Wasit Journal for Pure Science

Journal Homepage: https://wjps.uowasit.edu.iq/index.php/wjps/index
e-ISSN: 2790-5241 p-ISSN: 2790-5233

Traffic data classification in SDN network based on machine learning algorithms

Samah Adil Mohsin[1]*, Ali Saeed Alfoudi[1,2]

[1]College of Computer Science and Information Technology, University of Al-Qadisiyah, Al-Qadisiyah, IRAQ

[2]College of Computer Science, Liverpool John Moores University, Liverpool, UK

*Corresponding Author: Samah Adil Mohsin

DOI: https://doi.org/10.31185/wjps.375
Received 10 April 2024; Accepted 01 Jun 2024; Available online 30 Jun 2024

ABSTRACT: Traffic classification plays a crucial role in various domains of network management, including architectural structure,service measurement, advertising, and security monitoring. Software-defined networks (SDN) is a new technology that has the potential to solve typical network problems through the process of streamlining network administration, the introduction of network programmability, and the provision of a global perspective of a network. In recent years, Software-Defined Networking (SDN) has introduced novel opportunities for traffic classification. Various techniques for traffic classification within SDN environments have been examined, proposed, and developed. This survey delves into traffic classification under SDN, which is a vital component for improving network services, administration, and security. We give an in-depth assessment of traffic categorization algorithms adapted for SDN, emphasizing the fresh opportunities and problems they present. We cover the many metrics for assessing the effectiveness of these traffic classification algorithms, such as accuracy, precision, recall, and F1 score, and we examine the numerous datasets that serve as performance benchmarks. The study also synthesizes the findings of existing research, revealing trends and the efficacy of various techniques in the context of SDN-enabled settings. This document serves as a resource for scholars and practitioners seeking to optimize traffic classification strategies by providing a complete review and assessment of existing traffic classification approaches.

Keywords: Software-defined network – Network traffic classification - machine learning- Quality of Service (QoS)- network functions virtualization (NFV)

1. INTRODUCTION

Modern networks are growing increasingly complicated to meet the diverse demands of customers. Automation is required to handle these complicated networks. Anomaly detection, traffic engineering, and intrusion protection can all benefit from network traffic classification in a software network[1]. As a result, modern networks are becoming more software through the use of technologies such as Network Function Virtualization (NFV) and Software-Defined Networking (SDN) [2]. The data plane and control plane are separated in SDN. The centralized controller decides on data forwarding[3]. The SDN controller functions as a network operating system and has centralized control over the whole network. The controller allows for intelligent deployment[4].]. For various traffic classification methods, a number of techniques, such as payload-based and port-based algorithms, have been proposed throughout the years to achieve accurate traffic categorization. The port-based technique is the simplest and fastest, but because modern programs dynamically assign port numbers, it is no longer used. The payload-based solution is accurate, but it affects overall network performance and causes Quality of Service (QoS) issue[5].

Machine learning is an important method that can be used in SDN architecture to improve network functionality as well as non-functional characteristics such as performance, security, and others. Ml is a concept made up of three main components: the model, the parameters, and the learning system[6]. ML approaches can be used independently or in combination with other northbound applications of the SDN controller to enhance the intelligence of SDN. SDN controllers utilize machine learning to do network data analysis, optimization, and automation.[7]

Information technology is advancing rapidly, and the management system of the 5G/6G network is evolving towards integration, distribution, diversification, and intelligence [8]. The architecture of 5G is based on services, where network functions interact using well-defined interfaces. For example, the N2 interface facilitates communication between the radio access network (RAN) and the access and mobility function [9]. This design separates user data from the control plane through distinct access and forwarding layers. The control layer is responsible for the logical centralization of network control, while the access layer encompasses various wireless technologies and network topologies. The forwarding layer aims to provide high reliability, low latency, and a manageable load for service data flow by utilizing a common hardware platform. With the advent of 5G and the upcoming 6G technology, wireless networks have become increasingly complex and virtualized. Although traditional wireless network topologies have significant security flaws, 5G and 6G promote infrastructure sharing [10].

Network slicing is a promising technology that aims to handle specific traffic while meeting the Quality of Service (QoS) criteria for particular application data flows in a 5G network [11]. Key factors enabling network slicing include NFV, SDN, edge computing, and cloud computing [12 NFV allows for the use of generic hardware for cost-effective network function implementations, while SDN enables the separation of the control plane and data plane for flexible resource management and efficiency [13]. Therefore, the integration of NFV and SDN in network slicing has become an essential technology for 5G, 6G, and next network designs. [14]. Various models can be employed to distinguish network slices, including vertical and horizontal, RAN and core, as well as static and dynamic. The network slicing architecture consists of three layers:

- **Infrastructure layer**: Manages virtual and physical resources, providing connectivity and computational capabilities.
- **Network slice instance layer**: Operates at the highest level of the infrastructure stack and consists of network slice instances that collectively form end-to-end logical network slices.
- **Service instance layer**: Positioned above all other levels, encompassing business services and end-users. These services are provided by either the network operator or a third party through service instances.

A functional network slice comprises two primary subslices: The core subslice, which is related to the core network, and the RAN subslice, which is specifically designed for the next-generation RAN. [15].

