International Journal of Reconfigurable and Embedded Systems (IJRES)
Vol. 15, No. 3, November 2026, pp. 796~806
ISSN: 2089-4864, DOI: 10.11591/ijres.v15.i3.pp796-806

Intelligent flow characterization in SDN using multi-model supervised machine learning for enhanced QoS management

Gunavathie Mariappan[1], Jacophine Susmi Sunny Nayagam[2], Priya Jayaraman[3], Kavya Sabapathy[4]
[1]Department of Information Technology, Easwari Engineering College, Chennai, India
[2]Department of Information Technology, University College of Engineering, Tindivanam, India
[3]Department of Artificial Intelligence and Data Science, Easwari Engineering College, Chennai, India
[4]Department of Computer Science Engineering (Cyber Security), Easwari Engineering College, Chennai, India

Article Info

Article history:

Received Feb 25, 2026
Revised Jun 12, 2026
Accepted Jul 18, 2026

Keywords:

Naïve Bayes
Nearest centroid
Prediction
Software defined network
Support vector machine

ABSTRACT

Precise traffic classification is essential for network operations like traffic engineering, fault detection, network accounting, invoicing, and differentiating the quality of service (QoS) characteristics of distinct network services. The importance of network traffic classification has increased recently as a result of the sharp rise in internet users. The various archaic methods of classifying network traffic have not been able to deliver consistent accuracy due to the increase in both the number of devices and flows. This study proposes the merging of machine learning (ML) technology and software defined network (SDN) architecture to solve this issue. In a SDN platform, three distinct supervised learning models—support vector machine (SVM), nearest centroid (NC), and Naïve Bayes (NB) are utilized to categorize data flow according to its uses. After capturing the network traffic traces, flows features are created and supplied to the classifier for prediction. SVM achieved 93.2% accuracy, NB achieved 97.69% and NC achieved 92.01%. The SDN platform's live network data traffic capture and application classification present issues.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Gunavathie Mariappan
Department of Information Technology, Easwari Engineering College
Ramapuram, Chennai, India
Email*:* gunavathie.ap@gmail.com

1. INTRODUCTION

The network administrator can make informed decisions and allocate resources as efficiently as possible with the help of the data from the network traffic analysis. Internet of things (IoT) data is growing at an exponential rate due to the merging of networking technologies like cloud computing. Managing the resources in response to the rising demand is a difficulty. Considering today's scenario, people want quality of experience (QoE), scalability, fast service delivery, and personalized service level agreements (SLAs). The goal of the service provider is to achieve the greatest profit by allocating resources, using various strategies to satisfy SLAs and QoE requirements, and minimizing operating expenses. Whereas machine learning (ML) places its intricate nature in the network traces gathered and the training algorithm used, classic network traffic classification techniques place their complexity in the source code. Learning ability acquired is the most crucial element of ML algorithms. This skill is acquired via practice and improvement. For each of the many network traffic classes, the statistical characteristics of the network trace—such as duration, packet length, and packet inter-arrival time—are distinct. With the help of the supplied training data set, the machine discovers this distinctiveness in the network pattern. The network data that has been recorded is analyzed using the ML technique to identify any patterns. It can be difficult to identify different applications, like file

transfers, email, video streaming, and surfing. For the best use of the network resources, a powerful network management tool is needed to handle the exponential development in network traffic.

Classifying traffic is a crucial application of high availability and efficiency in network management. Network traffic classification can be applied using the following techniques [1]: understanding application and application layer protocols through the use of port numbers. However, these techniques are not entirely accurate. One method utilized is deep packet inspection (DPI). While dynamic ports and encrypted traffics are not yet enabled in current networks, these solutions have a high accuracy but some implementation challenges. It also compromises user privacy and adds a significant load to the system.

Due to the drawbacks of these approaches, studies have increasingly concentrated on ML strategies that use statistical characteristics to classify traffic. The global perspective of controllers in software defined network (SDN) enhances network management despite the numerous obstacles to traffic classification in modern networks, and its approach of gathering statistical data of network traffic from switches is straightforward and easy to use [2]. The study by [3] examined flows over SDN using traffic classification techniques based on DPI to identify application protocols in runtime. Enabling the controller to discern and isolate distinct application flows, as well as manage and program flows to provide quality of service (QoS) for applications that are sensitive to delays, is their main objective. DPI tools use a significant portion of the controllers' processing power while the network is loaded. The network's overall performance is impacted as a result.

