Machine Learning-Based Traffic Classification in Software-Defined Networking: A Systematic Literature Review, Challenges, and Future Research Directions

Daniel Nuñez-Agurto, Walter Fuertes, Luis Marrone, and Mayra Macas

Abstract—**The growing diversification of Internet applications and the continuous evolution of network infrastructure with emerging technologies have complicated network management. Network traffic classification is a key enabler for managing network resources according to the quality of service and security requirements. However, traditional traffic classification methods based on Deep Packet Inspection do not meet the stringent scalability, security, and privacy requirements. The centralized controller of Software-Defined Networking offers a global vision of the network, facilitating traffic analysis and providing direct programming capabilities. Traffic flows can be dynamically adjusted to satisfy the changing network requirements. These characteristics, along with the application of Machine Learning techniques have made it possible to integrate intelligence into networks, optimize them, and better manage and maintain them. In this context, this work aims to provide a Systematic Literature Review on traffic classification in Software-Defined Networking with Machine Learning techniques. Furthermore, we analyze and organize the selected seminal works based on the categorization of traffic classes and the employed Machine Learning techniques to draw meaningful research conclusions. Finally, we identify new challenges and future research directions on this topic.**

***Index Terms*—Machine learning, deep learning,software-defined networking, traffic classification**

I. Introduction

THE number of connections and devices worldwide is growing faster than the population and the Internet users, increasing network traffic exponentially [1]. Furthermore, the adoption of new devices with significant capabilities and intelligence (e.g., smartphones, smart televisions, video game consoles), combined with the proliferation of Machine-to-Machine (M2M) communications and the consequent development of new services and applications, have significantly changed traffic flow patterns and network performance. Managing different network infrastructures to satisfy the requirements of new devices, applications, and services has become a complex task. Therefore, there is great interest in building autonomous networks, characterized by their self-configuring, self-repairing, self-optimizing, and self-protecting abilities, using cognitive techniques and Machine Learning (ML) [2].

The deployment of cognitive networks to address the network operation and management complexity has been extensively investigated. However, the integration of ML brings some new challenges. First, each organization has its own network scheme, and no standards are applied to establish uniformity between networks. Second, the distributed implementation of traditional network systems dictates that each network node, such as a router or switch, can only see and execute actions on a particular network segment. However, the emerging technological advances in networks, such as the programmability achieved through Software-Defined Networking (SDN) and Network Functions Virtualization (NFV), enable ML to automatically discover patterns, trends, and relationships in network data [3].

Software-Defined Networking is a network paradigm that decouples the control plane from the data plane [4]. The centralized control plane is responsible for the routing and policy management of the network. Therefore, it has a global vision of the network by monitoring and collecting its status in real time. As such, it has enabled many novel network monitoring techniques [85]. The data plane takes care of forwarding, deleting, and modifying traffic flows based on the controller's instructions. Machine Learning aims to identify and exploit hidden patterns in data to deduce knowledge and is successfully employed in pattern recognition and anomaly detection problems. This capability enables the automation of complex tasks such as traffic classification, resource management, security, and general network administration. Therefore, the integration of ML in SDN is a research area of great interest, promising new ways to address traditional network problems by using novel data-driven techniques.

Traffic classification is an intelligent task that refers to categorizing traffic in different classes, and it is used, among other things, for network management, service measurement, and network monitoring. Moreover, traffic classification allows for efficient resource allocation and configuration of access controls, quality of service (QoS), and other network security parameters. The widely used traffic classification techniques include the port-based approach and Deep Packet Inspection (DPI) [21], [22]. However, most applications run on dynamic ports nowadays and network traffic is encrypted,

Manuscript received December 30, 2021; revised October 02, 2022. This research was supported in part by Universidad de las Fuerzas Armadas ESPE, Ecuador and Universidad Nacional de La Plata, Argentina.

Daniel Nuñez-Agurto is a Ph.D. candidate of Faculty of Computer Science, Universidad Nacional de La Plata, Argentina. He is also a professor at the Department of Computer Science, Universidad de las Fuerzas Armadas ESPE, Ecuador (e-mail: adnunez1@espe.edu.ec).

Walter Fuertes is a professor at the Department of Computer Science, Universidad de las Fuerzas Armadas ESPE, Ecuador (e-mail: wmfuertes@espe.edu.ec).

Luis Marrone is a professor at the Faculty of Computer Science, Universidad Nacional de La Plata, Argentina (e-mail: lmarrone@linti.unlp.edu.ar).

Mayra Macas is a Ph.D. candidate at the College of Computer Science and Technology, Zhejiang University, China (e-mail: mayramacas@ieee.org).

making both techniques no longer effective. Therefore, it is necessary to develop a new classification technique better suited to the current operational conditions.

This paper comprises a Systematic Literature Review (SLR) of traffic classification with ML techniques in SDN. We present an overview of SDN and the most commonly used ML techniques, followed by a comprehensive and comparative analysis of recent studies in the field. The main contribution of this work can be summarized as the systematic collection of relevant empirical evidence and the critical reviewing of state-of-the-art traffic classification methods in SDN, aiming to identify current limitations and suggest future research directions to address them.

The remainder of this paper is organized as follows: Section II briefly overviews related work. Then, the basic concepts of SDN and the most popular ML techniques are presented in Section III. In Section IV, the methodology followed to develop the SLR is described in detail. Section V presents the results and discusses how ML algorithms are applied in SDN from the perspective of traffic classification, while also identifying the limitations of the surveyed techniques. In Section VI, we summarize the outcomes of our analysis and discuss future work. Finally, Section VII concludes this paper. Table 1 provides the list of acronyms used in this manuscript to facilitate reading.

TABLE I
LIST OF ACRONYMS USED IN THIS MANUSCRIPT

CNN	Convolutional Neural Network
DPI	Deep Packet Inspection
DNN	Deep Neural Network
DT	Decision Tree
IoT	Internet of Things
k-NN	k-Nearest Neighbor
LSTM	Long Short-Term Memory
M2M	Machine-to-Machine
ML	Machine Learning
MLP	Multilayer Perceptron
NFV	Network Function Virtualization
NN	Neural Network
QoS	Quality of Service
QoE	Quality of Experience
ONF	Open Networking Foundation
RandNN	Random Neural Network
RNN	Recurrent Neural Network
RF	Random Forest
RL	Reinforcement Learning
SAE	Stacked Autoencoder
SDN	Software-Defined Networking
SVM	Support Vector Machine
SOM	Self-Organizing Map
SAE	Stacked Autoencoder
WAN	Wide Area Networks

II. RELATED WORK

This section briefly overviews previous works related to the research area of traffic classification in SDN with ML techniques, summarized in Table 2. Yan and Yuan [5] investigate emerging traffic classification methods in SDN. Boutaba et al. [6] analyze the application of ML techniques in different network technologies. Regarding traffic classification in SDN, the authors analyze the examined models' accuracy, employed dataset, traffic characteristics, and incorporated ML techniques. Xie et al. [7] conducted a comprehensive study on the use of ML algorithms in SDN in the context of traffic classification, routing optimization, quality of service (QoS)/quality of experience (QoE) prediction, resource management, and security. Mohammed et al. [8] analyze how ML techniques enable classification and traffic prediction in SDN, focusing on the use of Deep Learning for traffic prediction. Audah et al. [9] review the latest research on traffic classification that has been granted patents. The authors consider both traditional networks and SDNs in their study. Finally, Tamil and Thamilselvan [10] also focus on SDN traffic classification that leverages Deep Learning.

TABLE II
RELATED STUDIES

Ref.	Area of focus	Year
[5]	traffic classification	2018
[6]	application of diverse Machine Learning techniques in various key areas of networking across different network technologies	2018
[7]	traffic classification, routing optimization, QoS/QoE prediction, resource management, and security	2019
[8]	traffic classification and prediction	2019
[9]	traffic classification	2019
[10]	traffic classification	2020
This Study	SLR of traffic classification	2022

Even though the aforementioned surveys examine several works regarding traffic classification with ML techniques in SDN, they focus on different separate parts of this broad research topic. To address this issue, in this work, we attempt to thoroughly and systematically review the most recent state-of-the-art research, examining aspects not sufficiently covered before and highlighting the limitations identified by the comparative analysis of existing solutions, with the ultimate goal of suggesting future directions to mitigate them.

III. BACKGROUND

This section presents an overview of SDN and its architecture. Subsequently, it analyzes the main ML techniques enabling traffic classification in SDN and outlines the different traffic classification methods.

A. Software-Defined Networking

The Open Networking Foundation (ONF) [11] defines SDN as an emerging architecture that is dynamic, manageable, cost-effective, and adaptable, making it ideal for the high-bandwidth, dynamic nature of today's applications. This emerging paradigm decouples the control plane from the data plane, rendering the network directly programmable and enabling the underlying infrastructure to be abstracted for applications and services. The OpenFlow protocol is a foundational element for building SDN solutions [12], [13], [14], [15]. The general architecture of the SDN, the components and their interactions are shown in Fig.1.