2. SOFTWARE-DEFINED NETWORK (SDN)

SDN is a new technology that focuses on adaptability and flexibility. SDN applications use load balancing, routing, and access control to effectively manage and control networks, ensuring optimal performance. The most significant advantage of SDN is its ability to provide centralized control and improved network management [16]. NFV and SDN technologies are changing network architecture by converting complex physical entities into virtual and programmable nodes, consolidating network control to enhance the overall network structure. SDN achieves this by separating the control plane and data plane and utilizing a controller program for centralized network management. The main objective of SDN is to transform the complex and tangible network infrastructure into a virtualized, programmable, and openly accessible network architecture[17].

SDN architectures enable the use of APIs to offer various network services, such as routing, bandwidth allocation, access control, multicast, security measures, traffic optimization, QoS, processor and storage efficiency, energy consumption, and policy administration. These services can be tailored to align with business objectives, allowing for the management of the entire network through intelligent orchestration and provisioning systems [18]. The fundamental concept of SDN involves dividing data and control traffic into three layers: application, control, and data plane. This concept is depicted in Fig 1 [19].

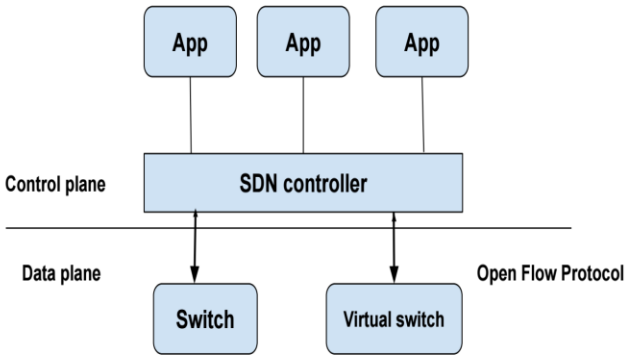

FIGURE 1. Software-defined networking (SDN) architecture [20]

Failure of communication links is common in any network. The protection-based recovery technique in SDN reduces the delay in recovering from failures by installing alternate routes at data plane switches [21]. The emergence of SDN technology facilitates the centralized management and operation of networks, enabling flexible configuration of networking resources, such as switches, using programmable interfaces [22]. SDN offers new approaches to designing and managing different types of networks [23].

OpenFlow is a widely recognized southbound API protocol in SDN that enables communication between controllers and network switches [24]. It operates within both software and hardware on the data plane. OpenFlow switches can have one or more flow tables that store flow entries. Each flow entry includes match fields and actions, which are populated by the controller [20]. The control mechanism of SDN has the potential to address the architectural and implementation complexity associated with edge computing. By introducing a new networking mechanism, SDN enables efficient resource management in parallel [25].

3. TRAFFIC CLASSIFICATION

Traffic classification is an intelligent task that involves categorizing traffic into different classes. This is used for network management, service measurement, network monitoring, and other applications. Additionally, traffic classification allows for effective resource distribution, ensures QoS, and enables the configuration of access constraints and other network security parameters [26]. The emergence of SDN has introduced new opportunities for traffic classification and feature selection. The comprehensive perspective of SDN controllers enables the extraction of network traffic information from switches [20]. With this centralized view of the entire network and the ability to classify traffic at the controller, application-specific rules critical for efficient and seamless network operation can be formulated [27]. The effectiveness and efficiency of the TC engine are crucial in SDN, as it regulates network traffic based on flows, as shown in Figure 2 [28].

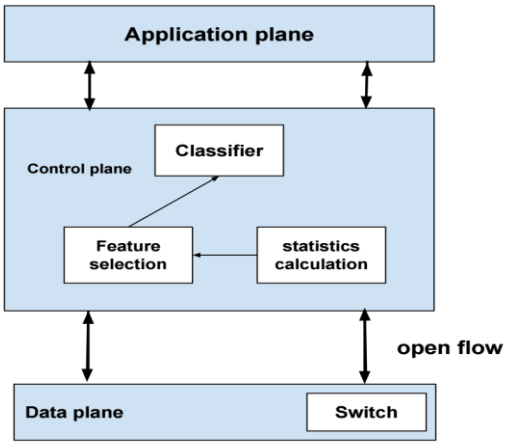

FIGURE 2. General framework of traffic classification in software-defined networking (SDN) [20]

3.1 Traffic classification without ML

Traffic classification algorithms utilize flows and their associated descriptive features as inputs. The effectiveness of these classification algorithms relies not only on the quality of the features gathered but also on their quantity. In order to achieve more accurate traffic classification, it is crucial to collect detailed information about specific features of each traffic flow, such as the average packet transmission time[29]. Historically, various methods for traffic classification have been explored and implemented within communication systems for many years.

- **IP port**: In the past, this was one of the most commonly used methods. However, it was only partially effective because several applications used fixed port numbers allocated by the Internet Assigned Numbers Authority (IANA).

- **DPI**: To overcome the limitations of IP port classification methods, payload-based classification methods were developed. Deep Packet Inspection (DPI) is a term often used interchangeably with payload-based methods [30].

These traditional methods have faced various challenges, leading recent research to emphasize machine learning (ML) techniques that utilize statistical properties for traffic classification [31].

3.2 Traffic classification with ML

ML has the ability to mine data and extract implicit, regular, and valuable information from large datasets [32]. The use of ML techniques has been implemented in network systems as a highly effective strategy for continuously monitoring the dynamic behaviour of networks, automatically analyzing network data, and making predictions regarding network usage [33]. Recently, ML-based TC algorithms have become popular for overcoming limitations imposed by traditional classification methods [30]. The application of ML to classify traffic has been a prominent area of research in network measurement. This is to accommodate the characteristics of vast amounts of Internet traffic data and the large dynamic changes in application attributes. When ML is used for traffic classification problems, the object of study usually involves a packet sequence with identical values for the five tuples (source port, destination port, source IP, destination IP, and transport layer protocol). Using ML techniques for traffic classification involves two primary components:

- Selecting the appropriate network flow attribute set to construct an attribute vector.
- Selecting the most suitable ML algorithm for building a classification model [34].