Service traffic in networks is growing quickly as a result of the swift advancement of network communication technologies and the notable rise in network terminal equipment. Simultaneously, the rapidly expanding service flows pose novel problems to the current communication networks. The Internet has undergone a significant transformation thanks to SDN, a new kind of intelligent network architecture. SDN has been used in numerous network scenarios [4]–[6] to satisfy demands for network load balancing, multi-path forwarding, green energy conservation, and centralized automatic administration. But many problems remain unsolvable with conventional approaches, thus some studies incorporate ML techniques into SDN to address specific issues and provide positive outcomes.

Globally, there are exponentially more connections and devices than people and Internet users, which is driving up network traffic [7]. Traffic flow patterns and network performance have also been dramatically altered by the adoption of new, highly capable, and intelligent devices (such as smartphones, smart televisions (TVs), and video game consoles), the growth of machine-to-machine (M2M) communications, and the ensuing creation of new services and applications. It is now difficult to manage various network infrastructures in order to meet the demands of new devices, applications, and services. As a result, there is a lot of interest in developing autonomous networks that use ML [8]. There has been a lot of research done on the use of cognitive networks to handle the complexity of network management and operation. But ML integration presents some additional difficulties. First off, there are no standards in place to ensure network homogeneity between organizations; instead, each has its own network architecture. Second, each network node, such as a router or switch, can only see and act on a certain network segment due to the distributed implementation of traditional network systems.

The control plane and data plane are separated by the SDN paradigm [9]. The management of network policies and routing is the responsibility of the centralized control plane. As a result, it has a global view of the network since it continuously gathers and monitors status data. It has therefore made numerous cutting-edge network monitoring approaches possible [10]. As directed by the controller, the data plane handles forwarding, deleting, and changing traffic flows. ML is effectively applied to pattern identification and anomaly detection challenges, with the goal of discovering and utilizing hidden patterns in data to infer knowledge. Complex operations like handling resources, traffic categorization, security, and generic network administration can be automated thanks to this feature. As a result, the application of ML to SDN is an exciting field of research that offers fresh approaches to solving well-known network issues through the use of cutting-edge data-driven methods.

System administration, service monitoring, and other applications use traffic classification, which is an intelligent process that implies classifying traffic into different groups. Additionally, traffic classification makes it possible to configure access controls, QoS, and other network security characteristics effectively and allocate resources. The port-based method and DPI are two popular traffic classification methods [11], [12]. Both strategies are no longer useful, though, since network communication is now encrypted and the majority of programs operate on dynamic ports. As a result, creating a new classification method more appropriate for the existing operational environment is required.

2. RELATED WORK

Port numbers and DPI are the foundations for the various traffic classification algorithms. Applications are developed using the dynamic ports in situations where port-based classification is not

feasible. When it comes to DPI, a pattern is identified for the intended use. Pattern recognition is accomplished via regular expressions. Pattern-based categorization has a serious problem in that it cannot update the pattern for a large number of applications due to the sudden development in applications. The DPI mechanism is unable to classify the encrypted traffic. Qazi *et al*. explain how ML techniques can classify different applications. In order to gather the actual data, the crowdsourcing method is utilized. The centralized SDN platform controls the policy management. A proposal for network management that takes joint optimization into account for resource usage can be found in [14]. The artificial intelligence (AI)-based methods for managing the network's spatiotemporal traffic fluctuations are provided. In order to forecast controller performance, AI and SDN are integrated, as explained in [15]. The enterprise network traffic is categorized using SDN and ML. Higher accuracy is obtained with supervised learning, as stated in reference [16]. The classification algorithms utilized for spam filtering to prevent cybersecurity assaults are random forest classification, decision tree classification, and K-nearest neighbors (KNN). Several approaches are used for network resource optimization, including the coupling of SDN and AI-based traffic classification.

