Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7684032

INTELLIGENT NETWORK TRAFFIC CLASSIFICATION ON SOFTWARE DEFINE NETWORK USING MACHINE LEARNING TECHNIQUES

Tarun Sharma and Dr. Sharanbasappa Gandage
Department of Computer Science & Engineering, Dr. A. P. J. Abdul Kalam University, Indore (M.P.)
Corresponding Author Email: sharma_tarun_in@yahoo.com

Abstract : Accurate traffic categorization is needed for security monitoring, traffic engineering, fault diagnosis, accounting for network consumption, pricing, and distinguishing the Quality of Service (QoS) of various network services. Classifying network traffic has become a critical problem due to the growth of internet users. With a thousand-fold increase in devices and flows, rudimentary techniques to network traffic classification have failed. In this study, researchers propose a method that combines SDN architecture with machine learning tools to solve the problem. This action is taken to counteract the drawback at hand. There are three main supervised learning models used in a software-defined network architecture to categorise traffic levels of the data dependent on the apps it serves. To name a few, we have models such as the Naive Bayes (NB) classifier, the closest proposed centroid method, and the Support Vector Machine (SVM). After the network traffic traces have been gathered and the flows characteristics have been formed, they are delivered to the classifier so that a prediction may be made. The accuracy that was acquired using SVM was found to be 98.21%, whereas the accuracy gained using NB was found to be 94.29% and proposed centroid method was found to be 99.98%. The difficulties lie in the real network data traffic capturing as well as the categorization of the applications inside the SDN platform.

Keyword: Software Define Network, Naïve Bayes , Support Vector Machine, Quality of Service, Traffic Classification.

I. INTRODUCTION

Enterprise networks have been seeing a significant growth in the volume of data that is being handled and made accessible to consumers, workers, and business partners during the last several decades. As a consequence of this, planning the use of resources and the flow of traffic through a network became an essential component of network administration. In modern networks, traffic engineering (TE) is an essential part of the management process for the networks themselves. Through strategic resource allocation, traffic engineering seeks to improve network performance and reliability.

Software-Defined Decoupling the control plane from the data plane is one of the goals of the networking paradigm known as networking [4]. The centralised control plane is in charge of managing the network's routing as well as the policies that govern it. As a result, it is able to maintain a global perspective of the network by continuously monitoring and compiling data on its state. As a result, it has made it possible to develop several unique ways for monitoring networks [85]. In accordance with the directives given by the controller, the data plane is responsible for the forwarding, deletion, and modification of traffic flows. Machine learning is used effectively in solving issues involving pattern identification and anomaly detection. Its primary objective is to discover and make use of latent patterns in data in order to derive knowledge. This functionality makes it possible to automate complicated operations, including the categorization of traffic, management of resources and security, as well as general network administration. Since a consequence, the integration of machine learning into SDN is a hot subject in academia, as it offers the possibility of bringing novel data-driven techniques to the resolution of standard network problems.

Traffic classification is an intelligent activity that refers to the process of classifying various types of traffic into distinct categories. It has a variety of applications, including network management, service assessment, and network monitoring, among others. In addition, traffic categorization makes it possible to effectively allocate resources and configure access restrictions, quality of service (QoS), and other network security characteristics. However, in today's world, almost all apps operate on dynamic ports, and all network communication is encrypted, so neither of these approaches is useful anymore. For this reason, it is required to build a new categorization method that is more suited to the operating circumstances that are now in place.

The following is a synopsis of the significant contributions that this research made. To begin, we demonstrate that algorithms based on machine learning are capable of effectively classifying and predicting network traffic. We assess the performance of three different supervised machine learning algorithms and compare them based on their ability to classify traffic. This allows us to determine how useful statistical characteristics are.

The remaining parts of the article are structured as described below. In the next section, "Section 2," we will offer an overview of the linked studies. The suggested method is broken down and discussed in Section 3. In Section 4, we provide the results of our analysis of the machine learning models as well as the system as a whole. In the fifth and last section of the paper, we draw to a close.

II. BACKGROUND STUDY AND RELATED WORK

2.1. Application of machine learning to the field of networking

The use of Machine Learning (ML) techniques allows for the detection of intrusions, the optimization of spectrum usage, the attainment of power efficiency, and the control of network traffic. In machine learning, the use of data to make judgments, as opposed to basing conclusions on pre-defined criteria stated in the algorithm, is of the utmost significance. supervised learning, unsupervised learning, and reinforcement learning are the three subcategories of ML algorithms. When doing supervised learning, the usage of labelled data for classification and regression is required. The primary goal of unsupervised learning is to assign each piece of unlabeled data to one of many distinct categories. In the process of reinforcement learning, the agent engages in interaction with the surrounding environment and discovers the behaviour that results in the greatest accumulation of rewards [8].