A typical ML technique consists of two primary stages: the training phase and the decision-making phase. During the training phase, ML techniques are applied to the training dataset to establish the system model. Using this trained model, the system can estimate the output of each new input during the decision-making phase [35]. ML algorithms can be divided into three groups: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning involves using labelled data for classification and regression purposes, while unsupervised learning focuses on classifying unlabelled data into distinct categories. Reinforcement learning involves the agent interacting with the environment to acquire knowledge about actions that yield the highest reward [36].

ML techniques are becoming increasingly common in research papers for classifying network traffic without requiring access to the content or port numbers of each packet. To find a solution, statistical characteristics that demonstrate the behaviour of specific protocols or application flows are collected [37]. ML enables the analysis of large volumes of data in the network, which can then be further analyzed to adjust any specific aspect efficiently and cost-effectively as required [38].

4. DATASET

Datasets are essential in all ML methods, as they form the basis for the process of learning classifiers and allow for the comparison of different techniques. In the field of TC, researchers often depend on different open data sets found in the literature. These data sets include various types of traffic data, such as raw traffic, flows, and characteristics. Here, we provide concise descriptions of some of the most commonly used data sets in this domain [39]. The influence of the data set on prediction accuracy has also been acknowledged [40]. During the process of selecting the dataset, several factors were taken into consideration, such as the diversity of network traffic, the availability of the PCAP attribute-relation file format (ARFF), the extensive use of the dataset in research papers, and its accessibility. The number 37 is enclosed in square brackets [37].

- **Moore** [37]: This dataset consists of network traces collected at Cambridge University laboratories in 2005. It contains data from different applications such as BitTorrent, MySQL, FTP, and HTTP, categorized into ten service areas, including multimedia games and P2P. The popularity of this dataset is due to its availability in ARFF format, which includes 249 well-defined statistical variables.
- **USTC-TFC2016**: This collection includes PCAP traffic for 10 different programs, such as Zeus, FaceTime, and Skype, which include both malicious and harmless applications [41].
- **CTU-13 dataset** [42]: Compiled from the 2011 CTU University botnet network activity, this dataset includes traffic from different infections and programs, as well as malicious traffic from botnets. It encompasses multiple C&C channel protocols linked to the virus in question.
- **KDD'99**: This dataset is an improved version of the DARPA98 dataset that was created by analyzing the tcpdump component. It includes various types of attacks, such as Pod-DoS, Smurf-DoS, buffer overflow, and Neptune-DoS, which were reported by the University of California in 2007. This dataset combines both benign and malicious network traffic in a simulated environment.
- **DARPA**: The dataset was generated specifically for the purpose of conducting network security analysis. and revealed vulnerabilities related to the artificial injection of malicious and benign traffic. Email, surfing, FTP, Telnet, IRC, and SNMP activity are all included in this dataset .Additionally, the dataset covers a range of threats including DoS, guess password, buffer overflow, remote FTP, syn flood, Nmap, and rootkit attacks.
- **ISCX2012**: This dataset comprises two distinct profiles – the Alpha profile and the Beta profile. The Alpha profile is characterized by executing multiple multistage assault scenarios, while the Beta profile serves as a benign traffic generator designed to produce realistic network traffic alongside ambient noise. This dataset consists of HTTP network traffic, with supported protocols including IMAP, POP3, SMTP, SSH, and FTP, covering the entirety of data sent within a packet [43].
- **NSL-KDD**: This dataset contains features and labels that indicate whether a feature represents normal behavior or an attack. Attacks can take different forms, and each instance in the training set corresponds to a connection session. These sessions are divided into four categories: basic network connection features, content-related features, host-based traffic, and time-related features. The dataset covers two types of scenarios: normal and attack, with attacks classified into four groups: DoS, remote to local, user to Root, and probing.
- **CICIDS-2017:** The Canadian Institute for Cybersecurity at the University developed this dataset in 2017. Its objective was to develop intrusion detection systems by incorporating various attack scenarios. The attack profiles in this dataset were generated using easily accessible tools and scripts, encompassing six attack profiles: brute force, heartbleed, botnet, DoS, DDoS, web assault, and penetration assaults [44].
- **CSE-CIC-IDS2018** [45]: This dataset was developed by the Canadian Institute for Cybersecurity lab and is one of the most recent datasets that fulfils all research requirements. It encompasses comprehensive traffic, various attack types, and accurate labelling. There are seven distinct attack types represented: brute force, heartbleed, botnet, intrusion, online attacks, DoS, DDoS, and network infiltration. The victim organization consisted of thirty servers, 420 computers, and five departments. The infrastructure used 50 machines to simulate the attacks. This dataset comprises 80 attributes extracted from recorded traffic using CICFlowMeter-V3. Detailed documentation of network traffic and system logs for each computer is also included.
- **Kyoto** [46]: This dataset was created by utilizing honeypots, thus eliminating the need for anonymization and manual labelling. However, it offers a limited depiction of network traffic, as it solely focuses on attacks directed at the honeypots. This dataset includes eleven additional attributes, such as malware detection, IDS Detection, and Ashula detection, which enhance its suitability for research and evaluation of NIDS. The simulated attacks imitate regular traffic patterns and generate only mail traffic data and DNS, which do not correspond to real-world traffic. Consequently, this results in no false positives. False positives are significant as they reduce the number of notifications.
- **MAWI** [37]: This dataset, operating under a working group as a part of Japan's WIDE project, has been collecting genuine network traffic from different sample sites across the extensive network since 2000. This dataset includes both malicious and harmless network activity from various applications.

