Научная статья
УДК 004.942
DOI:10.31854/1813-324X-2023-9-5-66-78

Модель классификации трафика в программно-конфигурируемых сетях с элементами искусственного интеллекта

Василий Сергеевич Елагин, v.elagin@sut.ru

Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича,
Санкт-Петербург, 193232, Российская Федерация

Аннотация: *Классификация приложений необходима для повышения производительности сети. Однако при постоянном росте числа пользователей и приложений, а также масштабирования сетей, традиционные методы классификации не могут справляться в полной мере с идентификацией и классификацией сетевых приложений с необходимым уровнем задержки. Применение технологии глубокого обучения совместно с особенностями архитектуры программно-конфигурируемых сетей (SDN, аббр. от англ. Software-Defined Networking) позволит реализовать новую гибридную глубокую нейронную сеть для классификации приложений, которая сможет обеспечить высокую точность классификации без ручного выбора и извлечения признаков. В предлагаемой структуре предложена классификация приложений, с учетом логического централизованного управления на контроллере SDN. Обработанные данные используются для обучения гибридной глубокой нейронной сети, состоящей из многоуровневого автокодировщика, с высокой размерностью скрытого слоя и выходного слоя на базе регрессии softmax. Необходимые параметры сетевого потока могут быть получены при обработке трафика многоуровневым автокодировщиком вместо ручной обработки. Слой регрессии softmax используется в качестве конечного классификатора приложений. В статье приведены результаты моделирования, которые демонстрируют преимущества предложенного метода классификации, по сравнении с методом опорных векторов.*

Ключевые слова: *программно-конфигурируемые сети, ПКС, нейронная сеть, классификация трафика, регрессия softmax*

Финансирование: *научная статья подготовлена в рамках прикладных научный исследований СПбГУТ, регистрационный номер 123060900012-6 в ЕГИСУ НИОКТР.*

Ссылка для цитирования: Елагин В.С. Модель классификации трафика в программно-конфигурируемых сетях с элементами искусственного интеллекта // Труды учебных заведений связи. 2023. Т. 9. № 5. С. 66–78. DOI:10.31854/1813-324X-2023-9-5-66-78

Traffic Classification Model in Software-Defined Networks with Artificial Intelligence Elements

Vasiliy Elagin, v.elagin@sut.ru

The Bonch-Bruevich Saint Petersburg State University of Telecommunications,
St. Petersburg, 193232, Russian Federation

Abstract: *Application classification is essential to improve network performance. However, with the constant growth in the number of users and applications, as well as the scaling of networks, traditional classification methods cannot fully cope with the identification and classification of network applications with the required level of delay. The use of deep learning technology together with the architecture features of software-defined networks (SDN) will allow the implementation of a new hybrid deep neural network for application classification, which can*

provide high classification accuracy without manual selection and feature extraction. The proposed structure proposes a classification of applications, taking into account the logical centralized management on the SDN controller. The processed data is used to train a hybrid deep neural network consisting of stacked autoencoder with a high dimensionality of the hidden layer and an output layer based on softmax regression. The necessary network flow parameters can be obtained by processing traffic with a stacked auto-encoder instead of manual processing. The softmax regression layer is used as the final application classifier. The article presents simulation results that demonstrate the advantages of the proposed classification method in comparison with the support vector machine.

Keywords*: software-defined networks, SDN, neural network, traffic classification, softmax regression*

Funding*:the article was prepared within the framework of applied scientific research of SPbSUT, registration number 123060900012-6 in the EGISU R&D.*

For citation: Elagin V. Traffic Classification Model in Software-Defined Networks with Artificial Intelligence Elements. *Proceedings of Telecommun. Univ.* 2023;9(5):66–78. DOI:10.31854/1813-324X-2023-9-5-66-78

Введение

Современные требования к обработке и управлению трафиком в сетях связи требуют постоянного увеличения производительности сетевых элементов и серверной инфраструктуры, однако, традиционная сетевая архитектура не удобна для сбора и обработки больших объемов данных и принятия решений из-за своей распределенности и децентрализованности. Поэтому целесообразно рассмотреть архитектуру программно-конфигурируемых сетей (SDN, *аббр. от англ.* Software-Defined Networking), в которых используется логически централизованное управление, в рамках которого можно организовать сбор нужных параметров. Кроме того, SDN на уровне управления (*англ.* Control Plane) открыты для развертывания новых сетевых сервисов. Архитектура SDN позволяет обеспечить полное управление трафиком (*англ.* Traffic Engineering) для повышения производительности сети и управления. Что еще более важно, SDN может значительно облегчить сбор и обработку больших объемов данных, благодаря централизованному управлению через SDN-контроллер [1, 2].

Максимально точная классификация сетевых приложений необходима для реализации различных сетевых функций, таких как динамическая маршрутизация трафика, трафик инжиниринг, обеспечение качества обслуживания (QoS, *аббр. от англ.* Quality of Service), прогнозирования событий и состояния сети [3].

Для повышения точности идентификации и классификации сетевых приложений за последнее время было предложено большое количество различных методов, в основном включая сигнатурный подход, основанный на заданных шаблонах трафика, подход, основанный на поведенческом анализе, и подход, основанный на машинном обучении [4]. В настоящее время активное развитие получила технология, основанная на машинном обучении и обработке больших данных, в которой обычно применяют нейронные сети с различными методами принятия решения: метод опорных векторов (SVM, *аббр. от англ.* Support Vector Machine), метод обратного распространения ошибки (*англ.* Backpropagation), Байесовские сети и алгоритм C4.5 для построения деревьев решений [5–13].

В отличие от традиционных методов сигнатурного анализа и поведенческого анализа, в методах классификации, основанных на машинном обучении, статистические характеристики трафика, включая размер пакета, время между пакетами, длительность сеанса и т. д., используются для идентификации и классификации сетевого приложения с помощью машинного обучения.

Нейронная сеть с небольшим числом слоев обладает ограниченной возможностью к обучению из-за небольшого количества слоев для нелинейного извлечения признаков и больше подходит для решения задач с ограниченным набором данных. Однако в связи с постоянным расширением масштабов сетей и наступлением эры больших данных сетевой трафик и число приложений стремительно растет. Это не позволяет традиционным нейронным сетям при растущем объеме сетевого трафика своевременно справляться с классификацией из-за ограниченной способности к обучению. Для извлечения дополнительных признаков из трафика, в том числе на прикладном уровне, на крупных сетях требуется нейронная сеть с глубоким обучением.

На данный момент активно используются модели глубоких нейронных сетей (ГНС), которые уже были апробированы для решения различных задач, в частности: сверточная нейронная сеть (CNN, *аббр. от англ.* Convolutional Neural Network), многослойный автокодировщик, глубокая сеть доверия (DBN, *аббр. от англ*. Deep Belief Network); они показали свою эффективность в областях распознавания, обработки речи, классификации признаков и объектов. Для обучения и дообучения ГНС обычно используются оригинальные алгоритмы – глубокого машинного обучения, в тех случаях, когда традиционные алгоритмы обучения не подходят по причине ограничения значений в областях локальных оптимумов и исчезающего градиента [14–15].

В данной статье будет предложен метод классификации сетевых приложений на основе гибридной модели глубокого обучения (DL – *аббр. от англ.* Deep Learning).

При этом необходимо отметить, что процесс классификации приложений определяется тремя необходимыми функциями: обучение модели, сам процесс классификации и последующая проверка и валидация результатов. При этом необходима постоянная обработка большого объема трафика и его параметров при предварительной обработке, построении обучающего и тестового наборов, разметки данных, классов и т. д.

Применение программно-конфигурируемых сетей SDN позволит упростить эти задачи благодаря декомпозиции функций управления и плоскости данных в SDN, а также наличию единого управляющего контроллера, который используется для виртуализации и быстрого внедрения новых сетевых функций, трафик инжиниринга и гибкого управления сетью и др. [16–22].

Эти возможности SDN могут значительно облегчить сбор и обработку больших массивов данных [23]. Используя логический централизованный контроллер и вычислительные возможности, можно значительно упростить сбор и обработку большого объема сетевого трафика.

Таким образом, целесообразно принять во внимание архитектуру SDN и технологию глубокого обучения для формирования нового метода классификации сетевых приложений, который будет состоять из следующих модулей:

– на основе архитектуры SDN предлагается новая модель классификации сетевых приложений с использованием глубокого обучения (структура позволяет легко собирать и обрабатывать большие массивы трафика для классификации приложений);

– комбинация многоуровневого автокодировщика и регрессионной модели softmax, позволяет разработать гибридную модель ГНС для классификации приложений; в этой модели многоуровневый автокодировщик используется для извлечения параметров потока и его свойств, а регрессия softmax используется для классификации сетевых приложений.