A system for identifying the application type of current flows in a wireless network which consists of many mobile devices linked to an OF switch was proposed by Arslan *et al*. [17]. The information is sent to a ML-based trainer in the control plane. A method for clustering using flow attributes collected from a dataset is proposed by Zhang *et al.* [18] and used as input for the K-means algorithm during the learning phase. The traffic classification model is implemented using these clusters. The information gleaned from the packet content is used to aggregate the clusters with comparable properties.

A system for classifying applications in smart home networks that use encrypted data flows was presented by Wang *et al*. [19]. Three deep learning (DL) algorithms— multi-layer perceptron, stacked auto-encoder, and convolutional neural network (CNN)—are applied to the preprocessed data first. A DL model in SDN was used by Lim *et al*. [20] to suggest an application classification scheme. Preprocessing the network traffic to train two DL models—a multi-layer long short-term memory (LSTM) and a CNN combined with a single-layer LSTM—to conduct network traffic categorization is how the dataset is created from the functional load of the flows. The optimal hyper-parameter for the models is determined through a model tuning process. The multi-layer LSTM performs the best, according to the data, with an accuracy of more than 99.00%. An architecture that uses the OpenFlow protocol to gather traffic data from an enterprise network was presented by Amaral *et al*. [16]. Both conventional networks and SDNs can use this architecture. The authors classified traffic flows in eight applications by combining multiple ML methods with the collected dataset. The outcomes demonstrate that supervised learning produces performance that is higher than 90.00%. With the use of DL algorithms, Chen *et al*. [21] presented a classification strategy of encrypted traffic to identify applications in an intelligent home gateway. There are two levels in the hierarchical structure of the encrypted packet classifier. A classifier by service type based on applications with comparable QoS requirements makes up the first level. Applications are categorized at the second level using a fine-grained approach to traffic classification. In order to verify the effectiveness of the suggested classifiers, evaluation results were conducted at both levels; the accuracy attained is greater than 85.00% for all applications.

The goal of application-type-based traffic categorization is to identify the various traffic classes in the network that have comparable QoS requirements. In order to enhance service quality, Demirci and Abdulazzaq [22] suggested a traffic engineering system in SDN that is based on DL methods. Traffic flows from different applications are classified into classes with distinct priority by the classifier. The authors used a method known as the synthetic minority over-sampling technique to address the problem of an unbalanced dataset.

To identify applications in classes, Zhang *et al*. [23] developed a deep neural network (DNN)-based traffic categorization approach made up of a SoftMax regression layer and stacked autoencoder (SAE). In order to train the hybrid DNN, the suggested framework gathers and processes network data using the SDN controller. Using the SAE model, the properties of the processed data flows are automatically obtained. The outcomes of the simulation show that the suggested classifier successfully produces a classification accuracy higher than 90%.

Applications can be categorized into traffic classes using a method described by the authors of [24]. The suggested approach lowers the network traffic for categorization and the processing overhead for the drivers by not inspecting the packet payload. Additionally, the authors use ML methods to show how the suggested approach increases classification accuracy. An Open vSwitch (OVS) and two hosts were part of an SDN-enabled network used for the research. With an accuracy of 97.60%, the findings across the testing dataset demonstrate superior performance when compared to the Naïve Bayes (NB) algorithm. Fan and Liu [25] looked at ML-based traffic categorization methods and how feature selection and model fit impact the models' output. Their analysis concentrated on the K-means and support vector machine (SVM) algorithms for grouping traffic generated by applications into ten traffic types.

Using the three ML models of SVM, nearest centroid (NC), and NB, Raikar *et al*. [26] developed a framework for the classification of applications in traffic classes. The framework makes it possible to record network traffic traces. The created flows are then forwarded to the classifier for categorization once the flow statistics are gathered using the NetMate tool. The outcomes demonstrate that 96.79% accuracy is achieved using the NB algorithm. In order to enhance the traffic classification of applications with traffic, Zaki and Chin [27] suggested a hybrid technique called filter-wrapper feature selection (FWFS), which is based on the selection of filtering and wrapping features. Using the C4.5 wrapper, this method first minimizes the amount of features in the dataset before choosing the final features. The computational cost of the performance evaluation with feature selection is low, leading to a robust and dependable model for categorizing fresh data. The accuracy of the classifier is 98.90%.