5. EVALUATION

Three commonly utilized metrics to evaluate the effectiveness of classification methods are based on the outcomes classified as true positives (*TP*), true negatives (*TN*), false negatives (*FN*), and false positives (*FP*) [47]. Below, we define these performance metrics:

- **Accuracy (*AC*)**: *AC* is defined as the ratio of accurately predicted Positive Instances of Interest (PIMs), which include TP and TN, to the total observed PIMs, which encompass FP, FN, TP, and TN. It is calculated using the formula in Eq. 1:

$$AC = \frac{TP + TN}{TP + TN + FP + FN} . \quad (1)$$

- **Precision (*P*)**: *P* is defined as the ratio of accurately predicted positive PIMs to the total projected positive PIMs. It is calculated using the formula in Eq. 2:

$$P = \frac{TP}{TP + FP} \quad (2)$$

- **Recall (*RC*)**: *RC* is the ratio of accurately predicted positive PIMs to all PIMs in the actual class. It is calculated using the formula in Eq. 3:

$$RC = \frac{TP}{TP + FN} \quad (3)$$

- **F1 score (*F*1)**: The *F*1 score is the weighted average of *P* and *RC*. It is calculated using the formula in Eq. 4 [48]:

$$F1 = 2 \times \frac{P + RC}{P + RC} \quad (4)$$

The confusion matrix is used to calculate the metrics. It includes the following values: *TP*, which are correctly predicted positive values, and *TN*, which are correctly predicted negative values. *FP* represent cases where the predicted class result is positive, but the actual class result is negative. *FN* occur when the actual class is categorized as positive, but the predicted class is negative. Below are the individual formulas for analyzing the algorithm's performance [49]. Table 1 illustrates the network traffic classification evaluation

Table 1. Network Traffic Classification Evaluation

Reference	Proposed solution	Dataset	Result evaluation			
			Accuracy	Precision	Recall	F1_score
[50]	kNN, K-means, and EM	Moore	95%	-	-	-
[51]	K-means	MAWI	88%	-	-	-
[52]	GMM	UNIBS-2009	99%	-	-	-
[53]	CNN RF	ISCX 2012	99%	-	-	-
[54]	Autoencoder+SVM.	KDD99	95%	-	-	-
[55]	CNN	USTC-TFC2016	99.41%	99.6%	99.5%	99.5%
[56]	Entropy-based method	CTU-13	99.12%	-	-	-
[57]	Pearson correlation coefficient (PCC) Mutual information (MI)	KDD99	99.98%	-	-	-
[58]	ANN, SVM	NSL-KDD	94.02%	-	-	-

[59]	Random Forest: Bayes Net: Random Tree: Naive Bayes: J48:	CICIDS-2017	99.86% 99.75% 99.72% 99.68% 99.87%	-	-	-
[60]	CNNs, RNNs, DBNs	CSE-CIC-IDS2018	99%	-	-	-
[61]	2D-CNN	ISCX-VPN-NonVPN-2016	98.9%	98.7%	99.2%	98.9%
		Regular encrypted traffic identification:	97.8%	97.6%	98.1%	97.8%
		Malicious traffic identification:				

5.1- Comparison of Previous Studies Using Machine Learning:

This section compares previous research studies for network traffic classification that employ machine learning algorithms:

In this study, The K-means algorithm employs Bernaille et al. [62] to classify network flows by organizing the dataset into a predefined number of clusters (k clusters). This approach achieved an accuracy of up to 80%. Each input is represented as a coordinate based on feature values, consisting of groups of points. There are P dimensions in a space that shows network flows. Each dimension represents a different part, like the size of a packet. The size of packet p in the flow is shown by the position on dimension p.

The method continuously updates the clusters until they reach a state of stability, where the centroids stay unaltered.

In contrast, Roughan et al. [63]utilized the K-nearest neighbor (KNN) method for classifying network traffic. The K-Nearest Neighbors (KNN) algorithm, when using three nearest neighbors, obtained a maximum accuracy of 98%. The study showcased the superior accuracy and memory efficiency of KNN, especially when employing three closest neighbors. This may be attributed to the method's robustness and low computational cost. Nevertheless, it was observed that the memory and processing time may escalate due to an increase in the number of neighbors. However, K-means was superior in terms of processing time, making it an appropriate solution for real-time classification scenarios.

Liu et al. [64]present an enhanced SVM algorithm as the Optimized Facile Support Vector Machine (OFSVM) for the purpose of network traffic categorization. Their assessment demonstrates that the Network Traffic Malware Identification (NTMI) methodology, employing OFSVM, attains greater precision and a reduced rate of false positives in comparison to alternative methodologies. The NTMI technique demonstrates an average accuracy of 92.5% and a false positive rate of 5.527%.