1) Data Plane: The data plane, also known as the infrastructure plane, is the lowest layer of the SDN architecture and is responsible for forwarding, deleting, and modifying packets based on the controller's rules. This layer consists of network elements such as physical and virtual switches. Virtual switches [16] work on operating systems such as Linux. According to [7], [8], the virtual switches Indigo, Open

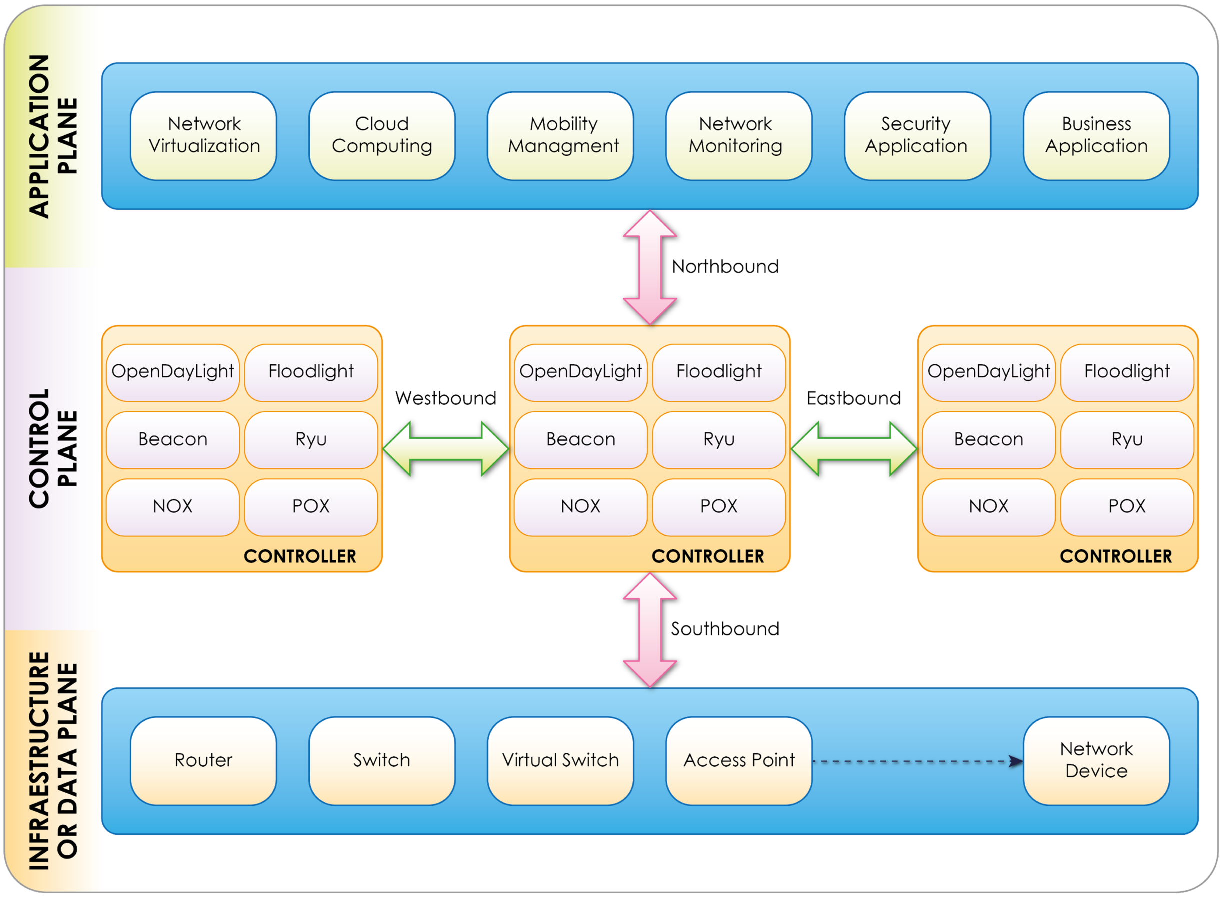

Fig. 1. Software-Defined Networking Architecture.

vSwitch, and Pantou are the most common implementations. The major telecom hardware vendors currently support SDN protocols and the corresponding switches. It should be noted that although virtual switches provide full implementations of the SDN protocols, physical switches may have limitations in the features they support. Communication between the data and control planes is achieved via the southbound communication interface (SBI).

2) Control Plane: The main component of the SDN architecture is the logically centralized control plane. This controller is responsible for dynamically scheduling network resources, updating data flow rules, and making flexible and agile network management [17]. Furthermore, it provides functionalities such as routing, storing the network topology, device configuration, and status information. Some open-source controllers are POX, NOX, Floodlight, Ryu, OpenDayLight, Trema, and Beacon. The three communication interfaces that allow the controllers to interact are: southbound, northbound, eastbound and westbound. The SDN controller provides a global view of the network, which enables traffic classification, among other applications [18].

3) Application Plane: The application plane is the highest layer of the SDN architecture. It is composed of business applications that provide new services. Applications obtain information from the network through the controllers' northbound interfaces. The best known applications at this level regard traffic engineering, security, distributed denial of service (DDoS) attacks, fault management, and QoS, among others [19].

SDN can be incorporated into a variety of different networks, such as Optical Networks, the Internet of Things (IoT), and Wide Area Networks (WAN), and can be used in combination with other enabling technologies like Cloud Computing and Network Function Virtualization (NFV).

B. Machine Learning in SDN: An Overview

ML is an application of artificial intelligence (AI) that allows the development of systems that learn autonomously by identifying complex patterns from large datasets. From an operational point of view, ML has two phases: 1) the first phase regards *training* and consists of providing ML algorithms with a subset of the employed dataset (called training set) from which the system model can learn, and 2) the second phase consists of *decision-making*, where the system can estimate the result of a new entry, based on the trained model. The ML algorithms are broadly classified into supervised, unsupervised, semi-supervised, and reinforcement learning [24], [25], [26], [27], [28]. The research efforts carried out during seven decades have resulted in many ML techniques. In Fig. 2, the most crucial milestones in the evolution of ML are highlighted.

Generally, ML is ideal for inferring solutions to problems with a large representative dataset. ML techniques are designed to identify hidden data and patterns in the data. Therefore, they are well suited for solving problems in SDNs. For example, a classification problem in SDN can be formulated to identify anomalous traffic. Fig. 3 presents the most commonly used ML techniques in SDN traffic classification.

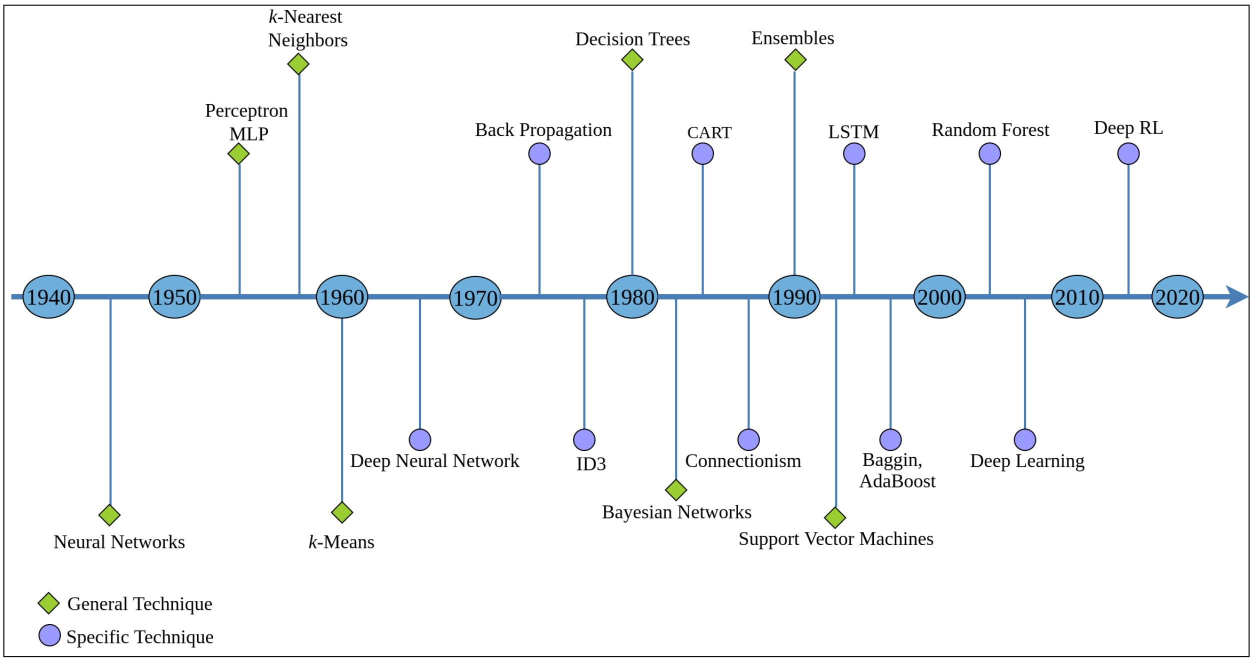

Fig. 2. Evolution of Machine Learning techniques with key milestones. Redrawn based on [6].

In the following, we list the main ML algorithms usually employed in networking, along with a brief description of their operation.