В дальнейшем в статье будут представлены результаты экспериментальной проверки предложенных моделей.

Существующие модели и решения для классификации трафика

Многоуровневый автокодировщик

Многоуровневый автокодировщик – это нейронная сеть с несколькими слоями Sparse Autoencoders. Структура автокодировщика состоит всего из 3-х типов слоев: входного слоя, скрытого слоя и слоя декодирования/реконструкции.

Стоит отметить основные отличия многоуровневого автокодировщика от других ГНС, например, ограниченной машины Больцмана (RBM, *аббр. от англ.* Restricted Boltzmann Machine) или сверточной нейронной сети:

1) в многоуровневом автокодировщике размерность ввода равна размерности вывода;

2) для обучения многоуровневого автокодировщика необходим только немаркированный (а не размеченный) набор данных, из-за неконтролируемого подхода к обучению (в отличие от CNN, представляющей собой алгоритм обучения с учителем, и DBN, состоящей из нескольких RBM и использующей алгоритм обучения с частичным привлечением учителя);

3) выходные результаты автокодировщика – это еще одно представление выходных данных, высокоуровневые характеристики потока могут быть извлечены посредством обучения многоуровневого автокодировщика алгоритмом без учителя.

Исходя из перечисленного выше, в данной статье был выбран многоуровневый автокодировщик в качестве нейронной сети с глубоким обучением для извлечения признаков потока.

Регрессия softmax

Модель регрессии softmax представляет собой поколение традиционной модели логистической регрессии и относится к алгоритму обучения с учителем. Он удобен при создании мультиклассовой логистической регрессии и заключается в том, чтобы не разбивать многоклассовые данные на несколько датасетов, используя бинарный классификатор, а сразу применять функции, которые позволяют работать с множеством классов. Это подходит для решения задачи многоклассовой классификации, поэтому имеет широкое применение во многих приложениях.

Для задачи классификации K классов вероятность того, что входные данные $x^{(i)}$ принадлежат классу j, может быть представлена как:

$$P\left(z^{(i)} = j \middle| x^{(i)}\right) = \frac{e^{\theta_j^T X(i)}}{\sum_{i=1}^{K} e^{\theta_i^T X(i)}}, \quad (1)$$

где $z^{(i)}$ – результат на выходе; θ_j – вектор входных параметров.

Кроме того, стоит отметить, что регрессия softmax часто используется в качестве выходного уровня гибридных ГНС как конечный классификатор [24]. В данной статье слой регрессии softmax планируется подключить к многоуровневому автокодировщику для повышения точности классификации приложений.

Стоит отметить, что модель регрессии softmax является одним из алгоритмов, реализующих множественную логистическую регрессию в случае категориальной переменной, поэтому применение данной регрессии позволяет исключить использование нескольких моделей множественной логистической регрессии для обработки большого числа классов трафика. Это, в основном, связано с практической реализацией моделей, так как большинство программных библиотек реализовано на низкоуровневых языках и оптимизированы под свои задачи, поэтому при программировании на более высоком уровне абстракции (как в нашем случае) возможны коллизии и неверные интерпретации отдельных классов или параметров. Поэтому для решения поставленной задачи многопараметрической классификации слой регрессии softmax планируется подключить к многоуровневому автокодировщику для повышения точности классификации приложений.

Большинство методов машинного обучения сосредоточены на эффективной настройке используемых методов и подбору комбинации признаков в потоке трафика, для повышения точности и полноты классификатора [4–13, 25, 26].

Перед обучением нейронной сети необходимо выбрать несколько параметров потока, так, например, в работе [4] выделено 37 параметров, включая наименование протокола, номера портов, продолжительность сессии, пропускную способность в пакетах и в байтах. Однако при экспоненциальном росте сетевого трафика в действующей сетевой архитектуре значительно усложняется задача сбора и обработки всего проходящего трафика, а также его обработка для получения искомых характеристик потока и повышения точности классификации

Существует новая архитектура выбора характеристик потока, которая может определить необходимое подмножество параметров для классификации различных типов сеансов связи, в рамках архитектуры SDN [26]. Данная архитектура состоит из менеджера потоков и системы выбора потока. Первый используется для исчисления характеристик потока, система выбора потока отвечает за анализ и интерпретацию этих характеристик.

В отдельных работах [11–13] были рассмотрены варианты применения архитектуры SDN для классификации сетевых приложений. Например, Прасад и Катаока [11] разработали многопутевой механизм пересылки пакетов с учетом приложений, объединив машинное обучение и SDN. Для получения данных о составе сервисов для построения классификатора приложений использовался алгоритм обучения деревьев решений C4.5, а в контроллер, как в единую точку управления и мониторинга сети, были интегрированы тренажер и классификатор, работающий на базе машинного обучения. Кроме того, в исследовании разработана новая архитектура применения машинного обучения для сбора данных и классификации трафика с учетом интеграции с SDN. В данной архитектуре контроллер получает статистику потоков от коммутаторов и агрегирует ее, в дальнейшем система применяет контролируемый алгоритм машинного обучения для классификации трафика [12].

Отдельные статьи предлагают применение классификации трафика для удовлетворения требований качества обслуживания QoS приложений, так, например, в работе [13] была разработана структура классификации трафика с учетом QoS в SDN. В этой структуре для обнаружения основных потоков применялась технология DPI (*от англ.* Deep Packet Inspection), а для классификации трафика с учетом QoS с помощью функции сопоставления использовался полууправляемый алгоритм машинного обучения. В частности, поток определенного приложения был сопоставлен с предварительно выделенным классом QoS в соответствии с его особенностями.

Хотя вышеупомянутые исследовательские работы учитывают архитектуру SDN для классификации приложений, эти методы в целом относятся к традиционной сети с неглубоким обучением, которая не может эффективно работать с массивными данными из-за ограниченных возможностей обучения функциям. В отличие от нее, сеть с глубоким обучением обладает более мощной способностью к изучению признаков и выделению параметров потока трафика [27].

Глубокая нейронная сеть в последнее время достаточно часто используется для обнаружения вторжений, прогнозирования трафика и классификации приложений [28–32]. Гибридные интеллектуальные архитектуры используются для различных целей, например, обнаружения сетевых аномалий [28, 29]. При этом исследователи предлагают различные варианты используемых нейронных сетей, в основном это RBM, иногда в комбинации с SVM.

Отдельно стоит отметить применение глубокого обучения для прогнозирования поведения сетевого трафика [30]. Для извлечения характеристик потока обычно используется модель традиционного многоуровневого автокодировщика или многоуровневого автокодировщика Левенберга – Марквардта [31]. В некоторых работах, например [32], сеть глубокого обучения для прогнозирования сетевого трафика предлагается комбинировать из DBN и слоя мультизадачной регрессии.

В данном исследовании предлагается применение архитектуры, объединяющей SDN и модели глубокого обучения. Контроллер SDN будет использоваться для обработки больших объемов

сетевого трафика и получения статистик потоков. Гибридную сетевую модель глубокого обучения, состоящую из многоуровневого автокодировщика и слоя регрессии softmax, планируется использовать для извлечения параметров потока и построения классификатора приложений.

Модельная сеть

Для идентификации потока предлагается ограничиться пятью основными параметрами: IP-адресом источника, IP-адресом назначения, транспортным протоколом (TCP или UDP), номером порта источника, номером порта назначения.

На рисунке 1 отражена схема предлагаемой структуры, в которой плоскость управления состоит из четырех основных функциональных модулей: мониторинга сети, сбора статистики потоков, обработки данных и глубокого обучения, а также базы данных, в которой хранится информация о параметрах потоков.

