иметь преимущество. Одним из способов поиска компромисса в данной ситуации является сбор статистики поведения реальных пользователей сети интернет и написание специальных алгоритмов-ботов, которые будут имитировать это поведение с максимальной правдоподобностью, при этом не выдавая никакую конфиденциальную информацию.

Такой подход к решению проблемы получил достаточно широкое распространение и позволил получить содержательные датасеты для многих задач, связанных с анализом трафика в сети. Одним из его существенных недостатков является, однако, большая упорядоченность и детерминизм действий по сравнению с реальными пользователями, что может нарушать естественное распределение статистических признаков потоков данных.

6.1.3 Место получения данных

Получаемые для анализа потоки данных зависят от места их сбора. Получение данных как можно ближе к конечным пользователям позволяет проще решать вопрос с шифрованием данных, в то время как сбор информации у провайдера, на глобальных маршрутизаторах позволяет получить больший объём разнообразных и репрезентативных данных. В то же время, такие особенности, как изменение временного распределения в потоке, изменение длины пакетов при туннелировании, наличие на маршрутизаторе, служащем точкой сбора информации, только одного направления потока данных, - все они могут приводить к тому, что модель, обученная на данных из одного источника, может не работать на данных из другого места. Поэтому, предпочтительной стратегией в данном случае является использование в качестве тренировочных данных, получаемых из тех же точек в сети, где предполагается работа создаваемой системы.

В [40] исследуется влияние особенностей характеристик сетей на работу классификатора, в частности показано падение результатов работы классификатора при его тестировании на трафике из сети, отличной от той, на которой он обучался.

6.1.4 Репрезентативность набора данных

После выбора системы используемых классов и разметки данных требуется убедиться в том, что полученный набор данных является репрезентативным, то есть содержит достаточное количество разнообразных примеров для каждого класса (по возможности, покрывает все возможные ситуации). Следует нивелировать перекос по количеству данных для разных классов для предотвращения переобучения классификатора. Некоторые источники получения данных требуют особого внимания и предобработки на данном этапе. Например, если набор данных получен только от нескольких пользователей, то нужно постараться убедиться, что модель сможет выделить из данных глобальные признаки, а не будет пытаться определить специфику работы каждого конкретного пользователя.

6.2 Используемые общедоступные наборы данных

Поскольку сам процесс получения большого количества релевантных данных, пусть и неразмеченных, может быть затруднителен, в исследованиях нередко используются широко доступные трассы сетевого трафика, которые можно найти в интернете. Обычно такие трассы содержат достаточно большой объём данных и различаются по месту и способу их получения, а также предоставленной разметке (если она есть) К наиболее популярным из общедоступных наборов данных сетевого трафика можно отнести следующие.

- *MooreSet* (2007) [18]. Около 59 Гигабайтов данных. Данные были собраны на границе сети университетского кампуса в течение 8 месяцев. Набор данных содержит только потоки TCP. Большая их часть классифицирована вручную на основе содержания на 10 классов, также присутствуют некоторые фоновые потоки и потоки без начала. Классы: web-browsing, mail, bulk, attack, p2p, database, multimedia, service, interactive, games.
- *UPC* (Политехнический университет Каталонии) dataset (2013) [38]. Около 36 Гб данных, собранных за 66 дней. Ради возможности публикации данных с полной полезной нагрузкой трафик был сгенерирован искусственно, авторы постарались максимально имитировать реальный трафик. Для 91% пакетов и 42% потоков присутствует название приложения, полученное с помощью специальной программы, записывающей информацию о каждом сетевом потоке, включая название приложения.
- *CAIDA* (2008-...). Содержит набор анонимизированных трасс, собранных на мониторах магистральных сетей связи. С 2008 по 2014 год в набор добавлялась одна часовая трасса в месяц, с 2014 года – раз в три месяца. Из пакетов удалена полезная нагрузка, IP-адреса зашифрованы инструментом Crypto-PAn (Cryptography-based Prefix-preserving Anonymization) [41]. Размер каждой трассы – несколько миллиардов пакетов.
- *ISCX* (2016). Содержит несколько трасс с искусственно сгенерированными данными для решения разных задач сетевого трафика (поиск аномального поведения, поиск ботнетов, классификация трафика и т.д.). Данные сгенерированы так, чтобы по возможности имитировать поведение реальных пользователей. Наиболее популярным набором данных в исследованиях по классификации трафика является VPN-nonVPN (ISCXVPN2016) [42]. Эта трасса содержит 7 категорий трафика: браузинг, электронная почта, чаты (мгновенные сообщения), стриминг, передача файлов, VoIP, p2p. При этом каждая категория представлена в двух видах - через VPN и без него. Полезная нагрузка пакетов сохранена, общий объём файлов составляет 28Гб.
- *NIMS* (2007). Этот набор данных был собран в специально собранной модели сети посредством прописанных сценариев её использования. Основной целью сбора трафика являлось получение SSH трафика, а в качестве фонового были собраны DNS, HTTP, FTP, P2P (limewire) и telnet. Всего датасет содержит около 700 000 потоков, из них около 35000 – SSH потоков.