In contrast, Pradhan et al.[65] study proposes machine learning algorithms, specifically Artificial Neural Network (ANN) and Support Vector Machine (SVM), for network traffic classification. The classification simulation model was designed using WEKA, incorporating Multilayer Perceptron (MLP) for ANN and Sequential Minimal Optimization (SMO) for SVM. However, the methodology of Liu et al.[64], which integrates enhanced feature selection, dimensionality reduction, and parameter optimization, achieves superior classification accuracy, and reduces false positive rates. Therefore, the OFSVM approach is particularly advantageous for applications requiring high precision.

6- CHALLENGES IN SOFTWARE-DEFINED NETWORKING TRAFFIC CLASSIFICATION

Traffic classification plays a crucial role in analyzing traffic and efficiently allocating network resources for various services. Accurate traffic classification is necessary for categorizing network traffic into predefined classes of interest [26]. While SDN greatly enhances network management and enables the control of diverse traffic flows, implementing this process efficiently requires an SDN controller with a real-time classification method that is scalable, reliable, and adaptable to future network expansion. ML-based traffic classification methods offer a promising alternative to traditional

approaches in SDNs [66]. However, several challenges need to be addressed, including computational complexity, classifier accuracy, imbalanced training datasets, and concept drift. Concept drift refers to the changing relationships between input and output data over time, particularly in network traffic flows, which can render a traffic classification model outdated. In this section, we discuss the key challenges associated with traffic classification in SDNs:

- **Availability of Datasets**: Superior datasets are essential for advancing and evaluating ML algorithms. However, the lack of openly accessible and universally standardized datasets is a significant challenge, not only for SDN but also for numerous other fields [67].
- **High Bandwidth Traffic Classification**: The rapid development of network technology poses a significant obstacle for TC systems, as certain situations might require processing traffic at gigabit rates [20].
- **Interpretability and Clarity**: Interpretability and clarity are crucial in traffic classification within SDNs, especially in cases where transparency is vital, such as network security [68].
- **Resource Management**: Another challenge is resource management, as sustaining network efficiency and dependability requires effectively distributing resources in accordance with changing demand patterns [69].
- **Real-World Networks**: Real-world networks exhibit significantly more complexity. Achieving great precision in a controlled setting is inadequate for practical deployment. The performance of networks in real-world scenarios is influenced by numerous factors, including scalability, availability, and the ability to adapt to dynamic conditions [68].
- **Ideal Network Assumption**: ML algorithms trained under idealized network assumptions may not perform well when dealing with the complexity and variances of real-world network settings. This jeopardizes the traffic classification system's dependability and may decrease classification accuracy [67].

7-CONCLUSIONS

Research on traffic classification has been a significant and long-lasting field of study. With the rapid advancement of SDN, an increasing number of researchers are focusing on traffic classification within SDN. Based on the provided network traffic classification evaluation (Table 1), ML algorithms have proven to be highly effective for network traffic classification. The top-performing algorithms achieved accuracy rates exceeding 99% on most datasets. Random Forest and convolutional neural networks emerged as the overall best-performing algorithms, demonstrating the highest accuracy across various datasets, including those for encrypted traffic and intrusion detection. Additionally, algorithms such as Gaussian mixture models, entropy-based methods, SVMs, and ANNs also showed commendable performance. However, the selection of the best algorithm for a specific application will depend on the particular dataset and the desired accuracy and performance characteristics.

References

[1] M. S. Towhid and N. Shahriar, “Encrypted Network Traffic Classification using Self-supervised Learning,” *Proc. 2022 IEEE Int. Conf. Netw. Softwarization Netw. Softwarization Coming Age New Challenges Oppor. NetSoft 2022*, pp. 366–374, 2022, doi: 10.1109/NetSoft54395.2022.9844044.

[2] S. W. Yoon and S. J. Jeong, “Implementing coordinative contracts between manufacturer and retailer in a reverse supply chain,” *Sustain.*, vol. 8, no. 9, 2016, doi: 10.3390/su8090913.

[3] K. Shingare, R. Nandurkar, P. Shrivastav, and S. Bendale, “Intrusion Dataset Over Network Traffic of SDN and TCP/IP Network,” *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 6, no. 1, pp. 694–701, 2021, doi: 10.48175/ijarsct-1459.

[4] Y. R. Chen, A. Rezapour, W. G. Tzeng, and S. C. Tsai, “RL-Routing: An SDN Routing Algorithm Based on Deep Reinforcement Learning,” *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 3185–3199, 2020, doi: 10.1109/TNSE.2020.3017751.

[5] W. J. Eom, Y. J. Song, C. H. Park, J. K. Kim, G. H. Kim, and Y. Z. Cho, “Network Traffic Classification Using Ensemble Learning in Software-Defined Networks,” *3rd Int. Conf. Artif. Intell. Inf. Commun. ICAIIC 2021*, pp. 89–92, 2021, doi: 10.1109/ICAIIC51459.2021.9415187.

[6] S. Faezi and A. Shirmarz, “A Comprehensive Survey on Machine Learning using in Software Defined Networks (SDN),” *Human-Centric Intell. Syst.*, vol. 3, no. 3, pp. 312–343, 2023, doi: 10.1007/s44230-023-00025-3.

[7] Imran, Z. Ghaffar, A. Alshahrani, M. Fayaz, A. M. Alghamdi, and J. Gwak, “A topical review on machine learning, software defined networking, internet of things applications: Research limitations and challenges,” *Electron.*, vol. 10, no. 8, 2021, doi: 10.3390/electronics10080880.

[8] Q. Long, Y. Chen, H. Zhang, and X. Lei, “Software Defined 5G and 6G Networks: a Survey,” *Mob. Networks Appl.*, vol. 27, no. 5, pp. 1792–1812, 2022, doi: 10.1007/s11036-019-01397-2.