2.1.1. Learning algorithms for selecting the most efficient paths. The controller for the SDN platform will make adjustments to the flow tables so that it may choose the most efficient route for the traffic. By consulting the flow table rules, the controller has the ability to decide whether traffic should be sent, dropped, or blocked. At the controller level, the machine learning techniques are put to use in order to produce a routing route that has been optimised.

2.1.2. Security: The use of machine learning techniques is detailed in reference [7], In [9], the difficulties associated with ensuring adequate security for IoT networks are outlined.

2.1.3. The goal of supervised learning is to create a database that can be used to classify incoming instances of a flow into preexisting classes in order to make decisions about how to handle them. The relationship between the inputs and the outputs is modeled at various points during the supervised learning process. The methods of training and testing are the two components that make up supervised learning. Training During the learning phase, which is also referred to as the training process, the classification model is developed by evaluating the training data set. This takes place during the learning phase. This takes place during the process of training. With the help of the networking programme known as tcpdump, real-time data saved in the format of pcap files may be retrieved. The training is done using the real-time network traces once the labels have been applied to them. The model that was developed during the training phase is used in the testing phase to conduct the categorization of the newly introduced instance. Supervised learning is used to determine which network traffic class is the result of currently active network traffic. Classifying network traffic requires a tagged data collection. To promote learning and assure accuracy, divide the available data into a training set (maybe 80%) and a testing set (20%). Second, if freshly created network traffic doesn't adhere to known traffic kinds, problems may develop. Real-time traffic classification, also known as online classification, is the third challenge that must be surmounted. Here's the flow instance data used to train ML:

1. The originating IP address, the receiving IP address, etc.
2. Information on the protocol and the length of the header
3. Including the forward and reverse packet counts, packet sizes, intervals, and durations.
4. The current status of the active and idle users, as well as the PUSH and URG counts

2.1.4. Reinforcement learning: The components of reinforcement learning that include agents, a variety of states (S), and action are as follows: (A). Through interaction with its surroundings, the agent is able to acquire knowledge about the behaviour that results in the greatest accumulation of rewards. The function of the controller in the SDN platform is that of an agent. The controller is responsible for performing the monitoring of the network state in order to make judgments on the forwarding of the data.

2.2 Related work

Multi-lane CapsNet assisted network traffic categorization (MANTA) has been proposed to identify and classify 5G network traffic patterns. This design uses multi-lane Capsule Network deep learning (CapsNet). Using deep learning, we compare the model to existing literature. The experimental findings show better performance, with an accuracy of 97.3975% [10].

Modern techniques for analyzing and extracting sound recordings have been adjusted to accommodate for traffic flow. This research examines three processes: To achieve this, we first develop an image representation for the road sound dataset sequences, then propose a convolutional neural network model for feature extraction, and then combine a convolutional

neural network with other machine learning models for classification. To test our method, we gathered road sound data at an urban asymmetric route in the morning and evening. Throughout the day, this happened. When identifying traffic amounts across different time periods, systems show 92% to 95% accuracy [11].
This study refines this approach. By recursively deconstructing the search space, it speeds up search. The modified KD-tree includes a leaf-pushing approach, which enhances classification search efficiency. The technique uses a bloom filter and hash table. The leaf-pushed KD-tree approach exceeded the classic KD-tree by 24 times in packet classification speed. The suggested approach might cut classification time compared to the most advanced tree-based algorithms [12].

This project attempts to create an Intelligent SDN Multi Spike Neural System using an MMSRNN controller and time-based encoding (IMSNS). If we can precisely detect traffic, we can predict the ideal network slice and cut energy use. This research presents a second RNN controller for load balancing and slice failure. Academia using accuracy, precision, recall, and F1-Score. The suggested model outperformed a CNN in simulating a 5G network [13], suggesting that it may be deployed in practice.
In the last step, the hybrid proposed model is implemented by employing the Random Forest (RF) machine learning technique to choose significant characteristics from the combined dataset (which includes V2V and V2R communications). The Gated Recurrent Unit (GRU) algorithm is used to predict network traffic flow since it produces the most accurate results. Simulation findings reveal that the proposed RF-GRU-NTP model surpasses the state-of-the-art algorithms presently employed in network traffic prediction [14] in terms of runtime and forecast errors.