Deep-SDN, a new DL model for SDN proposed by Malik *et al*. [28], can quickly and reliably identify a variety of traffic applications. The suggested approach is suitable for identifying internet traffic since it can quickly and accurately identify the different application kinds. The authors included a dropout layer in the design to prevent overfitting and enhance regularization. The algorithm's performance evaluation reveals a global accuracy of 96.00%. Amiri *et al*. [29] proposed an approach for allocating bandwidth to gaming traffic. The suggested technique addresses the bandwidth allocation issue that networks in cloud computing data centers face by using ML algorithms to classify incoming traffic flows in real-time while guaranteeing that game flows are given priority over others.

A traffic classification system that Xu *et al*. [30] devised enables allocating different network resources to greatly improve the QoS of different applications. Network functions virtualizations (NFVs) are used to implement the mechanism, and they function at the data plane. As a result, sample data can be transmitted over the SDN switch to the NFV without requiring control channels. There was a traffic classifier model presented by Owusu and Nayak [31]. Utilizing the statistical characteristics of the traffic data, the ML learning models in use classified the traffic according to QoS demands, delay, and throughput. In order to considerably improve the overall efficiency, Mondal *et al*. [32] introduced an innovative performance accelerator algorithm (PAA) that incorporates three ML classifiers. In order to assess the effectiveness, the authors developed a brand-new Docker-based SDN network system and integrated the suggested dynamic network classifier (DNC) into the Ryu controller, which improved QoS by eliminating the need to manually correlate traffic that comes into the network. A traffic classification approach for mobile applications in wireless networks supporting SDN was proposed by Uddin and Nadeem [33].

3. METHOD

The combination of SDN and an ML method for classifying network traffic is explained in the suggested system model. A supervised learning technique is used in the model generation process for classification. As more devices connect to the internet, there are more restrictions on bandwidth, latency, security, and user experience. It is suggested to use ML to classify traffic in order to simplify network administration tasks. The suggested model is divided into three sections: machine intelligence, virtual networks, and real-time networks. When real-world and virtual networks are connected for file transfer and email purposes, a hybrid network is created. Various topologies for both wireless and wired networks are developed in the virtual network.

The suggested model's machine intelligence component is used to classify traffic. The issue of traffic classification is stated as follows. Consider $F=\{F_1, F_2, ..., F_n\}$ to be the collected flow. F_i is an instance with k attributes. When referring to the j^{th} characteristic in the i^{th} occurrence of the flow, F_{ij} is utilized. The characteristics that are used to classify network traffic include inter-arrival time, time span, and the number of bytes and packets that are transported. For the purpose of solving the network traffic classification issue, let $C=\{C_1, C_2, ..., C_q\}$ represent the various classes. The "playing," "surfing," and "e-mail" applications are examples of C_i classes. The goal is to use methods of ML given by function $f(F) \rightarrow C$ to design an algorithm for transforming n-dimension data F to C. While data is being transmitted across the network, the training data set is being recorded offline. By using the classification algorithm on the training data set, which contains T tuples in the form (F_i, C_i), the mapping $f(F) \rightarrow C$ is learned. The training and testing data sets for network traffic classification are subjected to the three supervised learning algorithms—NB, SVM, and NC—and their results are compared in terms of accuracy, precision, recall, and f-score. Topology administration and tracking remotely are handled by the Python OpenFlow (POX) controller. The "mininet" utility is used to construct the network topology. Real-time (continuous) traffic is recorded using "tcpdump" and saved as "pcap" files. The "netmate" flow generator is used to create the flow features. This recorded data has the labels connected to it. There are various ML algorithms used to carry out the prediction.

Using the Bayes theorem, the categorizing of network traffic is achieved, presuming predictor independence.

$$P(c|f) = \frac{P(f|c)P(c)}{P(f)}$$

$$P(c|F) = P(f_1|c) * P(f_2|c) * \cdots * P(f_n|c)$$

The variables P(c|f), P(f |c), P(c), and P(f) represent the posterior probability for network class c, likelihood probability, prior probability of class, and predictor, respectively. The network traffic class c and the flow instance F are also included in this equation. Algorithms 1-3 shows the different classification using ML techniques.