Таким образом, единого набора данных для решения задач классификации трафика не существует, и каждая исследовательская группа должна найти или подготовить данные в соответствии со своими целями. При принятии решения о выборе данных для обучения и тестирования классификатора нужно учитывать приведённые выше соображения и ограничения, чтобы полученная модель соответствовала построенной задаче и была способна работать в реальных условиях.

7. Заключение: актуальные проблемы

7.1 Создание общедоступных датасетов и защита данных пользователей

Поскольку используемые и анализируемые данные, составляющие полезную нагрузку пакетов, являются приватной информацией пользователей интернета, работы в этой области должны придерживаться определённых моральных норм. Защита личной жизни пользователей и конфиденциальность их информации должна являться главным приоритетом для исследователей. Одним из аспектов этого вопроса является процесс получения данных, в частности данных для обучения моделей машинного обучения, которые заведомо должны содержать больший объём доступной для исследования информации, так как для них требуется правильно определить их принадлежность к определённым классам. Хранение и обработка этих данных должны обеспечивать безопасность личных данных пользователей и их анонимность.

Однако при этом, как и в других областях, в которых активно развиваются методы решения на основе машинного обучения, желательны создание и поддержка в актуальном состоянии достаточно больших и универсальных наборов правильно размеченных данных, которые могли бы использоваться для проверки работы и сравнения качества разных предлагаемых методов решения задач. Общепринятого компромисса о том, как организовать такой набор с соблюдением защиты информации пользователей, до сих пор нет, хотя и делались попытки предложить варианты решения этой проблемы.

Одним из предложенных решений является анонимизация трафика; как частный её пример – эквивалентная замена адресов во всех используемых пакетах [41]. Так как сами IP-адреса редко имеют какую-либо ценность при классификации трафика, предлагается шифровать их таким образом, чтобы каждому значению однозначно сопоставлялся IP-адрес, но при этом это адрес невозможно было бы восстановить по получаемому значению. Такой подход позволяет защитить конфиденциальность пользователей, не позволяя сопоставить наборы потоков данных с конкретными людьми по их IP-адресам. Но он не решает проблему в случае, если по сети передаются чувствительные данные, такие как пароли, номера кредитных карт и т.п.

В некоторых общедоступных наборах данных с этим борются удалением всей полезной нагрузки из пакета (оставляя только заголовки пакета). Этот способ плох тем, что затрудняет или даже делает невозможной предварительную разметку данных (недостаточно информации для систем DPI). Некоторые данные выкладываются параллельно с такой разметкой, полученной авторами датасета до модификации данных или даже в процессе их получения [38]. Но приведённая разметка может использовать другой набор классов, чем хотелось бы получить с использованием приведённых данных, поэтому такой способ также нельзя считать универсальным.