The YOLO-CFNN and YOLO-VCFNN car classification algorithms surpass state-of-the-art approaches on the GRAM-RTM data set with 99% accuracy and F1 values. YOLO-CFNN and YOLO-VCFNN have a high F1 score for vehicle classification and outstanding accuracy in vehicle counts in Taiwan. Both utilize deep learning's convolutional neural networks (CNNs). The suggested system can identify 30 frames per second on the AGX embedded platform. The given intelligent traffic monitoring system classifies and counts autos in real time in their local environments [15].
The multiresolution technique uses a bisection-based decision threshold and a neural network output probability estimate. When applied to genuine traffic circumstances, the recommended technique identifies all traffic abnormalities in the reference test data set. This result shows that the recommended method can detect all traffic irregularities in the reference test data set, improving on our prior approach. We demonstrate that the suggested technique can identify and categorize traffic abnormalities despite two situations: Some traffic characteristics are lost due to V2V or V2I communication deterioration [16], and they are only received from a limited number of vehicles.
In this paper, we present an intelligent reward-based data offloading architecture for next-generation vehicular networks (IR-DON) to optimize data traffic and pick intelligent RSUs. With Q-learning, a reinforcement learning algorithm was built and added to IR-I-ANDSF

DON's module for identifying and choosing networks. I-ANDSF was built as an SDN controller to solve dynamic optimization and offloading. Choosing an appropriate, intelligent RSU enhances system throughput. Simulation findings show that I-ANDSF can identify network traffic, pick the best available network, ensure QoS, decrease latency, and maximize throughput. [17].

Using publicly available surveillance video datasets, the authors show that their approach can effectively identify traffic anomalies such lane violations, abrupt speed changes, mid-movement pauses, and wrong-way driving. Using the suggested high-level features improves trajectory classification by 1% to 6% across a variety of datasets. Gradient representation improves anomaly detection by 30–35%. [18].

The NSGA-II search model exhibited the best population convergence times, model accuracy, Pareto optimality sets, model complexity, and running speeds. Our method exhibited better classification performance at lower computing cost on two public datasets (mostly FLOPs). Our network-based method compares prominent machine learning algorithms to alternative artificial learning techniques. Our method obtained 99.806% on the IDS2012 and ISCX VPN datasets using 11.501 MB and 4.718 MB FLOPs [19].

We analyze current VC trends. We propose a taxonomy of VCs and examine traffic data extraction methodologies. Traditional VC systems and modern innovations are then analyzed. The current VC techniques are assessed for their merits and shortcomings, and real-time alternatives are studied, such as Vehicular Ad-hoc Networks (VANETs). In conclusion, we examine a wide variety of soft computing technologies used in VC, including machine learning, neural networks, and other approaches [20].

Using the Inception-v3 model for feature extraction from images, this research proposes a powerful framework for vehicle categorization. Vehicle categorization is new territory for the cutting-edge Inception-v3 model of deep learning neural networks. The feature vector is then subjected to a number of different categorization methods. Three vehicle datasets were classified, and the results were shown, to help researchers choose datasets with high algorithm performance [21].

In this study, we investigate how machine learning may be used to automatically classify data gathered from networks. In particular, real-world network traffic data is gathered via the ONOS (Open Network Operating System) platform and then automatically categorized using a variety of machine learning approaches. From the results of these tests with simple network topologies, we infer that machine learning algorithms are capable of effectively categorizing the network traffic data. However, it has been shown that if machine algorithms are simply built without any thought, they may only demonstrate limited performance in reality. This is because, in a real network system, there isn't only the data that has to be sent to the clients, but also the data that's used to keep the network running smoothly. It is also crucial to develop machine learning algorithms that explicitly account for the characteristics of the real network traffic data in the intended network scenarios [22].

The primary idea of this study is that feeding morphological information into a deep learning network may increase the classification success rate. For the purposes of this study, deep morphological networks are proposed as a method for making better use of the morphological qualities of traffic signals. Convolutional neural networks and deep morphological networks provide similar results, although the latter has a higher classification success rate [23].

Algorithms for classifying network traffic based on PSO that have been fine-tuned for semi-supervised learning. As part of this research, traditional machine learning algorithms like k-means and KNN are improved with the use of PSO technology. This is carried out due to the limitations of more conventional machine learning methods like k-means and KNN. Finally, we provide the semi-supervised learning approach as a way to get around the intractable time and efficiency issues of supervised learning. We develop a system for classifying network traffic on the basis of the refined algorithm. Experimental results show that the PSO optimised KNN technique has a faster rate of convergence than KNN and is less likely to wander away from the global optimum solution [24].

In this piece, we propose two different classification schemes: the Classification of Flows by Intervals of a Small Number of Packets (CACFP) and the Classification of Flows by Intervals of a Large Number of Packets (CADFP). Both complete flow packet (CEP) classification and CFFP (classification of the first few packets) classification are utilized as benchmarks. The four aforementioned approaches are first compared on a basis of their mutual knowledge of features and applicability. After that, we utilize a live trace to double-check our theory. Using CFFP, the experimental findings suggest that CACFP and CADFP can achieve similar levels of mutual information and classification accuracy. These algorithms work well for online traffic categorization as well and allow for the deployment of a functional system of this kind [25].