[9] N. M. Akshatha, P. Jha, and A. Karandikar, “A Centralized SDN Architecture for the 5G Cellular Network,” *IEEE 5G World Forum, 5GWF 2018 - Conf. Proc.*, pp. 147–152, 2018, doi: 10.1109/5GWF.2018.8516960.
[10] S. Singh, V. Mehla, and S. Nikolovski, “LSSDNF: A Lightweight Secure Software Defined Network Framework for Future Internet in 5G–6G,” *Futur. Internet*, vol. 14, no. 12, 2022, doi: 10.3390/fi14120369.
[11] Z. Shu and T. Taleb, “A Novel QoS Framework for Network Slicing in 5G and beyond Networks Based on SDN and NFV,” *IEEE Netw.*, vol. 34, no. 3, pp. 256–263, 2020, doi: 10.1109/MNET.001.1900423.
[12] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, “Network Slicing in 5G: Survey and Challenges,” *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 94–100, 2017, doi: 10.1109/MCOM.2017.1600951.
[13] C. Bektas, S. Monhof, F. Kurtz, and C. Wietfeld, “Towards 5G: An Empirical Evaluation of Software-Defined End-to-End Network Slicing,” *2018 IEEE Globecom Work. GC Wkshps 2018 - Proc.*, no. 1, pp. 1–6, 2018, doi: 10.1109/GLOCOMW.2018.8644145.
[14] M. Condoluci and T. Mahmoodi, “Softwarization and virtualization in 5G mobile networks: Benefits, trends and challenges,” *Comput. Networks*, vol. 146, pp. 65–84, 2018, doi: 10.1016/j.comnet.2018.09.005.
[15] S. Wijethilaka and M. Liyanage, “Survey on Network Slicing for Internet of Things Realization in 5G Networks,” *IEEE Commun. Surv. Tutorials*, vol. 23, no. 2, pp. 957–994, 2021, doi: 10.1109/COMST.2021.3067807.
[16] M. H. H. Khairi *et al.*, “Detection and Classification of Conflict Flows in SDN Using Machine Learning Algorithms,” *IEEE Access*, vol. 9, pp. 76024–76037, 2021, doi: 10.1109/ACCESS.2021.3081629.
[17] C. C. Liu, Y. Chang, C. W. Tseng, Y. T. Yang, M. S. Lai, and L. Der Chou, “SVM-based Classification Mechanism and Its Application in SDN Networks,” *2018 10th Int. Conf. Commun. Softw. Networks, ICCSN 2018*, pp. 45–49, 2018, doi: 10.1109/ICCSN.2018.8488219.
[18] G. Tank, A. Dixit, A. Vellanki, and D. Annapurna, “Software Defined Networks : The New Norm for Networks,” no. 2013, pp. 2013–2015, 2017.
[19] A. Shirmarz and A. Ghaffari, “Automatic Software Defined Network (SDN) Performance Management Using TOPSIS Decision-Making Algorithm,” *J. Grid Comput.*, vol. 19, no. 2, 2021, doi: 10.1007/s10723-021-09557-z.
[20] J. Yan and J. Yuan, “A Survey of Traffic Classification in Software Defined Networks,” *Proc. 2018 1st IEEE Int. Conf. Hot Information-Centric Networking, HotICN 2018*, no. HotICN, pp. 200–206, 2019, doi: 10.1109/HOTICN.2018.8606038.
[21] M. A. Moyeen, F. Tang, D. Saha, and I. Haque, “SD-FAST: A Packet Rerouting Architecture in SDN,” *15th Int. Conf. Netw. Serv. Manag. CNSM 2019*, 2019, doi: 10.23919/CNSM46954.2019.9012703.
[22] G. Kim, Y. Kim, and H. Lim, “Deep Reinforcement Learning-Based Routing on Software-Defined Networks,” *IEEE Access*, vol. 10, pp. 18121–18133, 2022, doi: 10.1109/ACCESS.2022.3151081.
[23] K. Mershad, “SURFER: A Secure SDN-Based Routing Protocol for Internet of Vehicles,” *IEEE Internet Things J.*, vol. 8, no. 9, pp. 7407–7422, 2021, doi: 10.1109/JIOT.2020.3038465.
[24] 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, doi: 10.1109/COMST.2014.2326417.
[25] W. Rafique, L. Qi, I. Yaqoob, M. Imran, R. U. Rasool, and W. Dou, “Complementing IoT Services through Software Defined Networking and Edge Computing: A Comprehensive Survey,” *IEEE Commun. Surv. Tutorials*, vol. 22, no. 3, pp. 1761–1804, 2020, doi: 10.1109/COMST.2020.2997475.
[26] D. Nu˜nez-Agurto, W. Fuertes, L. Marrone, and M. Macas, “Machine Learning-Based Traffic Classification in Software-Defined Networking: A Systematic Literature Review, Challenges, and Future Research Directions,” *IAENG Int. J. Comput. Sci.*, vol. 49, no. 4, 2022.
[27] B. Pradhan, G. Srivastava, D. S. Roy, K. H. K. Reddy, and J. C. W. Lin, “Traffic Classification in Underwater Networks Using SDN and Data-Driven Hybrid Metaheuristics,” *ACM Trans. Sens. Networks*, vol. 18, no. 3, 2022, doi: 10.1145/3474556.
[28] P. Wang, S. C. Lin, and M. Luo, “A framework for QoS-aware traffic classification using semi-supervised machine learning in SDNs,” *Proc. - 2016 IEEE Int. Conf. Serv. Comput. SCC 2016*, pp. 760–765, 2016, doi: 10.1109/SCC.2016.133.
[29] A. S. Da Silva, C. C. Machado, R. V. Bisol, L. Z. Granville, and A. Schaeffer-Filho, “Identification and selection of flow features for accurate traffic classification in SDN,” *Proc. - 2015 IEEE 14th Int. Symp. Netw. Comput. Appl. NCA 2015*, pp. 134–141, 2016, doi: 10.1109/NCA.2015.12.
[30] A. A. El-serwy, E. AbdElhalim, and M. A. Mohamed, “Network Slicing Based on Real-Time Traffic Classification in Software Defined Network (SDN) using Machine Learning,” *MEJ. Mansoura Eng. J.*, vol. 47, no. 3, pp. 1–10, 2022, doi: 10.21608/bfemu.2022.261455.
[31] 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, doi: 10.14569/ijacsa.2017.080729.
[32] F. Xie, D. Wei, and Z. Wang, “Traffic analysis for 5G network slice based on machine learning,” *Eurasip J.*