Широкое распространение получило создание и предоставление в общий доступ наборов искусственно сгенерированных данных [38]. Авторы таких датасетов анализируют трафик реальных пользователей и на основе результатов этого анализа создают ботов, которые генерируют трафик с заданными определёнными свойствами. Получаемые данные имитируют существующий трафик в сети, но при этом не компрометируют реальных людей и не содержат личных и чувствительных данных. Здесь, опять же, возникает вопрос соответствия такого искусственного трафика и реального: все ли особенности были учтены, не являются ли искусственные данные излишне детерминированными, насколько все их характеристики, которые могут использоваться алгоритмами машинного обучения для выделения закономерностей, отражают реальную ситуацию, а насколько определяются скриптами генерации.

Единого ответа на все эти вопросы, которые нужно решить для создания общего универсального набора данных для задачи классификации трафика, на данный момент не существует. Однако для более эффективной работы в данной области нужно искать пути решения этой проблемы.

7.2 Классификация с использованием частей потоков, классификация по середине потока

Согласно исследованиям, наиболее хорошие результаты показывают системы, использующие для классификации потоков информацию о его первых пакетах. Также существует достаточное количество исследований, показывающих, что для классификации потоков не обязательно дожидаться их окончания, а достаточно использовать первые N пакетов. Однако мы не всегда можем поймать или зафиксировать начало потока. Вдобавок, для многих практических приложений требуется только классификация потоков достаточной длины, а многие из слишком коротких потоков являются ошибочными. В таком случае желательно было бы не начинать сохранять информацию о потоке до того момента, как мы увидим хотя бы несколько пакетов из него, для экономии памяти системы. На данный момент нет достаточного количества полных исследований, показывающих, насколько эффективной может быть классификация потоков в случае использования произвольной части пакетов из их середины и применимо ли данное предложение в реальной жизни.

7.3 Работа с новыми классами данных

Так как новые приложения и протоколы продолжают появляться регулярно, становится невозможно в любой момент времени поддерживать базы данных и классификаторы в актуальном состоянии (нужно зафиксировать момент появления нового класса, собрать достаточное количество данных для него и переобучить классификаторы - всё это займёт время, даже если есть возможность производить эти действия в автоматическом режиме). Распределение примеров только по набору известных классов также не является решением, поэтому возникает проблема выделения в классификаторе неизвестного класса (классов). В идеале система должна уметь объединять новые неизвестные для себя объекты в один класс (например, с помощью алгоритмов кластеризации) и запрашивать для него метку, однако даже более простую задачу выделения всех неизвестных примеров в один общий класс нельзя считать окончательно решённой.

7.4 Использование полученных моделей в других сетях

Как уже было отмечалось выше, выбор данных для обучения очень важен при построении модели, и результаты работы классификатора в сети или в месте сети, отличном от того, где он обучался, ухудшает качество его работы [40]. Для борьбы с этой проблемой нужно больше исследований относительно того, сколько данных необходимо для переобучения модели под другую сеть, как меняется точность классификации со временем, можно ли автоматически отслеживать необходимость в дообучении, и насколько можно автоматизировать этот процесс.

Список литературы / References

[1]. Rezaei S., Liu X. Deep learning for encrypted traffic classification: An overview. IEEE Communications Magazine, vol. 57, issue 5, 2019, pp. 76-81.

[2]. Jamshidi S. The Applications of Machine Learning Techniques in Networking. Available at: https://www.cs.uoregon.edu/Reports/AREA-201902-Jamshidi.pdf, accessed 30.10.2020.

[3]. Hubballi N., Swarnkar M. BitCoding: Network Traffic Classification Through Encoded Bit Level Signatures. IEEE/ACM Transactions on Networking, vol. 26, issue 5, 2018, pp. 1-13.