With the use of a Moderately Multi-Spike Return Neural Networks (MMSRNN) controller and time-based encoding, this research aims to offer an Intelligent SDN Multi Spike Neural System (IMSNS). If this is accomplished, we will be able to accurately identify traffic for the purpose of anticipating the optimal network slice, and our energy usage will be drastically reduced. A second intelligent Recurrent Neural Network (RNN) controller is presented in this study for load balancing and slice failure scenario. Metrics including accuracy, precision, recall, and F1-Score are now being used by the academics working on this issue. The proposed approach outperformed a convolutional neural network (CNN) in simulations, improving the accuracy of 5G network slicing by 5% [26].

The Deep Convolutional Neural Network, or DCNN, has become popular for classifying traffic signs due to its efficient feature extraction and accurate prediction. However, most efforts in this area only address one aspect, such as accuracy or required parameters, making the work unsuitable for real-time or practical applications. We've created a novel, lightweight neural network that can recognize and categorize traffic signs in roadside photographs in order to address this issue. Our network architecture allows for less use of parameter space without sacrificing accuracy. Note that we used the Belgium Traffic Sign dataset (hence referred to as BelgiumTS [27]) to illustrate the efficacy of our proposed model in terms of accuracy and parameter needs.

III. PROPOSED MODEL

SDN and a machine learning technique for classifying network traffic are both included into the proposed system paradigm. It is a supervised learning method that is used to generate the classification models.

3.1 The Problem Statement: With more and more devices connected to the internet, bandwidth, latency, user experience, and security are all being squeezed. Machine learning-based traffic categorization is suggested to streamline network admin tasks. The architecture in place to classify traffic based on machine learning models in SDN is shown in Fig. 1. The suggested architecture combines the real-time network, the virtual network, and artificial intelligence. When a conventional network is connected to the World Wide Web and other forms of virtual networking, such as those used for e-mail and file sharing, a hybrid network is formed. The virtual space allows for the simulation of various wired and wireless network architectures. The suggested approach uses the AI component for traffic categorization.

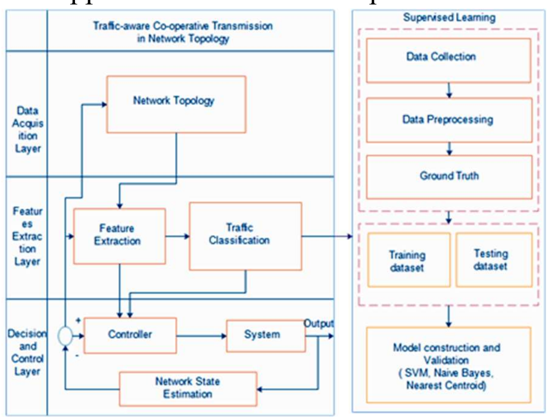

Figure 1 : Working Model

Specifically, the provided formulation is used to pose the traffic categorization issue. Imagine that the collected flow is labelled M = [M1, M2,..., Mn]. There is a single instance Mi with k characteristics. In the ith iteration of the flow, the j th property is referred to by the symbol Mij. Inter-arrival time, length, number of bytes transported, and packet size are the parameters that are relevant to network traffic categorization. Classification of network traffic into a set of categories (C = C1,C2,...,Cq) is the focus of this paper. Streaming media players, web browsers, and email clients are used as examples of Ci classes. The goal is to use machine learning methods to create a mapping procedure from n-dimensional value M to a c-dimensional target C, using the function f(M) C. Offline on the network, while data is being sent, the training data set is recorded. With the use of a classification method and a training set of L tuples of the form, we can figure out how to map M to C, denoted by f(M) (Mi, Ci). Classes C1, C2, and C3 are denoted by the numbers 1, 2, and 3, respectively, in Table 2. Tables 2 and 3 each have three rows, which stand for separate flow examples M1, M2, and M3.

For topology management and remote monitoring, the POX controller is put to good use. Mininet is used to design the network's physical layout. 'tcpdump' creates 'pcap' files, which contain the collected traffic in real time. The 'netmate' flow generator is used to create the flow characteristics. This recorded information has labels affixed to it. Several machine learning methods are used to make the prediction.