1) Supervised Learning: Supervised learning is an ML technique that allows for building a model from labeled datasets, meaning that the input and output data are known in advance. This type of learning aims to create a function where the input represents the analyzed characteristics, and the output represents the variables that have to be predicted. The output function is numerical in regression problems and categorical in classification problems [29]. The most commonly used algorithms in this category are listed below:

a) k-Nearest Neighbor(k-NN): This technique classifies a data sample based on its closest k-neighbors. The k-NN algorithm process is straightforward: if most of the nearest k-neighbors belong to a specific class, the unclassified sample will be assigned to that class. Since distance is the primary metric of the k-NN algorithm, several functions can be applied to define the distance between the unclassified sample and its neighbors, such as Chebyshev, City-block, Euclidean, and Euclidean squared [30].

b) Decision Tree (DT): It is a technique that performs the classification process through a learning tree. Each node in the tree represents a data feature, the branches represent the conjunctions of features that lead to classifications, and each leaf node is a class label. An unlabeled sample can be classified by comparing its characteristic values with the nodes in the DT [31]. The most commonly used algorithms to automatically classify a training dataset are ID3, C4.5, and CART [32].

c) Random Forest (RF): This technique is used for classification and regression tasks. It is an algorithm consisting of many decision trees. The random Forest method randomly chooses only a subset of the feature space to construct each Decision Tree [33]. In that way, it can attenuate the overfitting caused by decision trees to improve precision.

d) Support Vector Machine (SVM): It is an algorithm frequently used in classification and pattern recognition. SVM maps vectors in a high dimensional feature space, applying different core functions, such as linear and the radial base function (RBF). The selection of the core function depends on the training dataset and is a significant task in SVM that affects the classification accuracy [34], [35].

e) Naïve Bayes (NB): This technique applies the Bayes' theorem to calculate the probability of an event occurring, given prior knowledge of the conditions that could be related to the event. The advantage of this learning algorithm is that it requires a small training data set, considerably simplifying training by assuming the independence of attributes. Therefore, applying Bayes' theorem is easy and does not need iterative parameter estimation schemes, implying that it can be readily applied to large datasets [36], [37].

f) Neural Network (NN): The human brain inspires the concept of artificial neural networks. The NN nodes are the equivalent components of the neurons found in the human brain, and they execute highly complex non-linear and parallel calculations. More precisely, these nodes use activation functions to perform non-linear calculations, the most used of which are the sigmoid and the hyperbolic tangent. NN nodes are connected by variable link weights [38], [39]. The most frequently used NNs in SDN are the following:

- Random Neural Network (RandNN): RandNNs have been used in pattern classification and recognition. The main difference with other neural networks is that RandNN neurons exchange excitatory and inhibitory peak signals probabilistically [40].
- Deep Neural Network (DNN): It is an NN with multiple hidden layers between the input and output layers. The hierarchy of features makes DNNs capable of handling large, high-dimensional datasets. Due to the

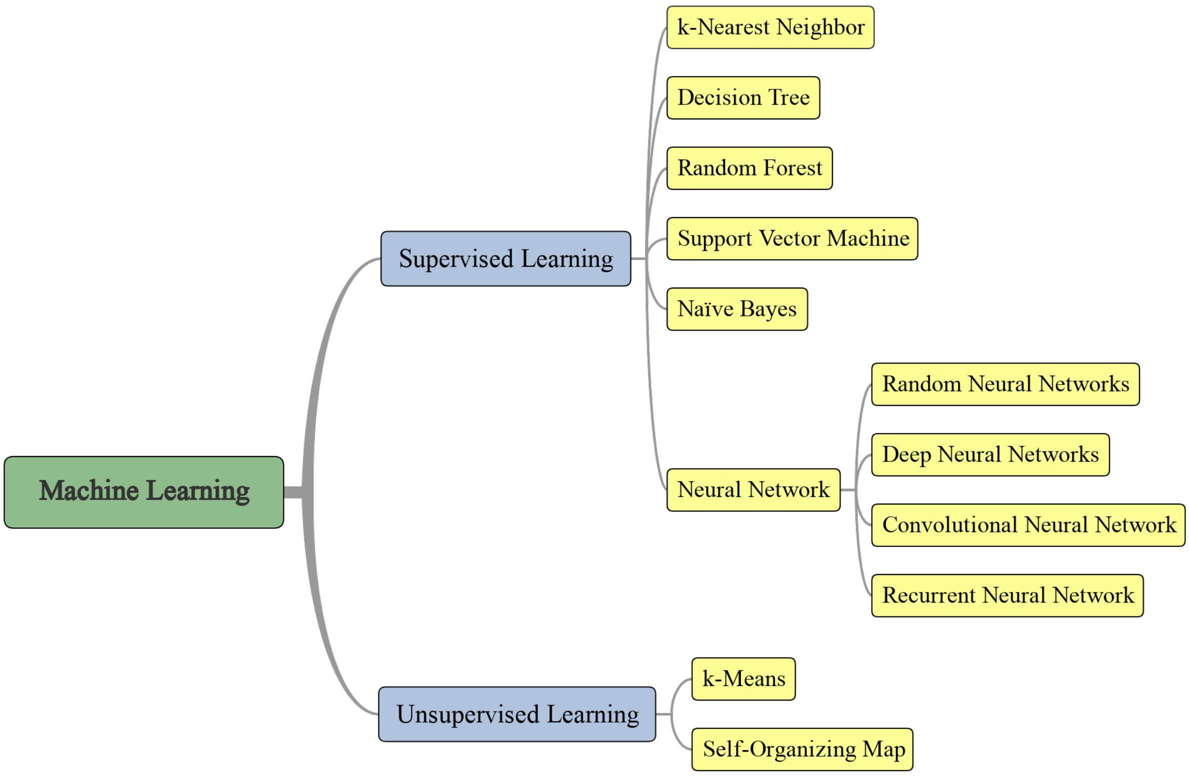

Fig. 3. ML techniques in Software-Defined Networking.

learning of multi-level feature representations, DNNs usually provide better performance compared to other ML techniques [41], [42], [43].

- Convolutional Neural Network (CNN): It is a neural network consisting of multiple layers of convolutional filters of one or more dimensions. Usually, after each layer, a function is added to perform the non-linear causal mapping. Scattered local connections between successive layers, weight distribution, and clustering are the three basic ideas of CNNs [44].
- Recurrent Neural Network (RNN): It is a stateful neural network that retains its input in its internal memory to handle sequential data. The behavior of an RNN is similar to that of a human brain in the sense that when it makes a decision, it is based on current information and previous experience acquired through loops. The most commonly used RNN implementation is Long Short-Term Memory, which back-propagates the errors through its layers to learn recurrently [45], [46].

2) Unsupervised Learning: Unsupervised learning algorithms receive a set of unlabelled input data. In other words, the input data is known, but there is no output data associated with a particular input. These algorithms are used to cluster unstructured data based on similarities and different patterns in the dataset. Therefore, they perform more complex processing tasks than supervised learning [47], [48]. The most commonly used unsupervised learning algorithm is *k-Means*. It is an algorithm used to classify an unlabeled dataset into different groups. In k-Means, it is only needed to set two parameters: the initial dataset and the desired number of groups [49].

3) Reinforcement learning: Reinforcement Learning (RL) is an iterative, agent-based process for modeling decision-making problems. Learning is based on training data samples. In RL, an agent interacts with the outside world, and instead of being trained by samples, it learns by exploring the environment and exploiting knowledge. Actions are rewarded or penalized. Therefore, the training data in the RL constitutes a set of state-action pairs and rewards or penalties. The agent uses the feedback from the environment to learn the best sequence of actions. RL can sacrifice immediate gains for long-term rewards when optimizing a cumulative reward. Therefore, RL is more suitable for making cognitive decisions, such as decision-making, planning, and scheduling [50], [51], [52].

IV. Methodology

This section presents the steps we followed to systematically examine and analyze existing work on traffic classification in SDN applying ML techniques. Kitchenham et al. [53] in their methodological guide define five steps for an SLR: 1) Define research questions; 2) Search for relevant documents; 3) Select primary studies; 4) Analyze abstracts and extract keywords and data; 5) Map selected primary studies.

A. Research Questions

This work is structured around the following research questions:

- **RQ1:** Which Machine Learning techniques are used to classify traffic in SDN?
- **RQ2:** What are the limitations of the current traffic classifiers when classifying traffic in SDNs?

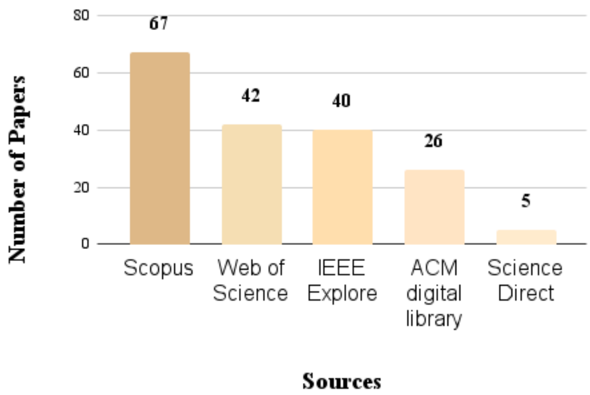

Fig. 4. Scientific databases used as sources.

B. Search string

The terms searched were machine learning techniques, traffic classification, and software-defined networking. The terms obtained from the control studies (i.e., the search string) are combined through logical operators: ”OR” to add synonyms and ”AND” to add new terms. The general string established for the search was:

((SDN OR (”Software Defined Network”) OR (”Software Defined Networking”)) AND ((”Traffic Classification”) OR (”Network Traffic Classification”) OR (”Internet Traffic Classification”)) AND ((”Machine Learning”) OR (”Deep Learning”) OR (”Artificial Intelligence”)))

C. Search Process

Several databases were consulted to gather the most relevant literature on traffic classification in SDN applying ML techniques, starting from 2014. The articles were examined following the identification of primary studies. The research procedure adopted in this article was extended with relevant documents from the following indexed databases (Fig. 4): Web of Science, Scopus, IEEE Explore, ACM Digital Library, and Science Direct.