Wirel. Commun. Netw., vol. 2021, no. 1, 2021, doi: 10.1186/s13638-021-01991-7.
[33] J. Kwon, D. Jung, and H. Park, "Traffic Data Classification using Machine Learning Algorithms in SDN Networks," *Int. Conf. ICT Converg.*, vol. 2020-Octob, pp. 1031–1033, 2020, doi: 10.1109/ICTC49870.2020.9289174.
[34] R. Edvhg, T. T. Frp, L. Khqj, I. X. H. G. X. Fq, and O. D. Hu, "08756496," pp. 5–9.
[35] 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, doi: 10.1109/COMST.2018.2866942.
[36] M. M. Raikar, S. M. Meena, 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, no. 2019, pp. 2750–2759, 2020, doi: 10.1016/j.procs.2020.04.299.
[37] A. Azab, M. Khasawneh, S. Alrabaee, K. K. R. Choo, and M. Sarsour, "Network traffic classification: Techniques, datasets, and challenges," *Digit. Commun. Networks*, no. June 2022, 2023, doi: 10.1016/j.dcan.2022.09.009.
[38] A. Thantharate, R. Paropkari, V. Walunj, and C. Beard, "DeepSlice: A Deep Learning Approach towards an Efficient and Reliable Network Slicing in 5G Networks," *2019 IEEE 10th Annu. Ubiquitous Comput. Electron. Mob. Commun. Conf. UEMCON 2019*, pp. 0762–0767, 2019, doi: 10.1109/UEMCON47517.2019.8993066.
[39] J. Krupski, W. Graniszewski, and M. Iwanowski, "Data transformation schemes for cnn-based network traffic analysis: A survey," *Electron.*, vol. 10, no. 16, 2021, doi: 10.3390/electronics10162042.
[40] A. W. Moore and D. Zuev, "Internet Class Moore and Zuev," *Sigmetrics*, pp. 50–60, 2005.
[41] S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Comput. Secur.*, vol. 45, pp. 100–123, 2014, doi: 10.1016/j.cose.2014.05.011.
[42] S. Nanda, F. Zafari, C. Decusatis, E. Wedaa, and B. Yang, "Predicting network attack patterns in SDN using machine learning approach," *2016 IEEE Conf. Netw. Funct. Virtualization Softw. Defin. Networks, NFV-SDN 2016*, pp. 167–172, 2017, doi: 10.1109/NFV-SDN.2016.7919493.
[43] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *ICISSP 2018 - Proc. 4th Int. Conf. Inf. Syst. Secur. Priv.*, vol. 2018-Janua, no. Cic, pp. 108–116, 2018, doi: 10.5220/0006639801080116.
[44] A. Ferriyan, A. H. Thamrin, K. Takeda, and J. Murai, "Generating network intrusion detection dataset based on real and encrypted synthetic attack traffic," *Appl. Sci.*, vol. 11, no. 17, 2021, doi: 10.3390/app11177868.
[45] Kurniabudi, D. Stiawan, Darmawijoyo, M. Y. Bin Bin Idris, A. M. Bamhdi, and R. Budiarto, "CICIDS-2017 Dataset Feature Analysis with Information Gain for Anomaly Detection," *IEEE Access*, vol. 8, pp. 132911–132921, 2020, doi: 10.1109/ACCESS.2020.3009843.
[46] I. Sharafaldin, A. Gharib, A. H. Lashkari, and A. A. Ghorbani, "Towards a Reliable Intrusion Detection Benchmark Dataset," *Softw. Netw.*, vol. 2017, no. 1, pp. 177–200, 2017, doi: 10.13052/jsn2445-9739.2017.009.
[47] Z. Nazari, M. Noferesti, and R. Jalili, "DSCA: An inline and adaptive application identification approach in encrypted network traffic," *ACM Int. Conf. Proceeding Ser.*, no. January, pp. 39–43, 2019, doi: 10.1145/3309074.3309102.
[48] A. S. Iliyasu and H. Deng, "Semi-Supervised Encrypted Traffic Classification with Deep Convolutional Generative Adversarial Networks," *IEEE Access*, vol. 8, pp. 118–126, 2020, doi: 10.1109/ACCESS.2019.2962106.
[49] M. E. Mihailescu *et al.*, "The proposition and evaluation of the roedunet-simargl2021 network intrusion detection dataset," *Sensors*, vol. 21, no. 13, pp. 1–20, 2021, doi: 10.3390/s21134319.
[50] A. Alalousi, R. Razif, M. AbuAlhaj, M. Anbar, and S. Nizam, "A Preliminary Performance Evaluation of K-means, KNN and EM Unsupervised Machine Learning Methods for Network Flow Classification," *Int. J. Electr. Comput. Eng.*, vol. 6, no. 2, p. 778, 2016, doi: 10.11591/ijece.v6i2.8909.
[51] Y. Wang, Y. Xiang, and J. Zhang, "A Novel Semi-Supervised Approach for Network Traffic Clustering," pp. 169–175, 2011.
[52] H. Alizadeh, "Traffic Classification and Verification using Unsupervised Learning of Gaussian Mixture Models," 2015.
[53] E. Min, J. Long, Q. Liu, J. Cui, and W. Chen, "TR-IDS : Anomaly-Based Intrusion Detection through Text-Convolutional Neural Network and Random Forest," vol. 2018, 2018.
[54] M. Al-qatf, M. Alhabib, and K. Al-sabahi, "Deep Learning Approach Combining Sparse Autoen- coder with SVM for Network Intrusion Detection," *IEEE Access*, vol. PP, no. c, p. 1, 2018, doi: 10.1109/ACCESS.2018.2869577.
[55] "Malware Traffic Classification Using Convolutional Neural Network for Representation Learning," 2017.
[56] A. Banitalebi and D. Mohammadreza, *The DDoS attacks detection through machine learning and statistical methods in SDN*, no. 0123456789. Springer US, 2020. doi: 10.1007/s11227-020-03323-w.
[57] A. Javadpour, "Feature Selection and Intrusion Detection in Cloud Environment based on Machine Learning Algorithms," 2017, doi: 10.1109/ISPA/IUCC.2017.00215.