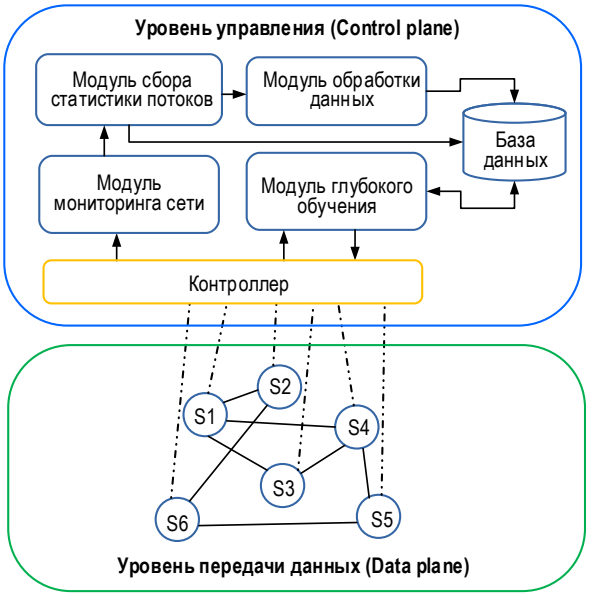


Рис. 1. Архитектура построения модельной сети для классификации приложений трафика

Fig. 1. The Application Classification Framework

В управлении сетевой инфраструктурой можно выделить ряд отдельных задач, которые должны решаться системой управления.

Модуль мониторинга сети – необходим для агрегации и анализа информации о сетевом трафике от коммутаторов в выделения информации о потоке (тип протокола, порт источника, порт назначения и т. д.).

Модуль сбора статистики потока – отвечает за обработку собранной информации о потоке и выделение из нее необходимых характеристик потока (например, продолжительность сессии, время между пакетами и распределение числа пакетов в потоке), а также данных о приложении (например, метка потока). В дальнейшем полученные параметры сохраняются в базе данных.

Модуль обработки данных – обрабатывает и использует характеристики потока и метки потока, хранящихся в базе данных, для создания обучающего набора в нейронной сети, а также тестового набора для встроенного классификатора приложений.

Модуль глубокого обучения – отвечает за обучение предложенной модели гибридной ГНС с использованием обучающего набора и тестового набора для работы встроенного классификатора приложений. По завершении обучения данный модуль при обнаружении нового потока классифицирует поступающий поток сетевых приложений в определенный класс приложений.

Информация, хранящаяся в базе данных, делится на 2 категории: информация из собранного сетевого трафика и размеченный набор данных, состоящий из характеристик потоков и меток потоков, которые нужны для обучения и валидации гибридной модели глубокого обучения. При этом наборы данных в БД периодически обновляются.

Основной процесс работы системы может быть представлен следующими этапами.

Этап 1. При поступлении нового потока коммутатор отправляет в контроллер информацию о нем (протокол, порт источника, порт получателя, время прибытия потока, продолжительность потока, интервалы прибытия пакетов в потоке и т. д.).

Этап 2. При получении этой информации на контроллере происходит выделение и первичная обработка данных для передачи их в модуль мониторинга сети.

Этап 3. На основе данных из контроллера модуль мониторинга сети периодически обновляет информацию о потоках в базе данных. В дальнейшем информация передается в модуль статистики потока.

Этап 4. Модуль статистики потока извлекает возможные характеристики и метки потока и сохраняет их в базе данных, а также передает в модуль обработки данных.

Этап 5. Данные обрабатываются для построения обучающего сета и классификации приложений.

Этап 6. ГНС обучается, и в результате модуль глубокого обучения может начать классификацию приложения.

Гибридная сетевая модель глубокого обучения

Далее рассмотрим предлагаемый гибридный метод классификации приложений на основе сети глубокого обучения.

Сеть глубокого обучения

Многоуровневый автокодировщик, исходя из определения, состоит из нескольких архитектур, и представляет собой одну специальную нейронную сеть, состоящую из входного слоя, скрытого слоя и выходного слоя. Типичная архитектура приведена на рисунке 2, где x_i – многомерный вектор входных признаков автокодировщика, y_i – выходные данные скрытого слоя автокодировщика, z_i – выходные данные выходного слоя автокодировщика.

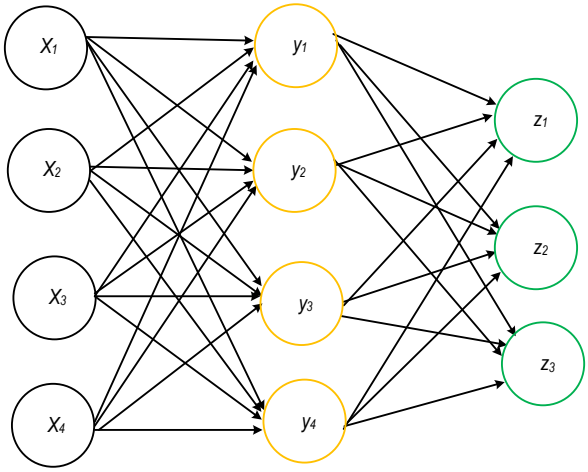


Рис. 2. Модель многоуровневого автокодировщика

Fig. 2 The Stacked Autoencoder Model

В отличие от традиционной модели нейронной сети, выход слоя автокодировщика используется как вход на следующий. В частности, преобразование отображения из входного слоя в скрытый уровень можно рассматривать как процесс кодирования, а отображение данных из скрытого слоя в выходной уровень – это декодирование.

В данном случае x – это M-мерный входной вектор признаков автокодировщика. При подаче необработанных данных на входной слой, их закодированное представление поступает на скрытый слой и может быть представлено по формуле:

$$y(x) = h(W_1 x + b_1). \quad (2)$$

Аналогичным образом можно реконструировать (декодировать) данные для выходного слоя, воспользовавшись выражением:

$$z(x) = f(W_2 y(x) + b_2), \quad (3)$$

где W_1, W_2 – матрицы весов связей между входным и скрытым слоем и между скрытым и выходным слоем в автокодировщике, соответственно; b_1, b_2 – смещение скрытого и выходного слоев в автокодировщике; $h(.)$ – функция активации скрытого слоя в автокодировщике; $f(.)$ – функция активации выходного слоя в автокодировщике.

Целевая функция автокодировщика, необходимая для достижения минимизации ошибки между потоком входных данных и выходных результатов автокодировщика, представлена формулой:

$$L(x, z) = \frac{1}{2N} \sum_{i=1}^{N} \|x_i - z(x_i)\|^2, \quad (4)$$

где N – общее количество обучающих выборок.

Выходные данные последнего скрытого слоя в многоуровневом автокодировщике представляют собой форму входных данных для выходного слоя, поэтому их нельзя использовать для исполнения функции классификации. В данной статье предложено объединить классификатор softmax с многоуровневым автокодировщиком для классификации приложений. Общая модель ГНС для классификации приложений состоит из одного входного слоя, нескольких скрытых слоев и классификатора softmax, как показано на рисунке 3.

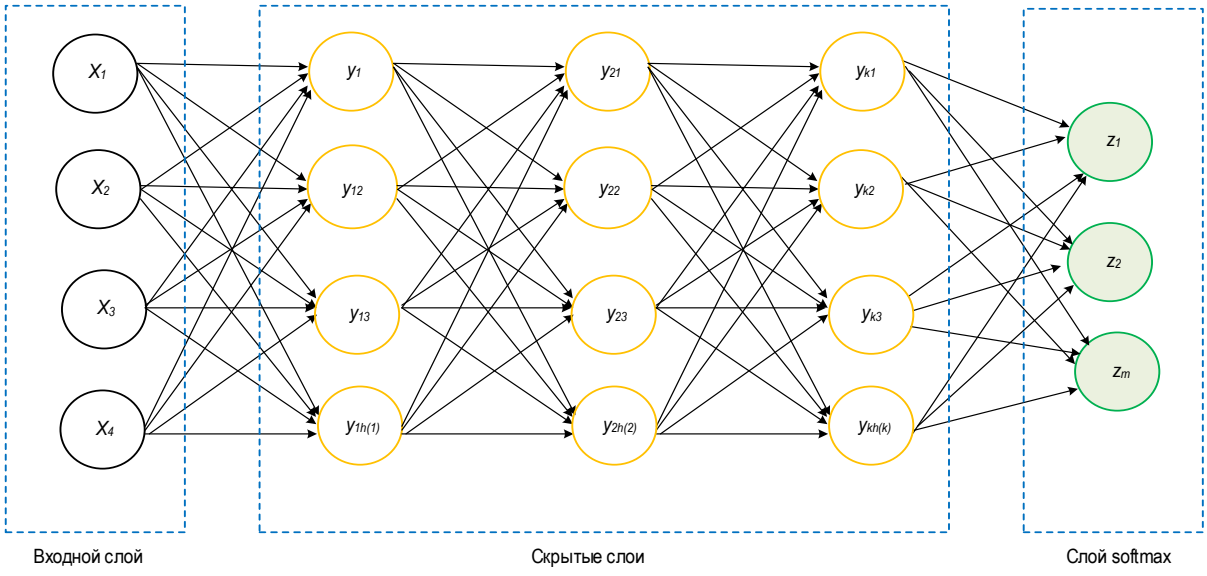


Рис. 3. Модель сети глубокого обучения

Fig. 3. Deep Learning Network Model

В целях обработки неразмеченного набора данных предлагается использование «жадного» алгоритма послойного предобучения [33]. В рамках работы «жадного» алгоритма принимаются локально оптимальные решения на каждом слое, при этом последовательно происходит обучение одного за другим скрытого слоя. Через обучение мы получаем веса и смещения следующего скрытого слоя.