D. Explicit inclusion criteria

Regarding the characteristics required for including an article, we established four inclusion criteria: 1) articles on traffic classification in SDN with ML techniques; 2) articles published after 2014; 3) articles presenting their content in English; 4) articles published in Journals and Conference Proceedings.

E. Explicit exclusion criteria

Four criteria are defined in the article exclusion process: 1) articles not included in the selected databases; 2) duplicate articles; 3) articles that are not written in English; 4) articles that are not published in Journals or Conference Proceedings.

F. Selection of primary studies

The procedure for any review article begins with the definition of the search string based on keywords. By applying the search string in the selected indexed databases, 180 articles were obtained. However, by eliminating duplicate articles,

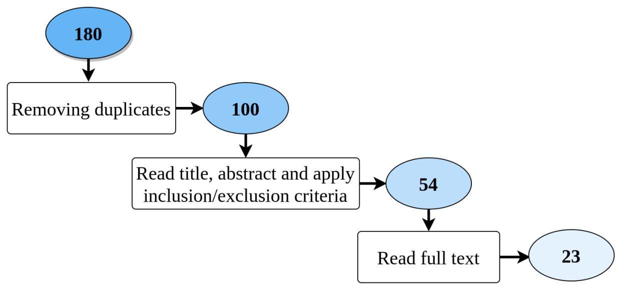

Fig. 5. Literature search process.

reading abstracts, reading the complete text, and applying the inclusion and exclusion criteria, 23 primary studies were selected, as illustrated in Fig. 5.

V. Results

This section analyzes, synthesizes, and discusses the works identified through conducting the SLR. Our main objective is to distill the established knowledge, recognize new contributions, and explore the alternative implementations of the leading ML algorithms used in traffic classification in SDN.

A. RQ1: Which Machine Learning techniques are used to classify traffic in SDN?

Traffic classification is fundamental for traffic analysis, enabling the management of different services and the efficient network resource allocation. Traffic classification requires accurately associating network traffic to predefined classes of interest. In general, we identified 23 relevant primary studies that use ML algorithms to classify traffic in SDN, as shown in Table 3. Furthermore, we identified about 16 ML algorithms used by researchers.

In particular, the most commonly used algorithms by researchers are: Support Vector Machine (8 times), Decision tree (7 times), k-NN (6 times), Convolutional Neural Networks (6 times), Naive Bayes (5 times), Random Forest (5 times), Deep Neural Networks (4 times), Multilayer Perceptrons (3 times), and Stacked Auto-Encoder (3 times), as depicted in Fig. 6. Moreover, the combination of different ML techniques was demonstrated to significantly improve the accuracy of the traffic classification task [56]. Finally, it is essential to note that as demonstrated in [76], the accuracy of the classification model can be improved with the introduction of reinforcement learning.

According to [20], depending on the degree of detail and the analysis capability, traffic classification can be further categorized as: 1) traffic clustering; 2) application type; 3) application protocol; 4) application software; 5) fine-grained, and 6) anomaly. Considering the nature of the classification process, Fig. 7 shows the categorization of the different traffic classification methods in the literature. As it can be observed, the most widely used methods are based on the application's traffic classification (five), and the traffic classification by type of application (twelve). With this in mind, the primary studies we identified are summarized below.

Application-aware traffic classification aims to identify applications based on traffic flow. Chang et al. [54] proposed an application-based online and offline traffic classifier

TABLE III
MACHINE LEARNING BASED TRAFFIC CLASSIFICATION IN SDNS

Ref.	Classification Level	ML Technique	Tool	Features Selection	Model Output	Accuracy
[54]	Application-aware	MLP, SAE and CNN	Tensorflow	5 flow features	6 applications	CNN: 93.35% MLP: 93.21% SAE: 93.13%
[55]	Application-aware	MLP, SAE and CNN	Keras and Tensor-flow	automatic by algorithm	15 applications	CNN: 99.30% SAE: 99.14% MLP: 97.14%
[56]	Application-aware	ML-LSTM and CNN-LSTM	Keras and Tensor-flow	automatic by algorithm	8 applications	ML-LSTM: 99.65% CNN-LSTM: 98.86%
[57]	Application-aware	RF	Not mentionet	12 flow features	8 applications	RF: 96.0%
[58]	Application-aware	CNN	Matlab	automatic by algorithm	6 applications	CNN: 99.0%
[59]	Application-type	k-NN, SVM, DT, RF, DNN and CNN	Scikit-learn and PyTorch	23 flow features	3 classes	k-NN: 98.00% RF: 97.00% CNN: 95.00% DNN: 94.00% DT: 88.00% SVM: 86.00%
[60]	Application-type	SAE	Weka and Matlab	automatic by algorithm	10 classes	SAE: 91.21%
[61]	Application-type	NB	Not mentioned	13 flow features	5 classes	NB: 97.6%
[62]	Application-type	SVM and K-Means	Not mentioned	30 flow features	8 classes	SVM: 98.7% K-M: 88.0%
[63]	Application-type	SVM, NB and NC	Scikit-learn	14 flow features	3 classes	NB: 96.79% SVM: 92.3% NC: 91.02%
[64]	Application-type	DT, k-NN, NB, and SVM	Not mentioned	features select by algorithm proposed	12 classes	DT: 99.39% k-NN: 98.34% NB: 96.71% SVM: 96.75%
[65]	Application-type	LapSVM	Not mentioned	9 flow features	4 classes	LapSVM: >90.0%
[66]	Application-type	SVM NC, B-NB and MC-SVM	Not mentioned	11 flow features	5 classes	NC, B-NB and MC-SVM: >90.0%
[67]	Application-type	DNN, SVM, k-NN and DT	TensorFlow and scikit-learn	9 flow features	4 classes	DNN: 88.00% DT: 85.0% SVM: 80.0% k-NN: 79.0%
[68]	Application-type	DNN	Not mentioned	automatic by algorithm	10 classes	DNN: 96.00%
[69]	Application-type	KNN, RF and DT	Scikit-learn	6 flow features	4 Classes	DT: 87.20% RF: 85.10% k-NN: 79.50%
[70]	Application-type	KNN, RF and DT	Scikit-learn and Keras	7 flow features	10 classes	k-NN: 97.14% RF: 96.69% DT: 95.80%
[71]	Fine-grained	CNN and autoen-coder, CNN, DNN	Tensorflow and Keras	flow characteristics select by autoencoders	24 applications	CNN and autoen-coder: 97.42% CNN: 96.03% DNN:94.36%
[72]	Fine-grained	RF and k-NN	Weka	flow characteristics by [24]	40 mobile ap-plications	RF: 95.5% k-NN: >90.0%
[73]	Fine-grained	DT	Weka	five tuple and set of statistical features	2 classes	DT: >90.0%
[74]	Fine-grained	ResNet and GRU	Not mentioned	8 flow features	2 classes	ResNet: >93.63% GRU: >86.53%
[75]	Anomaly	MLP	Keras and Tensor-Flow	22 flow features	2 classes	MLP: >96.0%
[76]	Anomaly	RL-RF, SVM, RF and NB	Not mentioned	13 flow features	2 classes	RL-RF: 99.54% SVM: 98.18% RF: 97.18% NB: 96.42%

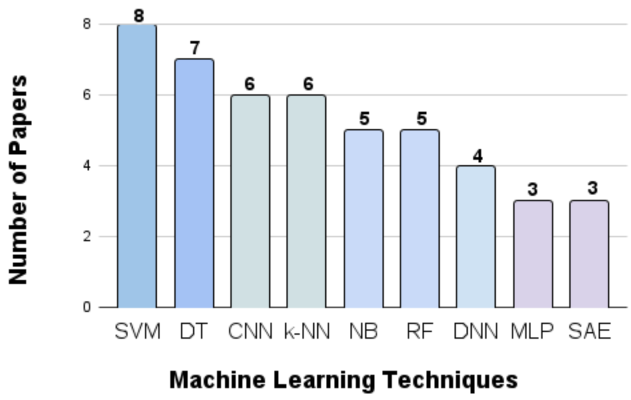

Fig. 6. Most commonly used Machine Learning algorithms in SDNs.

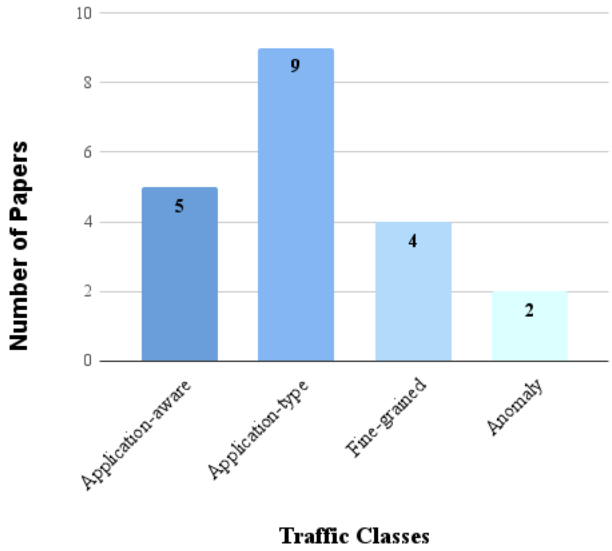

Fig. 7. Distribution of related studies to classification categories.

leveraging Deep Learning in SDN. The classifier is located in the SDN controller and consists of three deep learning models: MLP, CNN, and SAE. A TCP replay tool is used to emulate the online traffic with an open dataset including the seven most popular applications. The results show that the offline classifier achieved more than 93.00% accuracy, while the online classifier achieved around 87.00% accuracy in identifying the applications.