[58] K. A. Taher, “Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection,” *2019 Int. Conf. Robot. Signal Process. Tech.*, pp. 643–646, 2019.
[59] D. Stiawan, “CICIDS-2017 Dataset Feature Analysis With Information Gain for Anomaly Detection,” vol. 8, pp. 132911–132921, 2020, doi: 10.1109/ACCESS.2020.3009843.
[60] R. B. Basnet, R. Shash, C. Johnson, L. Walgren, and T. Doleck, “Towards Detecting and Classifying Network Intrusion Traffic Using Deep Learning Frameworks,” vol. 4, pp. 1–17, 2019.
[61] Y. Zhou, H. Shi, Y. Zhao, W. Gao, and W. Zhang, “Based on 2D-CNN Model I :,” *2021 22nd Asia-Pacific Netw. Oper. Manag. Symp.*, pp. 238–241, 2021, doi: 10.23919/APNOMS52696.2021.9562636.
[62] L. Bernaille *et al.*, “Traffic classification on the fly e Salamatian To cite this version : Traffic Classification On The Fly Universit e,” *Comput. Commun. Rev. Assoc. Comput. Mach.*, 2014, [Online]. Available: https://hal.inria.fr/hal-01097551%5Cr
[63] M. Roughan, O. Spatscheck, and N. Duffield, “Class-of-Service Mapping for QoS : A Statistical Signature-based Approach to IP Traffic Classification Categories and Subject Descriptors,” *Acm Imc*, pp. 135–148, 2004, [Online]. Available: http://portal.acm.org/citation.cfm?id=1028788.1028805%0Ahttp://dl.acm.org/citation.cfm?id=1028788.1028805
[64] B. Liu *et al.*, “An Approach Based on the Improved SVM Algorithm for Identifying Malware in Network Traffic,” *Secur. Commun. Networks*, vol. 2021, 2021, doi: 10.1155/2021/5518909.
[65] A. Pradhan, “Network Traffic Classification using Support Vector Machine and Artificial Neural Network,” *Int. Symp. Devices MEMS, Intell. Syst. Commun. ,* no. October 2011, pp. 8–12, 2011.
[66] M. Hayes, B. Ng, A. Pekar, and W. K. G. Seah, “Scalable Architecture for SDN Traffic Classification,” *IEEE Syst. J.*, vol. 12, no. 4, pp. 3203–3214, 2018, doi: 10.1109/JSYST.2017.2690259.
[67] R. H. Serag, M. S. Abdalzaher, H. A. E. A. Elsayed, M. Sobh, M. Krichen, and M. M. Salim, “Machine-Learning-Based Traffic Classification in Software-Defined Networks,” *Electron.*, vol. 13, no. 6, pp. 1–30, 2024, doi: 10.3390/electronics13061108.
[68] F. F. Jurado-Lasso, L. Marchegiani, J. F. Jurado, A. M. Abu-Mahfouz, and X. Fafoutis, “A Survey on Machine Learning Software-Defined Wireless Sensor Networks (ML-SDWSNs): Current Status and Major Challenges,” *IEEE Access*, vol. 10, pp. 23560–23592, 2022, doi: 10.1109/ACCESS.2022.3153521.
[69] S. Schneider, N. P. Satheeschandran, M. Peuster, and H. Karl, “Machine learning for dynamic resource allocation in network function virtualization,” *Proc. 2020 IEEE Conf. Netw. Softwarization Bridg. Gap Between AI Netw. Softwarization, NetSoft 2020*, pp. 122–130, 2020, doi: 10.1109/NetSoft48620.2020.9165348.