[4]. Hubballi N., Swarnkar M., Conti M. BitProb: Probabilistic Bit Signatures for Accurate Application Identification. IEEE Transactions on Network and Service Management, vol. 17, no. 3, 2020, pp. 1730-1741.
[5]. Finamore A., Mellia M., Meo M., Rossi D. KISS: Stochastic Packet Inspection Classifier for UDP Traffic. IEEE/ACM Transactions on Networking, vol. 18, no. 5, 2010, pp. 1505-1515.
[6]. Dorfinger P., Panholzer G., John W. Entropy estimation for real-time encrypted traffic identification. In Proc. of the Third international conference on Traffic monitoring and analysis (TMA'11), 2011, pp. 164-171.
[7]. Khakpour A.R., Liu A.X. High-Speed Flow Nature Identification. In Proc. of the 29th IEEE International Conference on Distributed Computing Systems, 2009, pp. 510-517.
[8]. Doroud H., Aceto G. et al. Speeding-Up DPI Traffic Classification with Chaining. In Proc. of the IEEE Global Communications Conference (GLOBECOM), 2018, pp. 1-6.
[9]. Vu L., Bui C.T., Nguyen Q.U. A Deep Learning Based Method for Handling Imbalanced Problem in Network Traffic Classification. In Proc. of the Eighth International Symposium on Information and Communication Technology, 2017, pp. 333-339.
[10]. Oudah H., Ghita B., Bakhshi T. A Novel Features Set for Internet Traffic Classification using Burstiness. In Proc. of the 5th International Conference on Information Systems Security and Privacy, vol. 1, 2019, pp. 397-404.
[11]. Aceto G., Ciuonzo D., Montieri A., Pescapé A. Multi-classification approaches for classifying mobile app traffic. Journal of Network and Computer Applications, vol. 103, 2018, pp. 131-145.
[12]. Lotfollahi M., Jafari Siavoshani M., Shirali Hossein Zade R. et al. Deep packet: a novel approach for encrypted traffic classification using deep learning. Soft Computing, vol. 24, issue 3, 2020, pp. 1999-2012.
[13]. Gómez S.E., Martínez B.C. et al. Ensemble network traffic classification: Algorithm comparison and novel ensemble scheme proposal. Computer Networks, vol. 127, 2017, pp. 68-80.
[14]. Lopez-Martin M., Carro B., Sanchez-Esguevillas A., Lloret J. Network Traffic Classifier with Convolutional and Recurrent Neural Networks for Internet of Things. IEEE Access, vol. 5, 2017, pp. 18042-18050.
[15]. Mercaldo N., Lu W. Classification of Web Applications Using AiFlow Features. In Proc. of the Workshops of the International Conference on Advanced Information Networking and Applications, 2020, pp. 389-399.
[16]. Wang P., Chen X., Ye F., and Sun Z. A survey of techniques for mobile service encrypted traffic classification using deep learning. IEEE Access, vol. 7, 2019, pp. 54024-54033.
[17]. Takyi K., Bagga A., Gupta P. A Semi-Supervised QoS-Aware Classification for Wide Area Networks with Limited Resources. International Journal of Innovative Technology and Exploring Engineering, vol. 8, issue 11, 2019, pp. 970-981.
[18]. Li W., Moore A. W. A Machine Learning Approach for Efficient Traffic Classification. In Proc. of the 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2007, pp. 310-317.
[19]. Ding Y. A method of imbalanced traffic classification based on ensemble learning. In Proc. of the IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), 2015, pp. 1-4.
[20]. Carela-Español V. et al. Analysis of the impact of sampling on NetFlow traffic classification. Computer Networks, vol. 55, issue 5, 2011, pp. 1083-1099.
[21]. Oudah H., Ghita B., Bakhshi T. Network application detection using traffic burstiness. In Proc. of the World Congress on Internet Security, 2017, pp. 148-152.
[22]. Soylu T., Erdem O., Carus A. Bit vector-coded simple CART structure for low latency traffic classification on FPGAs. Computer Networks, 2020, vol. 167, article id 106977.
[23]. Wang W., Zhu M., Wang J., Zeng X., Yang Z. End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In Proc. of the IEEE International Conference on Intelligence and Security Informatics (ISI), 2017, pp. 43-48.
[24]. Zhao S., Chen S. et al. Identifying Known and Unknown Mobile Application Traffic Using a Multilevel Classifier. Security and Communication Networks, vol. 2019, 2019, article id 9595081.
[25]. Brissaud P., Francçis J., Chrisment I., Cholez T., Bettan O. Transparent and Service-Agnostic Monitoring of Encrypted Web Traffic. IEEE Transactions on Network and Service Management, vol. 16, no. 3, 2019, pp. 842-856.