Wang et al. [55] proposed a framework for the classification of applications with encrypted data flows in smart home networks. The data are first preprocessed and then fed into three DL algorithms, namely Multilayer Perceptron, Stacked Auto-Encoder, and Convolutional Neural Network. An open dataset is used containing 15 applications and 200/,000 encrypted samples. The results show that the proposed framework can enable distributed application-aware classification in smart homes with an accuracy above 97,00% using the CNN model.

Lim et al. [56] proposed an application classification scheme using a DL model in SDN. The dataset is generated from the functional load of the flows by preprocessing the network traffic to train two DL models: (1) Multi-layer LSTM; and (2) a combination of convolutional neural network (CNN) with single-layer LSTM to perform the network traffic classification. A model tuning procedure is used to find the best hyper-parameter for the models. The results show that the multi-layer LSTM has the best performance, with an accuracy above 99.00%.

Amaral et al. [57] proposed an architecture that collects traffic data from an enterprise network using the OpenFlow protocol. This architecture can be used in SDNs and traditional networks. The authors used the gathered dataset with several ML algorithms to classify traffic flows in eight applications. The results show that the performance obtained using supervised learning is above 90.00%. Chen et al. [58] proposed a classification scheme of encrypted traffic to identify applications in an intelligent home gateway with the help of DL algorithms. The encrypted packet classifier has a two-level hierarchical structure. The first level consists of a classifier by service type based on applications with similar QoS requirements. The second level performs the classification of applications based on fine-grained traffic classification. The evaluation results were carried out at both levels to check the efficiency of the proposed classifiers; the accuracy obtained is over 85.00% for all applications.

The traffic classification by application-type seeks to identify the different traffic classes with similar QoS requirements in the network. Abdulrazzaq and Demirci [59] proposed a traffic engineering system in SDN to improve the quality of service based on deep learning techniques. The classifier performs the classification of traffic flows of various applications into classes with different priorities. The authors solved the issue of an imbalanced dataset by implementing an approach called Synthetic Minority Over-Sampling Technique. The better classifier performance results are obtained with traffic captured in 15s and 30s of the timeout with 1-D CNN and DNN, achieving and accuracy of over 95.00%.

Zhang et al. [60] proposed a DNN-based traffic classification method composed of SAE and a Softmax regression layer to identify applications in classes. The proposed framework uses the SDN controller to collect and process data from the network to train the hybrid DNN. The characteristics of the processed data flows are obtained automatically with the SAE model. The simulation results demonstrate that the proposed classifier effectively yields a classification accuracy above 90%.

Parzei et al. [61] introduced a method for classifying applications into traffic classes. The proposed method does not inspect the packet load, thus reducing the drivers' processing overhead and the network traffic for classification. Moreover, the authors employ ML algorithms to demonstrate that the proposed method improves classification accuracy. The experiments were conducted over an SDN-enabled network, including an OVS (Open V Switch) and two hosts. The results over the testing dataset show a better performance compared to the Naïve Bayes algorithm, reaching a 97.60% accuracy.

Fan and Liu [62] examined ML-based traffic classification techniques and how the models' fit and feature selection affect their performance. They focused on analyzing SVM and the K-means algorithms for classifying application traffic into ten traffic classes. The authors evaluated the performance of the SVM model with four different kernel functions: 1) linear, 2) polynomial, 3) sigmoid, and 4) radial. The results show that the SVM model based on radial based on kernel function provides the best accuracy over 95.00% and is more efficient computationally.

Raikar et al. [63] established a framework for the classification of applications in traffic classes with the application of three ML models: SVM, NC, and NB. The framework allows to capture network traffic traces, where the flow statistics are obtained with the netmate tool and the generated flows are sent to the classifier for classification. The results show that the best accuracy is 96.79% using the NB algorithm.

Zaki and Chin [64] recommended a hybrid method called Filter-Wrapper Feature Selection (FWFS) that is based on the selection of filtering and wrapping characteristics in order to improve the traffic classification of applications with traffic. This method reduces the number of dataset features at the beginning before the final features are selected with the C4.5 wrapper. The performance evaluation with feature selection is not expensive computationally, resulting in a reliable and stable model for classifying new data. The classifier has 98.90% accuracy.

Malik et al. [68] propose a new deep learning model for SDN that can accurately identify a wide range of traffic applications in a short time, called Deep-SDN. The proposed model can identify the applications-types with high accuracy and speed, making it applicable for online traffic identification. To avoid overfitting and improve the regularization, the authors implemented a dropout layer in the architecture. The evaluation of the model's performance shows 96,00% global accuracy.

Wang et al. [65] proposed a framework for traffic classification into four QoS classes. The framework consists of two components: 1) The first component is responsible for detecting significant flows for QoS in new incoming flows; 2) The second component carries out the classification of QoS traffic and related network management tasks. The framework assumes that applications with similar QoS characteristics have similar statistical properties. Therefore, different applications requiring similar QoS can be treated equally. The performance of the model shows an accuracy above 90.00%.

Amiri et al. [66] introduced a scheme of bandwidth utilization for game traffic. The proposed method uses ML algorithms to classify incoming traffic flows in real time while ensuring that game flows are prioritized over others, addressing in this way the bandwidth allocation problem of networks in cloud computing data centers. The simulation results in a realistic network demonstrate a good performance in network traffic classification accuracy, generally reducing the user-experienced delay by 8% compared to traditional methods.

Xu et al. [67] proposed a traffic classification mechanism that allows assigning different network resources to improve the QoS of distinct applications significantly. The mechanism is implemented using NFVs, which operate at the data plane. Therefore, sample data can be sent to the NFV via the SDN switch without using control channels. The experimental results show that the DNN model has an 87.00% accuracy, with a flow duration of 15s. Furthermore, the SDN controller assigns more appropriate route paths for different traffic classes and improves QoS.

Owusu and Nayak [69] presented a traffic classifier model. The employed ML learning models used the statistical features of the traffic data to classify the traffic into QoS requirements, latency, and bandwidth. Two feature selection methods, Shapley Additive Explanations (SHAP) and Sequential Feature Selection (SFS) were applied to the Random Forest and Decision Tree algorithm to improve the classifier's performance and shorten the training time by reducing the number of features. The results show that the RF algorithm with SFS has achieved the best accuracy (83.30%) using six features.

Mondal et al. [70] proposed a new performance accelerator algorithm (PAA), incorporating three ML classifiers to accelerate the overall performance significantly. To evaluate the performance, the authors introduced a new Docker-based SDN network system and implemented the proposed dynamic network classifier (DNC) in a Ryu controller, removing the burden of matching the incoming traffic manually and ensuring better QoS. The results show that the PPA improves the models' overall performance, reaching 99.29% of accuracy.

The fine-grained traffic classification consists of distinguishing the different traffic components of each application. Chiu et al. [71] proposed a framework called Convolutional Autoencoder Packer Classifier (CAPC) to classify incoming packets in fine-grained and coarse-grained manners based on DL. The classifier is a packet-based deep learning model consisting of a CNN and an Autoencoder. The experimental results show that CAPC classifies the traffic in a different type of service with 99,90% accuracy on the private dataset and 97.00% accuracy on the public dataset.

Uddin and Nadeem [72] proposed a traffic classification system for mobile applications in a wireless network with SDN support. The main component of the classification system is called TrafficVision Engine (TV Engine). This component's functions are: 1) collect, store, extract flow statistics and actual training data from the end and access devices; 2) use a DT classifier to detect the name of the applications; 3) apply a K-NN classifier to identify the flow types. The results show that the classifier achieves over 90% accuracy for most media flow types.

Xiao et al. [73] proposed a sensitive, low-cost classification method to classify elephant flows. The proposed detection strategy consists of two stages. In the first stage, suspicious elephant flows are distinguished from mice flows by measuring the packet header. In the second stage, a DT algorithm is used to classify whether the suspected elephant flows are elephant flows or not. The experiments on different settings and datasets demonstrate that the proposed strategy is efficient in detecting elephant flow, achieving an accuracy over 90.00%.

Liu et al. [74] introduced a classification scheme of two and four classes using Deep Learning (DL) to classify the flows as *cheetah, tortoise, porcupine, and elephant*. The pre-classification model uses a cost-sensitive Residual Neural Network (ResNet) plus A-Softmax to filter out most mice flows for the two-tier scheme. Also, the proposed accurate classification model is based on ResNet plus AM-Softmax to identify elephant flows. The Gated Recurrent Unit (GRU) detects cheetah, tortoise, porcupine, and elephant flows in the four classification scheme. The experimental results show that the 2-classification model accuracy is over 93.60%.