3.1.1. Naïve Bayes : By using Bayes' theorem and making the assumption of predictor independence, we are able to derive the network traffic categorization.

$$P((c|m) = \frac{P(m|c)P(c)}{P(m)} \qquad (1)$$

$$P(c|M) = P(m_1|c) * P(m_2|c) \ldots\ldots * P(m_n|c) \qquad (2)$$

A where c is the kind of traffic on the network and M is a single flow, and where P(c|m), P(m|c), P(c), and P(m)are posterior probabilities, likelihood, prior probability, and predictor for network class c, respectively, and P(c) represents the predictor

Algorithm 1: Network traffic classification using the Naïve Bayes algorithm

Inputs: M – Real-time flow instances
Output: Network traffic class of the application.
Steps:
1. The frequency table is obtained for the network training data set.
2. The likelihood table is created by determining the probabilities.
3. Using (1), for each class, the posterior probability is calculated.
4. The network traffic class c is predicted by choosing the highest posterior probability.

3.1.2. Support Vector Machine (SVM) : In support vector machine analysis, the classes are partitioned using a hyperplane. The goal is to locate the best hyperplane that differentiates between the different types of network traffic.

Algorithm 2: Network traffic classification using SVM

Assumptions: The network data set is linearly separable.
Inputs: M – Real-time flow instances
Output: Network traffic class of the application.
Steps:
1. The input flow instances is represented as: $M = \{M_i, C_i\} | \; M_i \in R, C_i \in \{-1,1\}_{i-1}^{n}$
2. Two hyperplanes H_1 and H_2 are selected, such that no data points lie in between them.
3. Determine margin 'm' to maximizes the distance between hyperplane H_1 and H_2

$$m = \frac{2}{\|w\|}$$

Instead of maximising the margin m, the hyperplane that has the lowest value for weight vector w is selected as the optimal solution. The issue of classifying network traffic is shown here as an optimization problem, represented by the equation (3), where b denotes the offset. The letter'm' in lowercase is used to denote margin, while the letter 'M' in capital letters is used to symbolise a single flow instance.

$$\min \|w\|$$
$$subject\ to\ C_i(w^T M_i + b) \geq 1$$
$$for\ i = 1, \ldots\ldots, n \quad (3)$$

3.1.3. Proposed nearest centroid algorithm : Each network traffic class has its own unique centroid that is calculated. It is determined how far, in terms of Euclidean distance, each network class is from its respective data point's respective centroid. By selecting the category that has the shortest distance, one may derive an estimate of the network traffic class.

Algorithm 3: Network traffic classification using Nearest centroid algorithm

Inputs: M – Real-time flow instances
Output: Network traffic class of the application.
Steps:
1. The input flow instances is represented as: $\{(M_1,C_1), (M_2,C_2),\ldots(M_n,C_n)\}$ with the labels $C_i \in C$
2. The centroid is computed for each class using the equation
$\overrightarrow{\mu_\iota} = \frac{1}{|C_i|}\sum_{i=C_i}\overrightarrow{M_\iota}$
3. The predicted network traffic class is given by equation
$\widehat{C} = \arg min_{\iota \in C} \|\overrightarrow{\mu_\iota} - \overrightarrow{M}\|$

IV. IMPLEMENTATION AND RESULT

SDN design categorizes client-generated network traffic. Software-defined networking separates data and control planes (SDN). OpenFlow is the protocol for southbound device-to-controller communication. Northbound interface connects network apps to controller. For decision-making and device management, the controller is supplied information on the network's topological dynamics. Adaptive control and data analytics are recommended to develop suitable control orders and handle network dynamics. The feedback mechanism offers network status updates, such as connection failures or topological changes. The data is utilized to make judgments and remotely control network equipment. Figure 1 shows the dynamic control mechanism's layers. The gathering of information on the network topology is made possible by the data acquisition layer. Training models and AI algorithms are included in the feature extraction layer of the architecture. This layer classifies network traffic. The information that was retrieved about the traffic categorization is made accessible to the controller so that they may make decisions. The decision and control layer is responsible for controlling the devices and making decisions based on the current condition of the network.

The controller is the primary component of the system and serves as its nerve centre. The SDN controller makes it possible to dynamically manage the resources available on the network. Because the controller can see the whole network, it is now possible to keep an eye on the devices. With the help of the POX controller, the devices can send data to each other. Flow rules modification is accomplished with the help of the OpenFlow protocol. The southbound interface connects the topology's networking devices to the controller. Northbound interface allows communication with application plane. For controller communication.

Data gathering, cleaning, labeling, model construction, validation, and prediction are involved in network traffic categorization. SDN load balancing maximizes network resources. SDN uses QoS criteria to categorize traffic.

4.1. Collecting of the Data [28]: Different applications need the collection of data from network traces. Online or offline modes are both viable options for carrying out the traffic categorization. The online collecting of network data gives us the ability to get input that we can use to make the most efficient use of the network's resources. Offline traffic categorization is carried out using the model that has been presented. When gathering information about a network, the software known as 'tcpdump,' which is a packet analyzer, is employed. After the output traces have been redirected, they are recorded in a.pcap file.