[26]. Jin Y., Duffield N. et al. A modular machine learning system for flow-level traffic classification in large networks. ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 6, issue 1, 2012, pp. 1-34.
[27]. Rezaei S., Kroencke B., Liu X. Large-Scale Mobile App Identification Using Deep Learning. IEEE Access, vol. 8, 2020, pp. 348-362.
[28]. Hasibi R., Shokri M., Dehghan M. Augmentation scheme for dealing with imbalanced network traffic classification using deep learning. arXiv preprint, arXiv:1901.00204, 2019.
[29]. Zhao L. et al. A novel network traffic classification approach via discriminative feature learning. In Proc. of the 35th Annual ACM Symposium on Applied Computing, 2020, pp. 1026-1033.
[30]. Rezaei S., Liu X. How to achieve high classification accuracy with just a few labels: A semi-supervised approach using sampled packets. arXiv preprint, arXiv:1812.09761, 2018.
[31]. Wang Z. The applications of deep learning on traffic identification. BlackHat USA, vol. 24, issue 11, 2015, pp. 1-10.
[32]. Zheng W., Gou C., Yan L., Mo S. Learning to Classify: A Flow-Based Relation Network for Encrypted Traffic Classification. In Proc. of the Web Conference, 2020, pp. 13-22.
[33]. Zeng Y., Qi Z. et al. TEST: an End-to-End Network Traffic Examination and Identification Framework Based on Spatio-Temporal Features Extraction. arXiv preprint, arXiv:1908.10271, 2019.
[34]. De Montigny-Leboeuf A. Flow attributes for use in traffic characterization. Technical Note CRC-TN-2005-00, Communications Research Centre Canada, 2005.
[35]. Burschka S., Dupasquier B. Tranalyzer: Versatile high performance network traffic analyser. IEEE Symposium Series on Computational Intelligence (SSCI), 2016, pp. 1-8.
[36]. Orebaugh A., Ramirez G., Beale J. Wireshark & Ethereal network protocol analyzer toolkit. Elsevier, 2006, 448 p.
[37]. Deri L. et al. ndpi: Open-source high-speed deep packet inspection. In Proc. of the IEEE International Wireless Communications and Mobile Computing Conference (IWCMC), 2014, pp. 617-622.
[38]. Carela-Español V., Bujlow T., Barlet-Ros P. Is our ground-truth for traffic classification reliable? Lecture Notes in Computer Science, vol. 8362, 2014, pp. 98-108.
[39]. Bujlow T., Carela-Español V., Barlet-Ros P. Independent comparison of popular DPI tools for traffic classification. Computer Networks, vol. 76, 2015, pp. 75-89.
[40]. Khatouni A. S., Heywood N. Z. How much training data is enough to move a ML-based classifier to a different network? Procedia Computer Science, vol. 155, 2019, pp. 378-385.
[41]. Fan J., Xu J., Ammar M. H., Sue M. Prefix-Preserving IP Address Anonymization: Measurement-Based Security Evaluation and a New Cryptography-Based Scheme. In Proc. of the 10th IEEE International Conference on Network Protocols (ICNP 2002), 2002, pp. 12-15.
[42]. Draper-Gil G., Lashkari A.H., Mamun M.S.I., Ghorbani A.A. Characterization of encrypted and VPN traffic using time-related. In Proc. of the 2nd International Conference on Information Systems Security and Privacy (ICISSP), 2016, pp. 407-414.

Информация об авторах / Information about authors

Александр Игоревич ГЕТЬМАН – кандидат физико-математических наук, старший научный сотрудник ИСП РАН, доцент ВШЭ. Сфера научных интересов: анализ бинарного кода, восстановление форматов данных, анализ и классификация сетевого трафика.

Aleksandr Igorevich GETMAN – PhD in physical and mathematical sciences, senior researcher at ISP RAS, associate professor at HSE. Research interests: binary code analysis, data format recovery, network traffic analysis and classification.

Мария Кирилловна ИКОННИКОВА – аспирант. Научные интересы: анализ сетевого трафика, машинное обучение.

Maria Kirillovna IKONNIKOVA – postgraduate student. Research interests: network traffic analysis, machine learning.