Anomaly traffic classification aims to discriminate between normal and abnormal traffic. Traffic anomalies can include network, transport, and application layer anomalies. Letteri et al. [75] proposed a methodology for detecting botnets based on traffic classification and ML techniques. The most significant features of botnets were extracted from a new dataset formed by samples of regular and botnet-generated traffic. The traffic classifier was based on the MLP algorithm. The experimental results show that the classifier has up to 96% accuracy. Xu et al. [76] proposed a defense strategy against DDoS attacks based on traffic classification. They used a Software-Defined Network Virtualization Architecture (SDNFV) and a traffic classification strategy to improve flexibility and reduce the SDN load under DDoS attacks. The results show that the implemented strategy reduces the risk of an attack on the SND controller. The classifier has been improved by introducing reinforcement learning, reaching an accuracy of 99.54%.

B. RQ2: What are the limitations of the current traffic classifiers when classifying traffic in SDNs?

Internet traffic is dynamic and managing it requires classifiers with ensemble classification principles and incremental learning to produce active and efficient systems. In order to classify different traffic flows efficiently, it is necessary to design algorithms that find unique and specific traffic characteristics, overcoming the variation of traffic flows, and thus, surpassing concept drift. Training traffic classification models needs to achieve a high degree of accuracy and precision.

Software-Defined Networking significantly extends network administration and allows for the management of different types of traffic flows. However, for the efficient realization of this process, the SDN controller requires a real-time classification method that is scalable, reliable, and can adjust to the future network growth. Additionally, a classifier at the controller level requires a significant number of resources given that all new flows must be analyzed for classification. Therefore, two alternative options should be considered when implementing an SDN traffic classifier: 1) implement the classifier at the control plane to categorize new traffic flows; or 2) implement the classifier at the data plane, improving network scalability by reducing the switch communications with the controller.

Regarding specific traffic classes, methods focused on fine-grained classification used for application identification can increase the traffic processing delays. Therefore, they are not suitable for high availability networks such as data center networks. Traffic classification based on application recognition is not a viable solution due to the constant increase of applications on the Internet, which makes it practically impossible to identify all applications. Classification based on application-type and anomaly traffic detection allow traffic flows to be assigned to different traffic classes. In this way, QoS can be assigned and security policies can be established in SDNs.

Traffic classification methods based on Machine Learning techniques are a viable alternative to traditional traffic classification methods in SDNs. However, several challenges must be overcome, such as computational complexity, classifier accuracy, training datasets with unbalanced classes, and concept drift. Concept drift refers to the change in relationships between input and output data over time, mainly in network traffic flows, causing a traffic classification model to become obsolete. This problem affects traffic classifiers based on supervised learning because they need a set of labeled historical data for training. On the other hand, the problem of unbalanced classes affects the algorithms in their information generalization process and has a negative impact on minority classes. As illustrated in Fig. 8, the most commonly used ML techniques in traffic classification are supervised learning. Nonetheless, these algorithms were not designed to work with unbalanced datasets.

TABLE IV
TRAFFIC DATASETS

S/N	Ref.	Dataset Source	No. of Instances	Size
1	[77]	Dataset provided by ISCX VPN-nonVPN	206688	28GB
2	[78]	Dataset provided by UPC's Broadband Communications Research Group	545438	32.61GB
3	[79]	Tor Traffic Dataset	250963	Not mentioned
4	[80]	University Network	377526	90MB
5	[81]	MAWI Working Group	Not mentioned	Not mentioned
6	[82]	University Data Center	Not mentioned	Not mentioned
7	[83]	HogZilla dataset	994490	140MB
8	[84]	UNB ISCX dataset	2450324	Not mentioned

The limited accessibility to datasets is another important impediment for traffic classification in SDNs. Currently, there are no publicly available SDN traffic datasets in the scientific community. Having up-to-date datasets is essential for evaluating traffic classification models because concept drift makes older datasets irrelevant. However, most of the analyzed papers could not make the datasets' sources available, while some mentioned the lack of a current dataset to test their proposed model. This systematic review summarized some of the datasets used by the authors in SDN traffic classification as listed in Table 4.

VI. DISCUSSION

The outcomes and the results of the conducted systematic review are essential for driving future research in ML regarding traffic classification to improve QoS and security in SDN. As can be seen, several different techniques have been used by researchers to a greater or lesser degree, attesting to the usefulness and suitability of ML for the particular task of traffic classification. The model accuracy of the various traffic classification techniques found in the literature has been evaluated and comparatively analyzed in this work. The main findings of this analysis will be described in the following.

This SLR identified about 15 ML techniques used by researchers in the selected primary studies, as shown in Fig. 8. More precisely, in [56], [71], the authors proposed the fusion of various models such as ML-LSTM, CNN-LSTM, and

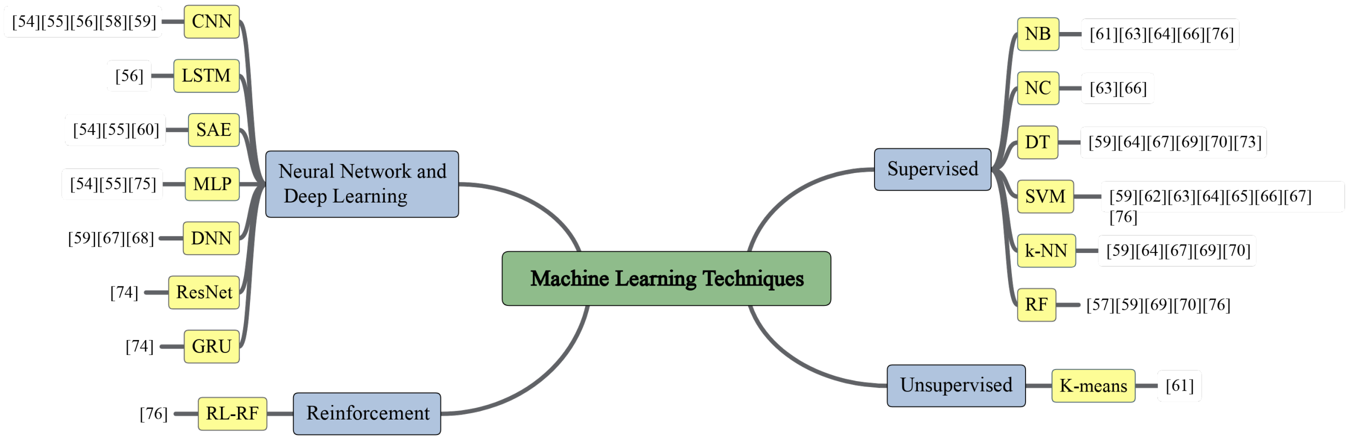

Fig. 8. Use of Machine Learning techniques in the surveyed papers

CNN-Autoencoders to improve the classification accuracy over 98.00%. Moreover, the contribution of the individual ML techniques that compose the proposed classifiers to the overall accuracy of traffic classification has been analyzed in detail. The most commonly used ML techniques among the group of primary studies are Support Vector Machines, Decision Trees, and Convolutional Neural Networks. Despite its low complexity, the SVM model achieved an accuracy with a lower bound of 80.00% and an upper bound of 98.00%. On the other hand, the CNN model has high complexity, and its accuracy has a lower bound of 93.00% and an upper bound over 99.00%. Overall, the best accuracy in traffic classification is 99.65% and was obtained by employing the ML-LSTM classifier proposed by [56].

Traffic classification models depend on several parameters to achieve satisfying accuracy. First, selecting a subset of relevant characteristics during model building is highly relevant. In 76.00% of primary studies, researchers use Feature Selection techniques to improve the classification model's accuracy. As a representative example, the application of a hybrid feature selection method in [64] significantly improved the achieved performance. More precisely, it reduced the time required for generating a classifier by approximately seven times compared to other methods based on conventional feature selection. Similarly, Uwusu et al. [69] employed two feature selection methods—namely, SHAP and SFS—to improve the classifiers' performance. Finally, Fan and Liu [62] performed suitable model tuning and feature selection prior to traffic classification.

One of the directions to be considered in the future is developing traffic classification techniques capable of identifying new traffic flows from a reduced training labeled dataset. For such models, it is important to consider the duration of the captured traffic flow since the best results in classifier accuracy have been observed in 15s and 30s; however, this implies a high computational cost [59], [67]. The Deep Learning-based classifiers proposed by [55], [56], [58], [60], [67], [74] proved to be a viable solution for traffic flow classification in terms of accuracy, efficiency, and scalability. For future research, we recommend exploring Deep Learning-based models, especially those demonstrating good performance and accuracy in other classification tasks. Such techniques will enable the selection of hidden traffic features that can alleviate the problem of working with imbalanced datasets. However, efforts should be made to reduce the DL-based classifiers' training time and improve their learning speed.

The properties and attributes of Internet traffic are continuously changing due to the evolution of technology, the design of new network paradigms, and the deployment of novel applications. This dynamic environment causes traffic classification models in SDN to quickly become inefficient. In this regard, there are still many research problems to be solved. The analyzed research articles and the previous systematic reviews showcase that research on traffic classification models is relatively limited. The main reason for this is the scarcity of labeled datasets. Moreover, the existing datasets have unbalanced classes, which directly affect the accuracy of traffic classification models. This problem is even more intense in SDN, where the limited availability of suitable datasets hinders the rapid development of traffic classifiers. To address this issue, we have compiled some suitable datasets that researchers can use in their works. However, the list is not exhaustive, and there is still a need for more sources with robust extracted features and SDN traffic. Obtaining a real-world labeled dataset in SDN, especially the classification labels of new network applications, requires much effort; thus, future research should focus on exploring unsupervised learning-based traffic classification models.