4.1.1 Surfing The surfing network traffic class is tracked by recording flow data through the client-server architecture. The 'SimpleHTTPServer' configuration is applied to one of the hosts that makes up the topology. The HTTPServer receives a request from the client to begin the file transfer.

4.1.2. The 'telnet' function is used to produce network traces for the SMTP (Simple Mail Transfer Protocol) protocol used for electronic mail (SMTP).

4.2. Data pre-processing: This step involves transforming the file that was collected into a ".csv" file in order to complete the data pre-processing. For the purpose of obtaining flow statistics, the 'netmate' software is used. Using 'netmate,' a file containing the flow information is created from the collected '.pcap' file and then transformed into a comma-separated file. The training data set has to be labeled so that the system can learn how one flow instance might lead to a certain network traffic class as an output. This is accomplished by providing the system with the ability to learn. The various traffic classifications that were designed are for email, streaming, and surfing the internet.

4.3. Model development and verification: During the building of the model, tasks such as selecting the model, training, and parameter tweaking are carried out. The Naive Bayes model, the SVM model, and the closest centroid technique were chosen as the three models to use. It is possible to apply a variety of machine learning models with the help of the scikit-learn tool.

4.4 Result

A correlation matrix, plotScatterMatrix, confusion matrix, and accuracy graph must be shown here.