Для предлагаемой модели гибридной DL-сети в процессе обучения используются как индексированный, так и неразмеченный набор данных. Обучение многоуровнего автокодировщика использует немаркированный набор данных. Проиндексированный набор данных необходим для процесса обучения слоя регрессии softmax и процесса тонкой настройки модели гибридной глубокой сети. Так, например: предположим, что помеченный набор данных определяется как X_L = = (X_{L1}, X_{L2}, …, X_{LM}) с M метками Y_L = (Y_{L1}, Y_{L2}, …, Y_{LM}), а немаркированный – как X_U = (X_{U1}, X_{U2}, …, X_{UN}) без N меток, где L –размеченные данные, а U – неразмеченные.

В качестве функции активации для $h(.)$ и $f(.)$ будем использовать сигмоидальную функцию:

$$h(x) = f(x) = S(x) = \frac{1}{1 + e^{-x}}. \quad (5)$$

Повышение эффективности обобщения многоуровневого автокодировщика может быть получено благодаря введению в функцию потерь ограничений по разреженности данных. Таким образом, функция потерь (*LS, от англ.* Lose Function) многоуровневого автокодировщика может быть представлена следующим образом:

$$LS(\Theta_1) = LA\big(x, z(x)\big) + \alpha \sum_{j=1}^{H} KL(\rho \,\|\widehat{\rho_j}), \quad (6)$$

$$LS(\Theta_1) = LA\big(x, z(x)\big) + \alpha \sum_{j=1}^{H} KL(\rho \,\|\widehat{\rho_j}), \quad (7)$$

$$KL\big(\rho\|\widehat{\rho_j}\big) = \rho \log\frac{\rho}{\widehat{\rho_j}} + (1-\rho)\log\frac{1-\rho}{1-\widehat{\rho_j}}, \quad (8)$$

$$\widehat{\rho_j} = \frac{1}{N}\sum_{i=1}^{N} Y_{Uj}(X_{Uj}), \quad (9)$$

где ρ – параметр разреженности многоуровневого автокодировщика; ρ_j – среднее значение активации j-го скрытого узла; H – количество скрытых узлов; $KL(.)$ – функция расхождения Кульбака – Лейблера, которая достигает минимального значения равного 0 при $\rho = \widehat{\rho_j}$, а также при $\theta_1 = (W_1, b_1)$, если W_1 и b_1 – вес и вектор смещения многоуровневого автокодировщика, соответственно; $LA(.)$ – ошибка реконструкции по всем обучающим выборкам; α – коэффициент для регулировки штрафа за разреженность.

Окончательный результат многоуровневого автокодировщика подается в качестве входных данных слоя softmax для обучения классификатора. Таким образом, регрессионная модель softmax обучается с использованием алгоритма контролируемого обучения с помеченным набором образцов.

Предполагается, что все размеченные образцы можно разделить на K классов; другими словами, для каждой выборки X_{Li} соответствующая выходная метка Y_{Li} может принимать K различных значений.

Значение вероятности $P(Z_i = j|X_{Li})$, где X_{Li} относится к j-*му* классу, может быть рассчитано по формуле (10). При этом Θ_2 = (W_2, b_2), где W_2 и b_2 – это вес и вектор смещения слоя регрессии softmax, соответственно.

Функция потерь классификатора softmax определяется согласно выражению (11), где M – общее количество наборов данных; $1(Z_i = j)$ – дискретная функция, которая принимает значение 1 при выполнении определенного условия, в противном случае принимает значение, равное 0; β – штрафной коэффициент; n – количество узлов входного слоя; θ_{ij} – вектор веса и смещения слоя softmax.

В соответствии с данной схемой можно определить целевую функцию гибридной модели на этапе тонкой настройки, как показано ниже, а третье слагаемое представляет собой штрафную функцию модели многоуровневого автокодировщика, согласно выражению (12) при θ = θ_1, θ_2, где H – общее количество скрытых слоев многоуровневого автокодировщика; θ_h – вектор веса и смещения многоуровневой модели автокодировщика.

$$h_{\Theta_2}\big(X_{L_i}\big) = \begin{bmatrix} P\big(Z_1 = 1\big|X_{L_i};\Theta_2\big) \\ P\big(Z_2 = 2\big|X_{L_i};\Theta_2\big) \\ \vdots \\ P\big(Z_K = K\big|X_{L_i};\Theta_2\big) \end{bmatrix} = \frac{1}{\sum_{j=1}^{K} e^{\Theta_j^T X_{L_i}}} \begin{bmatrix} e^{\Theta_1^T X_{L_i}} \\ e^{\Theta_2^T X_{L_i}} \\ \vdots \\ e^{\Theta_K^T X_{L_i}} \end{bmatrix}. \quad (10)$$

$$LSM(\Theta_2) = -\frac{1}{M}\left\{\sum_{i=1}^{M}\sum_{j=1}^{K} 1(Z_i = j)\log\frac{e^{\Theta_j^T X_{L_i}}}{\sum_{S=1}^{K} e^{\Theta_S^T X_{L_i}}}\right\} + \frac{\beta}{2}\sum_{i=1}^{K}\sum_{j=1}^{n}\Theta_{ij}^2\, 11\,. \quad (11)$$

$$LSS(\theta) = -\frac{1}{M}\left\{\sum_{i=1}^{M}\sum_{j=1}^{K} 1(Z_i = j)\log\frac{e^{\Theta_j^T X_{L_i}}}{\sum_{S=1}^{K} e^{\Theta_S^T X_{L_i}}}\right\} + \frac{\gamma}{2}\sum_{i=1}^{K}\sum_{j=1}^{n}\Theta_{ij}^2 + \frac{\gamma}{2}\sum_{h=1}^{H}\Theta_h^2\,. \quad (12)$$

Процесс обучения

Процесс обучения модели сети глубокого обучения состоит из 3-х этапов: предварительного обучения многоуровневого автокодировщика, обучения слоя softmax и тонкой настройки общей ГНС. На начальном этапе «жадный» алгоритм послойного предобучения используется для обучения весов многоуровневого автокодировщика методом без учителя. При обучении слоя регрессии softmax уже используется алгоритм с учителем. Во время тонкой настройки ГНС применяется алгоритм обратного распространения ошибки для точной настройки сети глубокого обучения, используя метод по размеченным данным. Таким образом, процесс обучения можно представить следующими этапами.

Этап 1. Немаркированные наборы данных X_U применяются для обучения первого скрытого слоя модели многоуровневого автокодировщика на основе метода без учителя путем минимизации целевой функции $LS(\theta_1)$.

Этап 2. Выходные данные первого скрытого слоя используются в качестве входных данных второго скрытого слоя для обучения второго автокодировщика в рамках той же целевой функцией $LS(\theta_1)$.

Этап 3. Выходные данные i-го скрытого слоя используются как входные данные $(i + 1)$-го скрытого слоя для обучения $(i + 1)$-го автокодировщика. Данная операция повторяется до тех пор, пока последний скрытый слой не будет достаточно обучен.

Этап 4. Выходные данные последнего скрытого слоя подаются в качестве входных данных слоя softmax для его обучения с помощью размеченного набора выборок данных X_L, используя алгоритм обучения с учителем.

Этап 5. Модель ГНС с полученными весами и смещением из вышеупомянутого процесса обучения активируется, а затем размеченный набор образцов X_L используется для точной настройки общей ГНС. Это минимизирует целевую функцию $LSS(\theta)$ с помощью алгоритма обратного распространения ошибки.