Application-aware traffic classification aims to recognize specific Internet applications. For example, several works (e.g., [54], [55], [56], [57], [58]) have focused on ranking the most popular mobile apps on Google Play and the most used web apps. Additionally, application-based classification is often applied in network management combined with fine-grained classification. Representative examples include [71], [72], [73], [74], where elephant and mice traffic flow classification is performed for network management in data centers. Traffic classification according to application type relies on identifying applications that maintain the same type of data flows. This way, network resources can be efficiently managed by assigning high QoS to traffic flows requiring special treatment. Following this approach, the authors in [59], [60], [61], [62], [63], [64], [65], [66], [67] classify the applications according to the type of underlying traffic flows and assign different QoS to each application based on the class to which it belongs. Finally, traffic classification based on anomalous traffic was employed to identify Botnets [75] and DDoS attacks [76], categorizing network traffic into a regular class and suspicious traffic. To sum up, organizing traffic into classes enables the better management of network resources, especially regarding QoS and security policies.

The most efficient traffic classification schemes are those implemented in the data plane. This type of implementation reduces the delay and decreases the completion time of a traffic flow. However, most of the analyzed studies focus on traffic classification in the control plane, which implies having more resources on the controller. Future research should strive to implement a classifier in actual hardware so that the classification and scheduling of traffic flows are performed directly on the switch.

VII. Conclusion

This paper presented a systematic review of traffic classification with ML techniques in SDN. The reviewed articles demonstrate the different traffic classification methods according to the employed ML algorithms and techniques for extracting dataset features. The selected works were grouped into four categories based on the type of classification: application-aware, application-type, fine-grained and anomalous. Classification based on application-type improves the enforcement of QoS and security policies in SDN. Most of the papers applied Feature Selection techniques to improve model accuracy and reduce the computational cost. Some issues that directly affect the classifiers' performance and accuracy have been identified: imbalanced training dataset, concept drift, and scalability in the control plane.

In future work, we will explore the construction of a traffic classification model that addresses the identified limitations in SDNs with ML techniques. Specifically, we plan to develop a Deep Learning-based classifier that identifies the hidden features of data to avoid concept drift and improve the classifier's accuracy with unbalanced classes of the training datasets. Another promising direction is compiling an accurate SDN traffic dataset to train the underlying models and create a real-time system to improve QoS and security in SDNs.

Acknowledgment

The authors would like to thank the Distributed Systems, Cybersecurity and Content Research Group (RACKLY) of Universidad de las Fuerzas Armadas ESPE for its scientific and collaborative contribution.

References

[1] CISCO, "Cisco Annual Internet Report (2018–2023)," 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf.

[2] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski, "A knowledge plane for the internet," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM,* 2003, p. 3.

[3] S. Ayoubi et al., "Machine Learning for Cognitive Network Management," *IEEE Communications Magazine*., vol. 56, no. 1, pp. 158–165, Jan. 2018.

[4] D. Kreutz, F. M. V. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proc. IEEE,* vol. 103, no. 1, pp. 14–76, Jan. 2015.

[5] J. Yan and J. Yuan, "A Survey of Traffic Classification in Software Defined Networks," in *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN),* Aug. 2018, pp. 200–206.

[6] R. Boutaba et al.,"A comprehensive survey on machine learning for networking: evolution, applications and research opportunities," *J. Internet Serv. Appl.*, vol. 9, no. 1, p. 16, Dec. 2018.

[7] J. Xie et al., "A Survey of Machine Learning Techniques Applied to Software Defined Networking (SDN): Research Issues and Challenges," *IEEE Commun. Surv. Tutorials,* vol. 21, no. 1, pp. 393–430, 2019.

[8] A. R. Mohammed, S. A. Mohammed, and S. Shirmohammadi, "Machine Learning and Deep Learning Based Traffic Classification and Prediction in Software Defined Networking," in *IEEE International Symposium on Measurements & Networking (M&N),* Jul. 2019, pp. 1–6,

[9] F. Audah, T. S. Chin, R. Kapsin, N. Omar, and A. Tajuddin, "Future Direction of Traffic Classification in SDN from Current Patents Point-of-view," in *2019 15th International Computer Engineering Conference (ICENCO),* Dec. 2019, pp. 121–125.

[10] K. Tamil Selvi and R. Thamilselvan, "Deep learning based traffic classification in software defined networking –a survey," *Int. J. Sci. Technol. Res.,* vol. 9, no. 2, pp. 2034–2041, 2020.

[11] "Open Networking Foundation," 2014. [Online]. Available: https://www.opennetworking.org/

[12] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A Survey on Software-Defined Networking," *IEEE Commun. Surv. Tutorials,* vol. 17, no. 1, pp. 27–51, 2015.

[13] C. Trois, M. D. Del Fabro, L. C. E. de Bona, and M. Martinello, "A Survey on SDN Programming Languages: Toward a Taxonomy," *IEEE Commun. Surv. Tutorials,* vol. 18, no. 4, pp. 2687–2712, 2016.

[14] T. Huang, F. R. Yu, C. Zhang, J. Liu, J. Zhang, and Y. Liu, "A Survey on Large-Scale Software Defined Networking (SDN) Testbeds: Approaches and Challenges," *IEEE Commun. Surv. Tutorials,* vol. 19, no. 2, pp. 891–917, 2017.

[15] A. Blenk, A. Basta, M. Reisslein, and W. Kellerer, "Survey on Network Virtualization Hypervisors for Software Defined Networking," *IEEE Commun. Surv. Tutorials,* vol. 18, no. 1, pp. 655–685, 2016.

[16] "Open vSwitch." [Online]. Available: https://www.openvswitch.org/, Accessed on: Oct. 22, 2020.

[17] F. Hu, Q. Hao, and K. Bao, "A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation," *IEEE Commun. Surv. Tutorials,* vol. 16, no. 4, pp. 2181–2206, 2014.

[18] J. Xie, D. Guo, Z. Hu, T. Qu, and P. Lv, "Control plane of software defined networks: A survey," *Comput. Commun.,* vol. 67, pp. 1–10, Aug. 2015.

[19] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," *IEEE Commun. Surv. Tutorials,* vol. 16, no. 3, pp. 1617–1634, 2014.

[20] B. Park, J. Hong, and Y. Won, "Toward fine-grained traffic classification," *IEEE Commun. Mag.,* vol. 49, no. 7, pp. 104–111, Jul. 2011.

[21] IANA, "Service Name and Transport Protocol Port Number Registry," 2013, [Online]. Available: http://www.iana.org/

[22] S. H. Yeganeh, M. Eftekhar, Y. Ganjali, R. Keralapura, and A. Nucci, "CUTE: Traffic Classification Using TErms," in *2012 21st International Conference on Computer Communications and Networks (ICCCN),* Jul. 2012, pp. 1–9.

[23] Y. Li and J. Li, "MultiClassifier: A combination of DPI and ML for application-layer classification in SDN," in *The 2014 2nd International Conference on Systems and Informatics (ICSAI 2014),* Nov. 2014, pp. 682–686.

[24] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," in *IEEE Communications Surveys & Tutorials,* vol. 10, no. 4, pp. 56-76, Fourth Quarter 2008.

[25] M. Kubat, "An Introduction to Machine Learning". *Cham: Springer International Publishing*, 2015.

[26] M. Mohammed, M. B. Khan, and E. B. M. Bashier, "Machine Learning: Algorithms and Applications," *CRC Press,* 2016.

[27] X. Zhu, "Semi-supervised learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, 2005.

[28] H. Wu and S. Prasad, "Semi-Supervised Deep Learning Using Pseudo Labels for Hyperspectral Image Classification," *IEEE Trans. Image Process.,* vol. 27, no. 3, pp. 1259–1270, Mar. 2018.

[29] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine learning: a review of classification and combining techniques," *Artif. Intell. Rev.,* vol. 26, no. 3, pp. 159–190, Nov. 1967.

[30] P. E. H. T.M. COVER, "Nearest Neighbor Pattern Classification," *IEEE Trans. Inf. THEORY,* vol. I, pp. 1–28, 1967.