,

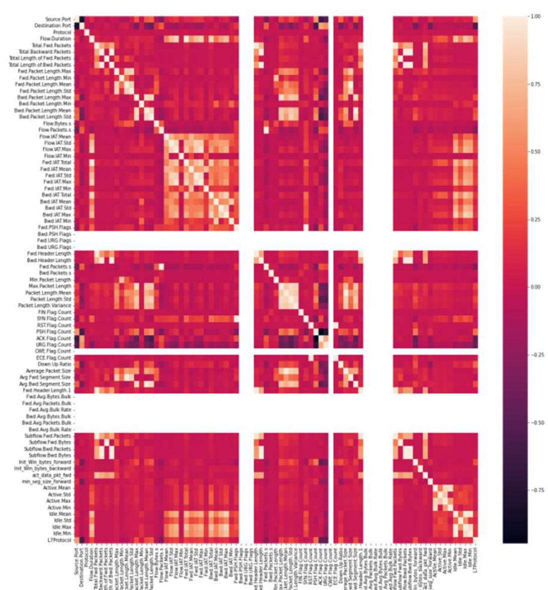

Figure 2. PlotCorrelationMatrix

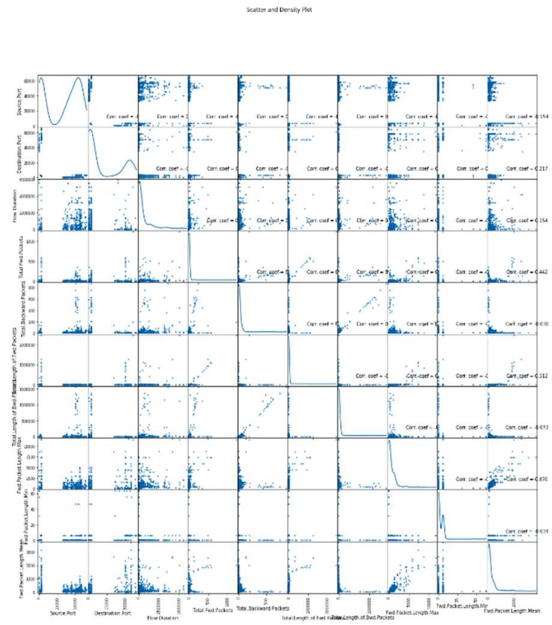

Figure 3. PlotScatterMatrix

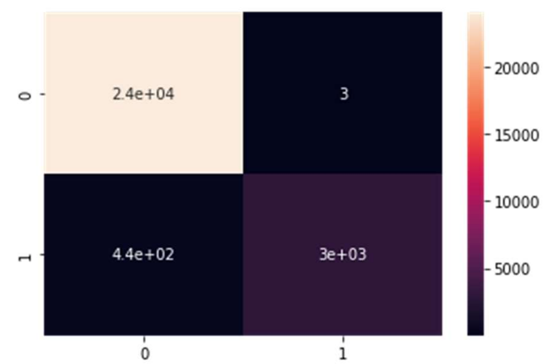

Figure 4. Confusion Matrix of Proposed nearest centroid

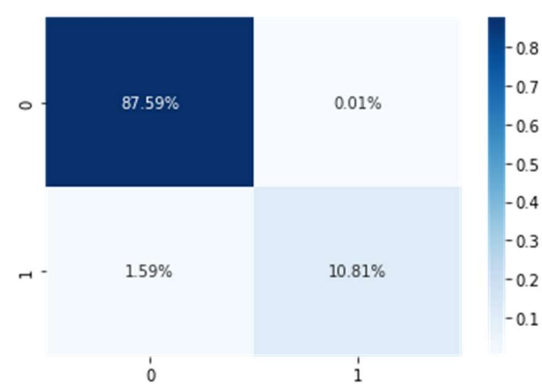

Figure 5. Confusion matrix of SVM

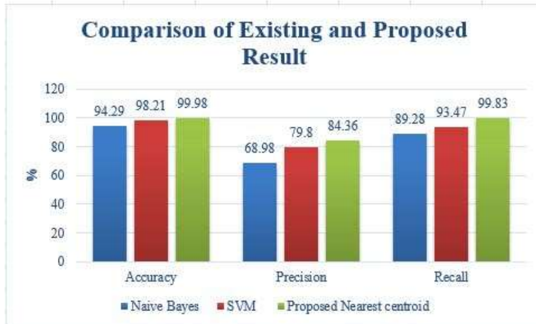

Figure 6. Comparison of existing and proposed result

VI. CONCLUSION

In the context of a software-defined network (SDN), a categorization of data traffic based on supervised learning is presented. Three separate models are used, including the Support Vector Machine (SVM), Naive Bayes (NB), and Nearest Centroid (NC) models. The conventional methods of traffic categorization have limitations as a result of the 2124-fold increase in the amount of data traffic on the network. Three different supervised learning models are able to categorize traffic data with an accuracy of above 99%. In order to make the most use of the network's resources, the proposed framework for traffic classification might be improved by integrating with a system for network slicing.

References

[1] S. Khan, S. Khan, Y. Ali, M. Khalid, Z. Ullah, and S. Mumtaz, ''Highly accurate and reliable wireless network slicing in 5th generation networks: A hybrid deep learning approach,'' J. Netw. Syst. Manage., vol. 30, no. 2, pp. 1–22, Apr. 2022

[2] N. Hu, F. Luan, X. Tian, and C. Wu, ''A novel SDN-based applicationawareness mechanism by using deep learning,'' IEEE Access, vol. 8, pp. 160921–160930, 2020.

[3] N. A. S. Al-Jamali and H. S. Al-Raweshidy, ''Intelligent traffic management and load balance based on spike ISDN-IoT,'' IEEE Syst. J., vol. 15, no. 2, pp. 1640–1651, Jun. 2021.

[4] B. E. Yellakuor, A. A. Moses, Q. Zhen, O. E. Olaosebikan, and Z. Qin, ''A multi-spiking neural network learning model for data classification,'' IEEE Access, vol. 8, pp. 72360–72371, 2020.

[5] J. Xie et al., "A Survey of Machine Learning Techniques Applied to Software Defined Networking (SDN): Research Issues and Challenges," in IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 393-430, Firstquarter 2019.

[6] 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

[7] Joseph Nathanael Witanto and Hyotaek Lim. 2019. Software-Defined Networking Application with Deep Deterministic Policy Gradient (ICCMS 2019). ACM, New York, NY, USA, 176-179. DOI: https://doi.org/10.1145/3307363.3307404

[8] 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.

[9] F. Restuccia, S. D'Oro and T. Melodia, "Securing the Internet of Things in the Age of Machine Learning and Software-Defined Networking," in IEEE Internet of Things Journal, vol. 5, no. 6, pp. 4829-4842, Dec. 2018.

[10] B. Mareri, G. Owusu Boateng, R. Ou, G. Sun, Y. Pang and G. Liu, "MANTA: Multi-Lane Capsule Network Assisted Traffic Classification for 5G Network Slicing," in IEEE Wireless Communications Letters, vol. 11, no. 9, pp. 1905-1909, Sept. 2022, doi: 10.1109/LWC.2022.3186529.

[11] K. -H. N. Bui, H. Oh and H. Yi, "Traffic Density Classification Using Sound Datasets: An Empirical Study on Traffic Flow at Asymmetric Roads," in IEEE Access, vol. 8, pp. 125671-125679, 2020, doi: 10.1109/ACCESS.2020.3007917.

[12] M. Abbasi, H. Rezaei, V. G. Menon, L. Qi and M. R. Khosravi, "Enhancing the Performance of Flow Classification in SDN-Based Intelligent Vehicular Networks," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 7, pp. 4141-4150, July 2021, doi: 10.1109/TITS.2020.3014044.