Алгоритм обучения для всей глубокой нейронной сети проиллюстрирован на рисунке 4. На первом шаге параметры многоуровневого автокодировщика включают веса W, смещение b, количество скрытых слоев M и комбинацию скрытых узлов в каждом скрытом слое. На десятом шаге параметры в слое softmax состоят из весов и смещения первичных для этого слоя. После этого на шаге 12 происходит использование параметров, полученных в процессе предварительной подготовки, для обновления весов гибридной ГНС, а алгоритм обратного распространения ошибки затем точнее настраивает веса во всей модели.

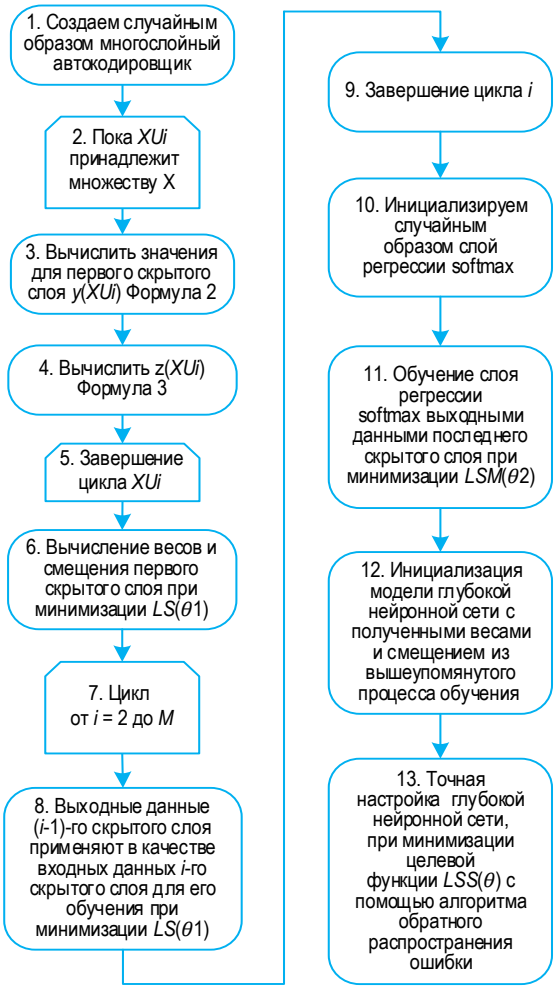


Рис. 4. Алгоритм обучения ГНС

Fig. 4. Deep Neural Network Learning Algorithm

Экспериментальные исследования

Для оценки производительности предлагаемого метода классификации приложений был проведен эксперимент по моделированию, который реализован в программной платформе Weka и MATLAB, работающей на персональном компьютере Intel (R) Core™ / 2,93 ГГц / 4 ГБ под управлением операционной системы Windows.

При моделировании каждый эксперимент повторялся 100 раз и бралось среднее значение в качестве окончательного результата. Коэффициенты ρ, α, β и γ инициализируются равными 0.1, 3, 0.001 и 0.001, соответственно. Кроме того, мы устанавливаем число итерации обучения равным 3000. Перед предварительной тренировкой случайным образом генерируются начальные векторы веса и смещения многоуровневого автокодировщика и слоя регрессии softmax.

Оценка характеристик предложенного метода классификации, с предлагаемым методом и моделью глубокого обучения будет производиться в сравнении с традиционной моделью классификатора на базе SVM.

Набор данных

Для тестирования метода был взят набор данных, полученный в компьютерной лаборатории Кембриджского университета и широко используемый во многих исследовательских работах по классификации трафика. [4, 8, 10, 25]. Набор данных состоит из отдельных файлов, собранных в разное время дня, и каждый набор – только из потоков TCP-трафика, для которых выделено 249 характеристик: в частности, продолжительность потока, метка класса приложения и др. Все сетевые потоки подразделяются на 10 классов, что приведено ниже в нотации «Условный номер: тип трафика / приложения (протоколы)»: 1) Передача данных / ftp; 2) Базы данных / postgress, sqlnet oracle, ingress; 3) Служебный трафик / Ssh, klogin, rlogin, telnet; 4) Почта / imap, pop2/3, smtp; 5) Сервисы / X11, dns, ident, ldap, ntp; 6) Трафик Интернет / www; 7) P2P / KaZaA, BitTorrent, GnuTella; 8) Злонамеренный трафик / Сигнатуры сетевых атак; 9) Игры / Microsoft Direct Play; 10) Мультимедиа / Windows Media Player, Real.

Исходные наборы данных можно скачать с сайта отдельного проекта компьютерной лаборатории Кембриджского университета [34].

Оценка работы модели

Чтобы оценить эффективность и производительность предложенного метода классификации, рассматривались такие показатели, как вероятность верного срабатывания, точность и полнота.

Вероятность верной классификации – это отношение общего количества приложений, правильно типизированные классификатором, к общему количеству образцов приложений. Используется для оценки точности классификатора приложений на всем наборе данных.

Точность классификации – это доля приложений, которые правильно отнесено к данному классу приложений.

Полнота классификации – это доля приложений, принадлежащих классу *i*, которые правильно отнесены к конкретному классу *i*.

Поскольку гибридная глубокая нейронная сеть с 5 скрытыми слоями и 10 скрытыми узлами имеет самую высокую вероятность верной классификации, поэтому в качестве основной DL-модели будет использоваться именно она (таблица 1).

На рисунке 5a (слева) приведено сравнение вероятностей верной классификации приложений для DL-моделей и на базе SVM для 10 различных наборов данных. Из сравнения видно, что первая имеет более высокую точность классификации. Однако стоит отметить, что DL-модель состоит из бóльшего количества скрытых слоев и использует алгоритм обучения более высокого уровня, поэтому способность к обучению классификации получается более эффективной. Кроме того, поскольку распределение трафика в каждом наборе данных отличается, общая точность классификации для разных наборов данных также незначительно разнится.

ТАБЛИЦА 1 Значения вероятности верной классификации в рассматриваемой модели

TABLE 1. The Performance of Deep Learning Network

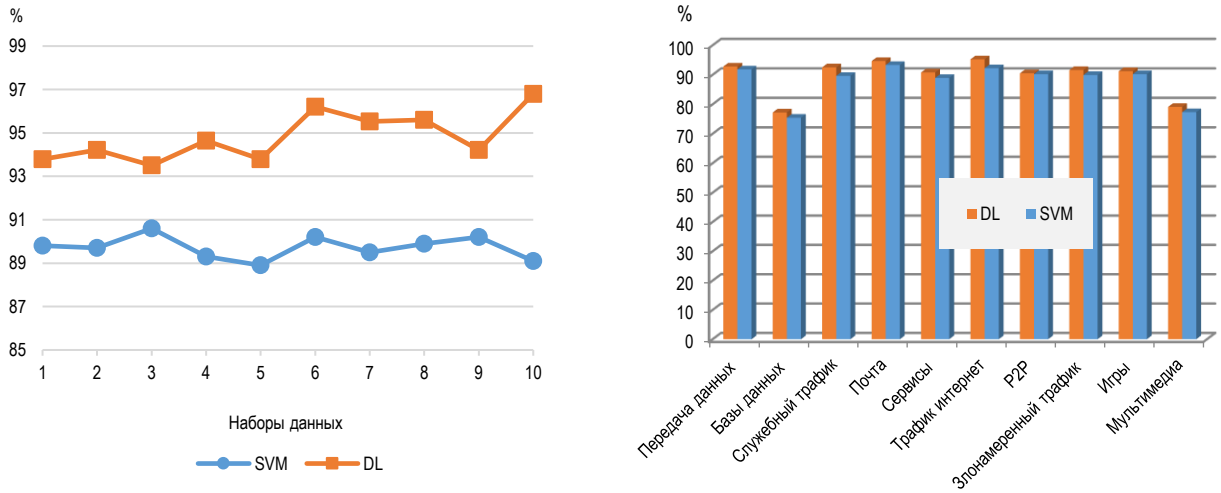
№ п/п	Число скрытых слоев	Число скрытых узлов	Вероятность верного срабатывания, %
1	3	5	82.44
2	3	10	85.05
3	3	15	85.33
4	4	5	85.63
5	4	10	88.75
6	4	15	88.15
7	5	5	90.9
8	5	10	91.55
9	5	15	90.2
10	6	5	90.77
11	6	10	88.04
12	6	15	86.31

На рисунке 5a (справа) показана вероятность верной классификации приложений для моделей DL и на базе SVM (список классов приложений приведен выше). Можно отметить, что точность классификации для трафика Интернет, почты, передачи данных, служебного трафика и сервисов относительно высока, независимо от того, какая модель используется. В связи с тем, что количество таких приложений в выборках составляет большинство в каждом наборе данных, у моделей больше данных для их идентификации.