[31] L. Breiman, J. Friedman, C. Stone and R. Olshen, "Classification and Regression Trees," *The Wadsworth and Brooks-Cole Statistics-probability Series,* New York: Taylor & Francis; 1984.
[32] A. Kumar, P. Bhatia, A. Goel, and S. Kole, "Implementation and Comparison of Decision Tree Based Algorithms," *Int. J. Innov. Adv. Comput. Sci.,* vol. 4, no. May, pp. 190–196, 2015.
[33] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, pp. 5–32, 2001.
[34] A. Patle and D. S. Chouhan, "SVM kernel functions for classification," in *2013 International Conference on Advances in Technology and Engineering (ICATE),* Jan. 2013, pp. 1–9.
[35] B. Yekkehkhany, A. Safari, S. Homayouni, and M. Hasanlou, "A comparison study of different kernel functions for SVM-based classification of multi-temporal polarimetry SAR data," *ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.,* vol. XL-2/W3, no. 2W3, pp. 281–285, Oct. 2014.
[36] M. G. Nir Friedman, Dan Geiger, "Bayesian Network Classifier," *Mach. Learn.,* vol. 29, pp. 131–163, 1997.
[37] D. Heckerman, "A Tutorial on Learning with Bayesian Networks," in In: Jordan M.I. (eds) Learning in Graphical Models. NATO ASI Series (Series D: Behavioural and Social Sciences), vol 89. Springer, Dordrecht. 1998.
[38] S. Haykin,*Neural Networks: A Comprehensive Foundation.* Prentice Hall PTR, 1994.
[39] K. Lee, D. Booth, and P. Alam, "A comparison of supervised and unsupervised neural networks in predicting bankruptcy of Korean firms," *Expert Syst. Appl.,* 2005.
[40] S. Timotheou, "The random neural network: A survey," *Comput. J.,* vol. 53, no. 3, pp. 251–267, 2010.
[41] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature,* vol. 521, pp. 436–444, 2015.
[42] G. Pandey and A. Dukkipati, "Learning by stretching deep networks," *31st Int. Conf. Mach. Learn. ICML 2014,* vol. 5, pp. 3707–3716, 2014.
[43] J. Schmidhuber, "Deep Learning in neural networks: An overview," *Neural Networks,* vol. 61, pp. 85–117, 2015.
[44] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM,* vol. 60, no. 6, pp. 84–90, 2017.
[45] T. Mikolov, M. Karafiát, L. Burget, C. Jan, and S. Khudanpur, "Recurrent neural network based language model," in *Proceedings of the Eleventh Annual Conference of the International Speech Communication Association,* pp. 1045–1048, 2010.
[46] H. Sak, A. Senior, and F. Beaufays, "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling," *Fifteenth Annu. Conf. Int. Speech Commun. Assoc.,* 2014.
[47] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," *Proceedings of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering,* pp. 3–24, 2007.
[48] T. Kohonen, "The self-organizing map," *Proc. IEEE,* vol. 78, no. 9, pp. 1464–1480, 1990.
[49] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 24, no. 7, pp. 881–892, 2002.
[50] L. Kaelbling, M. Littman, and A. Moore, "Reinforcement Learning: A Survey," *J. Artif. Intell. Res.,* vol. 4, no. 1, pp. 237–285, 1996.
[51] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep Reinforcement Learning: A Brief Survey," *IEEE Signal Process. Mag.,* vol. 34, no. 6, pp. 26–38, Nov. 2017.
[52] G. Tesauro, "Reinforcement Learning in Autonomic Computing: A Manifesto and Case Studies," *IEEE Internet Comput.,* vol. 11, no. 1, pp. 22–30, 2007.
[53] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering - A systematic literature review," *Inf. Softw. Technol.,* vol. 51, no. 1, pp. 7–15, 2009.
[54] L.-H. Chang, Tsung-Han Lee, Hung-Chi Chu, and Cheng-Wei Su, "Application-Based Online Traffic Classification with Deep Learning Models on SDN Networks," *Adv. Technol. Innov.,* vol. 5, no. 4, pp. 216-229, Sep. 2020.
[55] P. Wang, F. Ye, X. Chen, and Y. Qian, "Datanet: Deep learning based encrypted network traffic classification in SDN home gateway," *IEEE Access,* vol. 6, pp. 55380–55391, 2018.
[56] H-K. Lim , J-B. Kim , K. Kim, Y-G. Hong and Y-H. Han , "Payload-based traffic classification using multi-layer LSTM in software defined networks," *Appl. Sci.,* vol. 9, no. 12, 2019.
[57] P. Amaral, J. Dinis, P. Pinto, L. Bernardo, J. Tavares and H. S. Mamede, "Machine Learning in Software Defined Networks: Data collection and traffic classification," *2016 IEEE 24th International Conference on Network Protocols (ICNP),* Singapore, 2016, pp. 1-5.
[58] X. Chen, J. Yu, F. Ye, and P. Wang, "A Hierarchical Approach to Encrypted Data Packet Classification in Smart Home Gateways," in *2018 IEEE 16th Intl Conf on Dependable (DASC/PiCom/DataCom/CyberSciTech),* pp. 41–45, 2018.
[59] S. Abdulazzaq and M. Demirci, "A Deep Learning Based System for Traffic Engineering in Software Defined Networks", *IJISAE,* , vol. 8, no. 4, pp. 206-213, Dec. 2020.
[60] C. Zhang, X. Wang, F. Li, Q. He, and M. Huang, "Deep learning-based network application classification for SDN," *Trans. Emerg. Telecommun. Technol.,* vol. 29, no. 5, 2018.
[61] M. Reza, M. Javad, S. Raouf, and R. Javidan, "Network Traffic Classification using Machine Learning Techniques over Software Defined Networks," *Int. J. Adv. Comput. Sci. Appl.,* vol. 8, no. 7, 2017
[62] Z. Fan and R. Liu, "Investigation of machine learning based network traffic classification," in *2017 International Symposium on Wireless Communication Systems (ISWCS),* 2017, vol. 2017-Augus, pp. 1–6.
[63] M. M. Raikar, M. S M, M. M. Mulla, N. S. Shetti, and M. Karanandi, "Data Traffic Classification in Software Defined Networks (SDN) using supervised-learning," *Procedia Comput. Sci.,* vol. 171, pp. 2750–2759, 2020.
[64] F. A. Md. Zaki and T. S. Chin, "FWFS: Selecting Robust Features Towards Reliable and Stable Traffic Classifier in SDN," in *IEEE Access,* vol. 7, pp. 166011-166020, 2019.
[65] P. Wang, S. Lin and M. Luo, "A Framework for QoS-aware Traffic Classification Using Semi-supervised Machine Learning in SDNs," *2016 IEEE International Conference on Services Computing (SCC),* San Francisco, CA, 2016, pp. 760-765.
[66] M. Amiri, H. Al Osman and S. Shirmohammadi, "Game-Aware and SDN-Assisted Bandwidth Allocation for Data Center Networks," *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR),* Miami, FL, 2018, pp. 86-91.
[67] J. Xu, J. Wang, Q. Qi, H. Sun and B. He, "Deep neural networks for application awareness in SDN-based network," *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP),* Aalborg, 2018, pp. 1-6.
[68] A. Malik, R. de Fréin, M. Al-Zeyadi and J. Andreu-Perez, "Intelligent SDN Traffic Classification Using Deep Learning: Deep-SDN," *2020 2nd International Conference on Computer Communication and the Internet (ICCCI),* pp. 184-189, 2020.
[69] A. I. Owusu and A. Nayak, "An Intelligent Traffic Classification in SDN-IoT: A Machine Learning Approach," *2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom),* pp. 1-6, 2020.
[70] P. K. Mondal, L. P. Aguirre Sanchez, E. Benedetto, Y. Shen, and M. Guo, "A dynamic network traffic classifier using supervised ML for a Docker-based SDN network," *Connection Science,* vol. 33, no. 3, pp. 693–718, 2021.
[71] K. -C. Chiu, C. -C. Liu and L. -D. Chou, "CAPC: Packet-Based Network Service Classifier With Convolutional Autoencoder," *in IEEE Access,* vol. 8, pp. 218081-218094, 2020.
[72] M. Uddin and T. Nadeem, "TrafficVision: A Case for Pushing Software Defined Networks to Wireless Edges," in *2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS),* 2016, pp. 37–46.
[73] P. Xiao, W. Qu, H. Qi, Y. Xu, and Z. Li, "An efficient elephant flow detection with cost-sensitive in SDN," *Proc. 2015 1st Int. Conf. Ind. Networks Intell. Syst. INISCom 2015,* pp. 24–28, 2015.
[74] W.-X. Liu, J. Cai, Y. Wang, Q. C. Chen, and J.-Q. Zeng, "Fine-grained flow classification using deep learning for software defined data center networks," *J. Netw. Comput. Appl.,* vol. 168, 2020.
[75] I. Letteri, G. Della Penna, and G. De Gasperis, "Botnet Detection in Software Defined Networks by Deep Learning Techniques," in *International Symposium on Cyberspace Safety and Security,* Springer, Cham, 2018, pp. 49-62.
[76] C. Xu, H. Lin, Y. Wu, X. Guo and W. Lin, "An SDNFV-Based DDoS Defense Technology for Smart Cities," *IEEE Access,* vol. 7, pp. 137856–137874, 2019.
[77] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of Encrypted and VPN Traffic using Time-related Features," in *Proceedings of the 2nd International Conference on Information Systems Security and Privacy,* 2016, pp. 407–414.
[78] V. Carela-Español, T. Bujlow and P. Barlet-Ros, "Is our ground-truth for traffic classification reliable?" in *Passive and Active Measurement, of Lecture Notes in Computer Science,* vol. 8362, pp. 98-108, 2014.
[79] A. Habibi Lashkari, G. Draper Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of Tor Traffic using Time based Features," in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy,* 2017, vol. 1, pp. 253–262.
[80] A. Moore, D. Zuev, and M. Crogan, "Discriminators for use in flow-based classification," *Intel Research Tech. Rep.* Cambridge, 2005.

[81] "MAWI Working Group traffic archive." [Online]. Available: http://mawi.wide.ad.jp/mawi/
[82] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement,* 2010, pp. 267–280,
[83] P.A.A. Resende, A.C. Drummond, *The hogzilla dataset,* 2018. [Online]. Available: https://ids-hogzilla.org/dataset/
[84] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Comput. Secur.,* vol. 31, no. 3, pp. 357–374, 2012.
[85] G. Kakkavas, A. Stamou, V. Karyotis, and P. Symeon, "Network Tomography for Efficient Monitoring in SDN-Enabled 5G Networks and Beyond: Challenges and Opportunities," *IEEE Communications Magazine,* vol. 59, no. 3, pp. 70-76, 2021.