[13] N. S. Soud, N. A. S. Al-Jamali and H. S. Al-Raweshidy, "Moderately Multispike Return Neural Network for SDN Accurate Traffic Awareness in Effective 5G Network Slicing," in IEEE Access, vol. 10, pp. 73378-73387, 2022, doi: 10.1109/ACCESS.2022.3189354.

[14] S. S. Sepasgozar and S. Pierre, "Network Traffic Prediction Model Considering Road Traffic Parameters Using Artificial Intelligence Methods in VANET," in IEEE Access, vol. 10, pp. 8227-8242, 2022, doi: 10.1109/ACCESS.2022.3144112.

[15] C. -J. Lin and J. -Y. Jhang, "Intelligent Traffic-Monitoring System Based on YOLO and Convolutional Fuzzy Neural Networks," in IEEE Access, vol. 10, pp. 14120-14133, 2022, doi: 10.1109/ACCESS.2022.3147866.

[16] S. Thajchayapong, E. S. Garcia-Trevino and J. A. Barria, "Distributed Classification of Traffic Anomalies Using Microscopic Traffic Variables," in IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 1, pp. 448-458, March 2013, doi: 10.1109/TITS.2012.2220964.

[17] G. Raja, A. Ganapathisubramaniyan, S. Anbalagan, S. B. M. Baskaran, K. Raja and A. K. Bashir, "Intelligent Reward-Based Data Offloading in Next-Generation Vehicular Networks," in IEEE Internet of Things Journal, vol. 7, no. 5, pp. 3747-3758, May 2020, doi: 10.1109/JIOT.2020.2974631.

[18] K. Kumaran Santhosh, D. P. Dogra, P. P. Roy and A. Mitra, "Vehicular Trajectory Classification and Traffic Anomaly Detection in Videos Using a Hybrid CNN-VAE Architecture," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 8, pp. 11891-11902, Aug. 2022, doi: 10.1109/TITS.2021.3108504.

[19] X. Wang et al., "Evolutionary Algorithm-Based and Network Architecture Search-Enabled Multiobjective Traffic Classification," in IEEE Access, vol. 9, pp. 52310-52325, 2021, doi: 10.1109/ACCESS.2021.3068267.

[20] A. Gholamhosseinian and J. Seitz, "Vehicle Classification in Intelligent Transport Systems: An Overview, Methods and Software Perspective," in IEEE Open Journal of Intelligent Transportation Systems, vol. 2, pp. 173-194, 2021, doi: 10.1109/OJITS.2021.3096756.

[21] Zakria, J. Cai, J. Deng, M. S. Khokhar and M. Umar Aftab, "Vehicle Classification Based on Deep Convolutional Neural Networks Model for Traffic Surveillance Systems," 2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 2018, pp. 224-227, doi: 10.1109/ICCWAMTIP.2018.8632593.

[22] J. Kwon, D. Jung and H. Park, "Traffic Data Classification using Machine Learning Algorithms in SDN Networks," 2020 International Conference on Information and Communication Technology Convergence (ICTC), 2020, pp. 1031-1033, doi: 10.1109/ICTC49870.2020.9289174.

[23] F. Zerey, M. Şamil Fidan and E. Uslu, "Detection and Classification of Traffic Signs with Deep Morphological Networks," 2021 Innovations in Intelligent Systems and Applications Conference (ASYU), 2021, pp. 1-6, doi: 10.1109/ASYU52992.2021.9599039.

[24] X. Bian, "PSO Optimized Semi-Supervised Network Traffic Classification Strategy," 2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), 2018, pp. 179-182, doi: 10.1109/ICITBS.2018.00053.

[25] Shupeng Zhao, Z. Chen, L. Peng, Xiaomei Yu and B. Yang, "Online traffic classification based on few sampled packets," 2012 IEEE 14th International Conference on Communication Technology, 2012, pp. 1182-1187, doi: 10.1109/ICCT.2012.6511376.

[26] N. S. Soud, N. A. S. Al-Jamali and H. S. Al-Raweshidy, "Moderately Multispike Return Neural Network for SDN Accurate Traffic Awareness in Effective 5G Network Slicing," in IEEE Access, vol. 10, pp. 73378-73387, 2022, doi: 10.1109/ACCESS.2022.3189354.

[27] A. Kherraki, M. Maqbool and R. E. Ouazzani, "Robust Traffic Signs Classification using Deep Convolutional Neural Network," 2022 International Conference on Intelligent Systems and Computer Vision (ISCV), 2022, pp. 1-6, doi: 10.1109/ISCV54655.2022.9806122.

[28] https://www.kaggle.com/datasets/jsrojas/ip-network-traffic-flows-labeled-with-87-apps