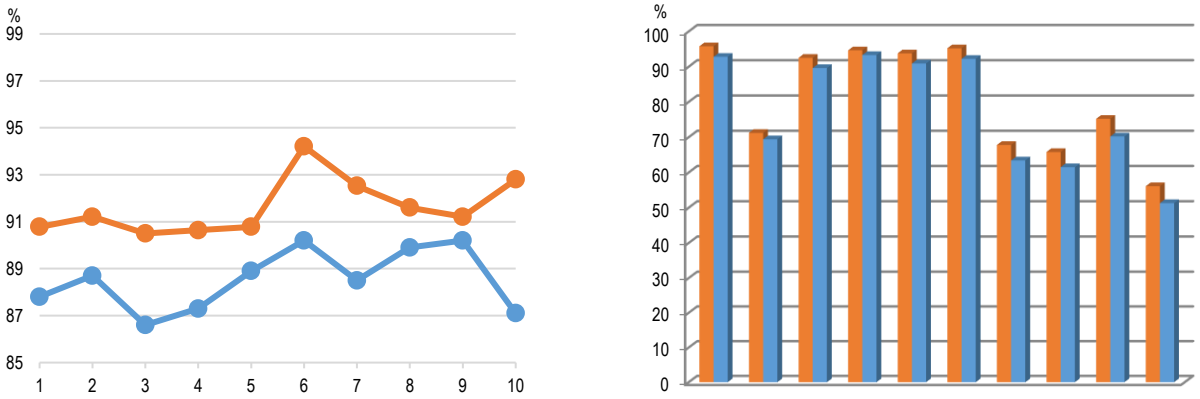
Кроме того, гибридная модель DL-сети, состоящая из многоуровневого автокодировщика и слоя регрессии softmax, более эффективна с точки зрения обучения и возможностей классификации. При этом стоит отметить, что отдельные классы трафика, такие как мультимедиа и базы данных, имеют невысокую вероятность верной классификации, что объясняется небольшим числом пакетов этих приложений в используемых наборах данных. Аналогичные графики для определения точности классификации (рисунок 5b) и полноты классификации (рисунок 5c) приведены ниже.

На рисунке 5c (справа) можно констатировать, что полнота классификации DL-модели выше, чем на базе SVM для различных сетевых приложений. Возможно, это связано с бóльшим числом скрытых слоев, которые обеспечивают дополнительные возможности к обобщению и принятию решения, а также эффективности извлечения признаков и абстракции приложений в трафике.

a) Вероятность верной классификации
Accuracy Comparison



b) Точность классификации
Precision Comparison



c) Полнота классификации
Recall Comparison

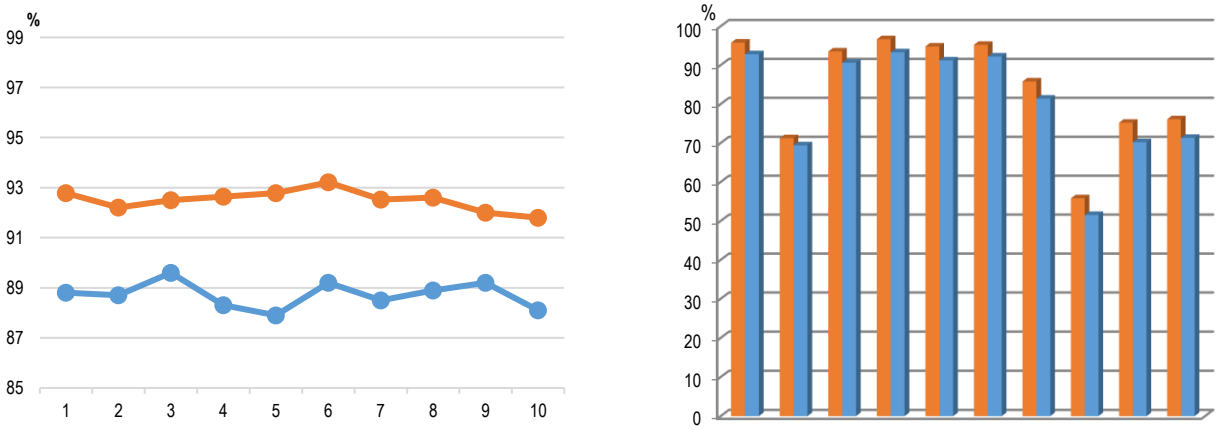


Рис. 5. Сравнение классификаций характеристик для моделей глубокого обучения и на базе SVM (для различных наборов – слева, в рамках каждого класса – справа)

Fig 5. Comparison between Deep Learning and Support Vector Machine: (left – for different data sets, right – for different applications)

Заключение

Применение архитектуры SDN позволило разработать эффективное применение новой системы классификации сетевых приложений на основе гибридной сети глубокого обучения, состоящей из многоуровневого автокодировщика и слоя регрессии softmax. Благодаря логическому централизованному управлению в SDN и мощным вычисли-

тельным возможностям потоки сетевых приложений могут собираться и обрабатываться в контроллере. В дальнейшем простой многоуровневый автокодировщик используется для получения признаков потока, а также выделения признаков более высокого уровня, а уровень регрессии softmax – в качестве классификатора для итоговой идентификации сетевых приложений.

Отдельно стоит отметить применение немаркированных данных для обучения многоуровневого автокодировщика и набора размеченных данных для обучения слоя softmax, а также эффективное применение метода обратного распространения ошибки для тонкой донастройки всей гибридной ГНС.

Несмотря на то, что в статье предложена структура классификации приложений, основанная на архитектуре SDN и технологии DL, основной акцент сделан на подтверждение эффективности и оценку предложенного метода классификации приложений на основе глубокого обучения в соответствии с имеющимся набором данных. Результаты экспериментов указывают на более высокие комплексные характеристики точности классификации по сравнению с моделью на базе SVM.

С развитием гетерогенных сетей, при взрывном росте сетевого трафика, сетевые приложения становятся все более диверсифицированными, и, хотя предлагаемый метод обеспечивает классификацию приложений, остаются нерешенными отдельные прикладные вопросы. Один из них связан с созданием предварительных размеченных наборов сетевого трафика, получение которых в реальной сети, с учетом большого числа разнообразных приложений, достаточно сложно. Поэтому в дальнейшем рассматривается возможность использования алгоритма глубокого обучения без учителя для классификации приложений. Кроме того, необходимо снизить временные затраты на обучение ГНС, которые на данный момент достаточно велики, по сравнению с сетевой задержкой.

Список источников

1. Елагин В.С. Динамическое управление нагрузкой в программно-конфигурируемых сетях // Труды учебных заведений связи. 2017. Т. 3. № 3. С. 60–67.

2. Елагин В.С., Дмитриева Ю.С. Моделирование сетевого ресурса в программно-конфигурируемых сетях // Вестник связи. 2020. № 6. С. 35–40.

3. Zhang J., Chen X., Xiang Y., Zhou W., Wu J. Robust Network Traffic Classification // IEEE /ACM Transactions on Networking. 2015. Vol. 23. Iss. 4. PP. 1257–1270. DOI:10.1109/TNET.2014.2320577

4. Kim H., Claffy K.C., Fomenkov M., Barman D., Faloutsos M., Lee K. Internet traffic classification demystified: myths, caveats, and the best practices // Proceedings of the Conference on emerging Networking EXperiments and Technologies (Madrid, Spain, 9–12 December 2008). New York: Association for Computing Machinery, 2008. DOI:10.1145/1544012.1544023

5. Auld T., Moore A.W., Gull S.F. Bayesian Neural Networks for Internet Traffic Classification // IEEE Transactions Neural Networ. 2007. Vol. 18. Iss. 1. PP. 223–239. DOI:10.1109/TNN.2006.883010

6. Nguyen T.T.T., Armitage G. A survey of techniques for internet traffic classification using machine learning // IEEE Communication Survive Tutorials. 2008. Vol. 10. Iss. 4. PP. 56–76. DOI:10.1109/SURV.2008.080406

7. Valenti S., Rossi D., Dainotti A., Pescapè A., Finamore A., Mellia M. Reviewing Traffic Classification // Biersack E., Callegari C., Matijasevic M. (eds) Data Traffic Monitoring and Analysis. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2013. Vol. 7754. PP. 123–147. DOI:10.1007/978-3-642-36784-7_6

8. Zhang J., Chen C., Xiang Y., Zhou W., Xiang Y. Internet Traffic Classification by Aggregating Correlated Naive Bayes Predictions // IEEE Transactions on Information Forensics and Security. 2013. Vol. 8. Iss. 1. PP. 5–15. DOI:10.1109/TIFS.2012.2223675

9. Grimaudo L., Mellia M., Baralis E., Keralapura R. SeLeCT: Self-Learning Classifier for Internet Traffic // IEEE Transactions Network Service Management. 2014. Vol. 11. Iss. 2. PP. 144–157. DOI:10.1109/TNSM.2014.011714.130505

10. Cao J., Fang Z., Qu G., Sun H., Zhang D. An accurate traffic classification model based on support vector machines // International Journal of Network Management. 2017. Vol. 27. Iss. 1. P. e1962. DOI:10.1002/nem.1962

11. Pasca S.T.V., Prasad S.S., Kataoka K. AMPF: Application-aware Multipath Packet Forwarding using Machine Learning and SDN // arXiv:1606.05743. 2016. DOI:10.48550/arXiv.1606.05743

12. Amaral P., Dinis J., Pinto P., Bernardo L., Tavares J., Mamede H.S. Machine Learning in Software Defined Networks: Data Collection and Traffic Classification // Proceedings of the 24th International Conference on Network Protocols (ICNP, Singapore, 08–11 November 2016). IEEE, 2016. DOI:10.1109/ICNP.2016.7785327

13. Wang P., Lin S.C., Luo M. A Framework for QoS-aware Traffic Classification Using Semi-supervised Machine Learning in SDNs // Proceedings of the International Conference on Services Computing (SCC, San Francisco, USA, 27 June – 02 July 2016). IEEE, 2016. DOI:10.1109/SCC.2016.133

14. LeCun Y., Bengio Y., Hinton G. Deep learning // Nature. 2015. Vol. 521. Iss. 7553. PP. 436–444. DOI:10.1038/nature14539

15. Chen X.W., Lin X. Big Data Deep Learning: Challenges and Perspectives // IEEE Access. 2014. Vol. 2. PP. 514–525. DOI:10.1109/ACCESS.2014.2325029

16. Kreutz D., Ramos F.M.V., Verissimo P.E., Rothenberg C.E., Azodolmolky S., Uhlig S. Software-Defined Networking: a Comprehensive Survey // Proceedings of the IEEE. 2015. Vol. 103. Iss. 1. PP. 14–76. DOI:10.1109/JPROC.2014.2371999

17. Bu C., Wang X., Cheng H., Huang M., Li K., Das S. Enabling Adaptive Routing Service Customization via the Integration of SDN and NFV // Journal of Network Computing Applications. 2017. Vol. 93. PP. 123–136. DOI:10.1016/j.jnca.2017.05.010

18. Yi B., Wang X., Huang M. Design and evaluation of schemes for provisioning service function chainwith function scalability // Journal of Network Computing Applications. 2017. Vol. 93. PP. 197–214. DOI:10.1016/j.jnca.2017.05.013
19. Lv J., Wang X., Huang M., Shi J., Li K., Li J. RISC: ICN routing mechanism incorporating SDN and community division // Computing Network. 2017. Vol. 123. PP. 88–103. DOI:10.1016/j.comnet.2017.05.010
20. He Q., Wang X., Huang M. OpenFlow-based low-overhead and high-accuracy SDN measurement framework // Transactions on Emerging Telecommunications Technologies. 2018. Vol. 29. Iss. 2. P. e3263. DOI:10.1002/ett.3263
21. Yi B., Wang X., Li K., Das S.K., Huang M. A comprehensive survey of Network Function Virtualization // Computing Network. 2018. Vol. 133. PP. 212–262. DOI:10.1016/j.comnet.2018.01.021
22. Shu Z., Wan J., Lin J., Wang S., Li D., Rho S., et al. Traffic engineering in software-defined networking: Measurement and management // IEEE Access. 2016. Vol. 4. PP. 3246–3256. DOI:10.1109/ACCESS.2016.2582748
23. Cui L., Yu F.R., Yan Q. When big data meets software-defined networking: SDN for big data and big data for SDN // IEEE Network. 2016. Vol. 30. Iss. 1. PP. 58–65. DOI:10.1109/MNET.2016.7389832
24. Zhang L., Huang H., Jing X. A modified cyclostationary spectrum sensing based on softmax regression model // Proceedings of the 16th International Symposium on Communications and Information Technologies (ISCIT, Qingdao, China, 26–28 September 2016). IEEE, 2016. DOI:10.1109/ISCIT.2016.7751707
25. Zhang H., Lu G., Qassrawi M.T., Zhang Y., Yu X. Feature selection for optimizing traffic classification // Computing Communicdtion. 2012. Vol. 35. Iss. 12. PP. 1457–1471. DOI:10.1016/j.comcom.2012.04.012
26. da Silva A.S., Machado C.C., Bisol R.V., Granville L.Z., Schaeffer A. Identification and Selection of Flow Features for Accurate Traffic Classification in SDN // Proceedings of the 14th International Symposium on Network Computing and Applications (NCA, Cambridge, USA, 28–30 September 2015). IEEE, 2015. DOI:10.1109/NCA.2015.12
27. Schmidhuber J. Deep learning in neural networks: an overview // Neural Network. 2015. Vol. 61. PP. 85–117. DOI:10.1016/j.neunet.2014.09.003
28. Salama M.A., Eid H.F., Ramadan R.A., Darwish A., Hassanien E. Hybrid Intelligent Intrusion Detection Scheme // Gaspar-Cunha A., Takahashi R., Schaefer G., Costa L. (eds) Soft Computing in Industrial Applications. Advances in Intelligent and Soft Computing. Berlin, Heidelberg: Springer, 2011. Vol. 96. PP. 293–303. DOI:10.1007/978-3-642-20505-7_26
29. Fiore U., Palmieri F., Castiglione A., De Santis A. Network anomaly detection with the restricted Boltzmann machine // Neurocomputing. 2013. Vol. 122. PP. 13–23. DOI:10.1016/j.neucom.2012.11.050
30. Lv Y., Duan Y., Kang W., Li Z., Wang F.Y. Traffic Flow Prediction with Big Data: a Deep Learning Approach // IEEE Transactions Intelligent Transport System. 2015. Vol. 16. Iss. 2. PP. 865–873. DOI:10.1109/TITS.2014.2345663
31. Yang H.F., Dillon T.S., Chen Y.P. Optimized Structure of the Traffic Flow Forecasting Model with a Deep Learning Approach // IEEE Transactions on Neural Networks and Learning Systems. 2017. Vol. 28. Iss. 10. PP. 2371–2381. DOI:10.1109/TNNLS.2016.2574840
32. Huang W., Song G., Hong H., Xie K. Deep Architecture for Traffic Flow Prediction: Deep Belief Networks with Multitask Learning // IEEE Transactions Intelligent Transport System. 2014. Vol. 15. Iss. 5. PP. 2191–2201. DOI:10.1109/TITS.2014.2311123
33. Bengio Y., Lamblin P., Popovici D., Larochelle H. Greedy Layer-Wise Training of Deep Networks // Proceedings of the Conference on Advances in Neural Information Processing Systems 19 (2006). MIT Press, 2007. PP. 153–160.
34. BRASIL. Characterizing Network-based Applications. Data sets // University of Cambridge Computer Laboratory. URL: https://www.cl.cam.ac.uk/research/srg/netos/projects/brasil/data/index.html (дата обращения 15.06.2023)

References

1. Elagin V. Dynamic Load Balancing in Software-Defined Network. Proceedings of Telecommun. Univ. 2017;3(3):60–67.
2. Elagin V.S., Dmitrieva Yu.S. The Modeling of Network Resources in Software-Defined Networks. *Vestnik Communications.* 2020;6:35–40.
3. Zhang J., Chen X., Xiang Y., Zhou W., Wu J. Robust Network Traffic Classification. *IEEE /ACM Transactions on Networking*. 2015;23(4):1257–1270. DOI:10.1109/TNET.2014.2320577
4. Kim H., Claffy K.C., Fomenkov M., Barman D., Faloutsos M., Lee K. Internet traffic classification demystified: myths, caveats, and the best practices. *Proceedings of the Conference on emerging Networking EXperiments and Technologies, 9–12 December 2008, Madrid, Spain*. New York: Association for Computing Machinery; 2008. DOI:10.1145/1544012.1544023
5. Auld T., Moore A.W., Gull S.F. Bayesian Neural Networks for Internet Traffic Classification. *IEEE Transactions Neural Networ*. 2007;18(1):223–239. DOI:10.1109/TNN.2006.883010
6. Nguyen T.T.T., Armitage G. A survey of techniques for internet traffic classification using machine learning. *IEEE Communication Survive Tutorials*. 2008;10(4):56–76. DOI:10.1109/SURV.2008.080406
7. Valenti S., Rossi D., Dainotti A., Pescapè A., Finamore A., Mellia M. Reviewing Traffic Classification. *In: Biersack E., Callegari C., Matijasevic M. (eds) Data Traffic Monitoring and Analysis. Lecture Notes in Computer Science, vol.7754*. Berlin, Germany: Springer; 2013. p.123–147. DOI:10.1007/978-3-642-36784-7_6
8. Zhang J., Chen C., Xiang Y., Zhou W., Xiang Y. Internet Traffic Classification by Aggregating Correlated Naive Bayes Predictions. *IEEE Transactions on Information Forensics and Security*. 2013:8(1):5–15. DOI:10.1109/TIFS.2012.2223675
9. Grimaudo L., Mellia M., Baralis E., Keralapura R. SeLeCT: Self-Learning Classifier for Internet Traffic. *IEEE Transactions Network Service Management*. 2014;11(2):144–157. DOI:10.1109/TNSM.2014.011714.130505
10. Cao J., Fang Z., Qu G., Sun H., Zhang D. An accurate traffic classification model based on support vector machines. *International Journal of Network Management*. 2017;27(1):e1962. DOI:10.1002/nem.1962
11. Pasca S.T.V., Prasad S.S., Kataoka K. AMPF: Application-aware Multipath Packet Forwarding using Machine Learning and SDN. *arXiv:1606.05743*. 2016. DOI:10.48550/arXiv.1606.05743

12. Amaral P., Dinis J., Pinto P., Bernardo L., Tavares J., Mamede H.S. Machine Learning in Software Defined Networks: Data Collection and Traffic Classification. *Proceedings of the 24th International Conference on Network Protocols, ICNP, 08–11 November 2016, Singapore*. IEEE; 2016. DOI:10.1109/ICNP.2016.7785327
13. Wang P., Lin S.C., Luo M. A Framework for QoS-aware Traffic Classification Using Semi-supervised Machine Learning in SDNs. *Proceedings of the International Conference on Services Computing, SCC, 27 June – 02 July 2016, San Francisco, USA*. IEEE; 2016. DOI:10.1109/SCC.2016.133
14. LeCun Y., Bengio Y., Hinton G. Deep learning. *Nature*. 2015;521(7553):436–444. DOI:10.1038/nature14539
15. Chen X.W., Lin X. Big Data Deep Learning: Challenges and Perspectives. *IEEE Access*. 2014;2:514–525. DOI:10.1109/ACCESS.2014.2325029
16. Kreutz D., Ramos F.M.V., Verissimo P.E., Rothenberg C.E., Azodolmolky S., Uhlig S. Software-Defined Networking: a Comprehensive Survey. *Proceedings of the IEEE*. 2015;103(1):14–76. DOI:10.1109/JPROC.2014.2371999
17. Bu C., Wang X., Cheng H., Huang M., Li K., Das S. Enabling Adaptive Routing Service Customization via the Integration of SDN and NFV. *Journal of Network Computing Applications*. 2017;93:123–136. DOI:10.1016/j.jnca.2017.05.010
18. Yi B., Wang X., Huang M. Design and evaluation of schemes for provisioning service function chainwith function scalability. *Journal of Network Computing Applications*. 2017;93:197–214. DOI:10.1016/j.jnca.2017.05.013
19. Lv J., Wang X., Huang M., Shi J., Li K., Li J. RISC: ICN routing mechanism incorporating SDN and community division. *Computing Network*. 2017;123:88–103. DOI:10.1016/j.comnet.2017.05.010
20. He Q., Wang X., Huang M. OpenFlow-based low-overhead and high-accuracy SDN measurement framework. *Transactions on Emerging Telecommunications Technologies*. 2018;29(2):e3263. DOI:10.1002/ett.3263
21. Yi B., Wang X., Li K., Das S.K., Huang M. A comprehensive survey of Network Function Virtualization. *Computing Network*. 2018;133:212–262. DOI:10.1016/j.comnet.2018.01.021
22. Shu Z., Wan J., Lin J., Wang S., Li D., Rho S., et al. Traffic engineering in software-defined networking: Measurement and management. *IEEE Access*. 2016;4:3246–3256. DOI:10.1109/ACCESS.2016.2582748
23. Cui L., Yu F.R., Yan Q. When big data meets software-defined networking: SDN for big data and big data for SDN. *IEEE Network.* 2016;30(1):58–65. DOI:10.1109/MNET.2016.7389832
24. Zhang L., Huang H., Jing X. A modified cyclostationary spectrum sensing based on softmax regression model. *Proceedings of the 16th International Symposium on Communications and Information Technologies, ISCIT, 26–28 September 2016, Qingdao, China*. IEEE; 2016. DOI:10.1109/ISCIT.2016.7751707
25. Zhang H., Lu G., Qassrawi M.T., Zhang Y., Yu X. Feature selection for optimizing traffic classification. *Computing Communicdtion*. 2012;35(12):1457–1471. DOI:10.1016/j.comcom.2012.04.012
26. da Silva A.S., Machado C.C., Bisol R.V., Granville L.Z., Schaeffer A. Identification and Selection of Flow Features for Accurate Traffic Classification in SDN. *Proceedings of the 14th International Symposium on Network Computing and Applications, USA, 28–30 September 2015, NCA, Cambridge*. IEEE; 2015. DOI:10.1109/NCA.2015.12
27. Schmidhuber J. Deep learning in neural networks: an overview. *Neural Network*. 2015;61:85–117. DOI:10.1016/j.neunet.2014.09.003
28. Salama M.A., Eid H.F., Ramadan R.A., Darwish A., Hassanien E. Hybrid Intelligent Intrusion Detection Scheme. *In: Gaspar-Cunha A., Takahashi R., Schaefer G., Costa L. (eds) Soft Computing in Industrial Applications. Advances in Intelligent and Soft Computing, vol.96*. Berlin, Heidelberg: Springer; 2011. p.293–303. DOI:10.1007/978-3-642-20505-7_26
29. Fiore U., Palmieri F., Castiglione A., De Santis A. Network anomaly detection with the restricted Boltzmann machine. *Neurocomputing*. 2013;122:13–23. DOI:10.1016/j.neucom.2012.11.050
30. Lv Y., Duan Y., Kang W., Li Z., Wang F.Y. Traffic Flow Prediction with Big Data: a Deep Learning Approach. *IEEE Transactions Intelligent Transport System*. 2015;16(2):865–873. DOI:10.1109/TITS.2014.2345663
31. Yang H.F., Dillon T.S., Chen Y.P. Optimized Structure of the Traffic Flow Forecasting Model with a Deep Learning Approach. *IEEE Transactions on Neural Networks and Learning Systems*. 2017;28(10):2371–2381. DOI:10.1109/TNNLS.2016.2574840
32. Huang W., Song G., Hong H., Xie K. Deep Architecture for Traffic Flow Prediction: Deep Belief Networks with Multitask Learning. *IEEE Transactions Intelligent Transport System*. 2014;15(5):2191–2201. DOI:10.1109/TITS.2014.2311123
33. Bengio Y., Lamblin P., Popovici D., Larochelle H. Greedy Layer-Wise Training of Deep Networks. *Proceedings of the Conference on Advances in Neural Information Processing Systems 19, 2006*. MIT Press; 2007. p.153–160.
34. *University of Cambridge Computer Laboratory*. BRASIL. Characterizing Network-based Applications. Data sets. URL: https://www.cl.cam.ac.uk/research/srg/netos/projects/brasil/data/index.html [Accessed 15.06.2023]

Статья поступила в редакцию 26.07.2023; одобрена после рецензирования 21.08.2023; принята к публикации 28.08.2023.
The article was submitted 26.07.2023; approved after reviewing 21.08.2023; accepted for publication 28.08.2023.

Информация об авторе:

ЕЛАГИН Василий Сергеевич
кандидат технических наук, доцент, доцент кафедры Инфокоммуникационных систем Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича
https://orcid.org/0000-0003-4077-6869