NB is first employed to classify network traffic based on the probability distribution of the observed network flow features. The NB approach estimates the likelihood of each traffic class using the training dataset and assigns the input flow to the class with the highest posterior probability. The steps involved in the NB-based network traffic classification are presented in Algorithm 1.

Algorithm 1. Classifying network traffic with the NB approach
Input: Actual flow instances (F).
Output: The application's network traffic class.
Steps:
1. The network training data set's frequency table is acquired.
2. The probabilities are ascertained in order to produce the likelihood table.
3. By using (1), the posterior probability is determined for every class.
4. The highest posterior probability is selected in order to predict the network traffic class c.

SVM is subsequently employed to classify network traffic by constructing an optimal hyperplane that separates different traffic classes. The objective of the SVM approach is to maximize the margin between the classes while minimizing classification errors. The selection of the optimal hyperplane is based on the weight vector and the distance between the separating hyperplanes. The procedure for SVM-based network traffic classification is described in Algorithm 2.

Algorithm 2. SVM-based network traffic classification
Input: Actual flow instances (F).
Output: The application's network traffic class.
Steps:
1. The occurrences of the input flow are shown as:

$$F = \{(F_i, C_i|F_i) \in R, C_i \in \{-1,1\}\}, i = 1 \; to \; n$$

2. In order to ensure that no data points fall between the two hyperplanes, H1 and H2, they are chosen.
3. Calculate the margin "h" to optimize the separation between the hyperplanes H1 and H2.

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

Rather of optimizing margin "h", the hyperplane with smallest weight vector (w) is selected.

The NC algorithm is then applied to classify network traffic based on the similarity between an input flow and the centroid of each traffic class. For each class, the centroid is calculated from the training flow instances. The Euclidean distance between the input flow and each class centroid is then determined, and the traffic class having the minimum distance is assigned to the input flow. The steps involved in the NC-based network traffic classification are presented in Algorithm 3.

Algorithm 3. Employing the NC algorithm to classify network traffic
Input: Actual flow instances (F).
Output: The application's network traffic class.
Steps:
1. The occurrences of the input flow are indicated as:

$$\{(F1, C1), (F2, C2), \ldots . (Fn, Cn)\}$$

2. The equation is used to calculate the centroid per each class.

$$\overrightarrow{\mu_i} = \frac{1}{c_I} \sum_{i=c_i} \overrightarrow{F_I}$$

3. Equation below yields the expected network traffic class.

$$\hat{C} = \arg min_{l \in d} \| \vec{\mu}_i - \overrightarrow{F} \|$$

Network trace data is gathered for several purposes. One of the two methods for classifying traffic is online or offline. We can get feedback for optimizing the use of network resources thanks to the online network data gathering. The offline traffic classification is carried out in the suggested model. The packet analyzer tool "tcpdump" is used to gather network data. The output traces are diverted and saved in files with the extension ".pcap". The architecture of the client-server is configured to obtain the flow statistics for network traffic class surfing. In the topology, one of the hosts is set up as a "SimpleHTTPServer." The HTTPServer receives a file transfer request from the client. Using the 'telnet' command, network traces for the simple mail transfer protocol are generated. A connection is made across the real-time network and the simulated network structure in order to record the "SMTP" networking traces. The 'vlc-wrapper' class is used to collect the flowing network data between the client and server. The real-time protocol (RTP) is used to transmit music and video data live.

4. RESULTS AND DISCUSSION

The captured file is converted into a ".csv" file as part of the data pre-processing procedure. The flow data are obtained using the NetMate software. Using "NetMate" the captured ".pcap" file is transformed into a comma-separated values (CSV) file with the flow information. In order for the system to understand the connection between the input flow instance and the output network traffic class, the training data set must be labeled. Three distinct traffic classes have been established: email, streaming, and browsing. Model construction involves model selection, training, and parameter adjustment. The closest centroid algorithm, SVM, and NB are the three models that were chosen. Several ML models can be applied using the scikit-learn program. The categorization of the different applications according to the quantity of flow occurrences gathered for five and ten minutes, respectively, is displayed in Figure 1.

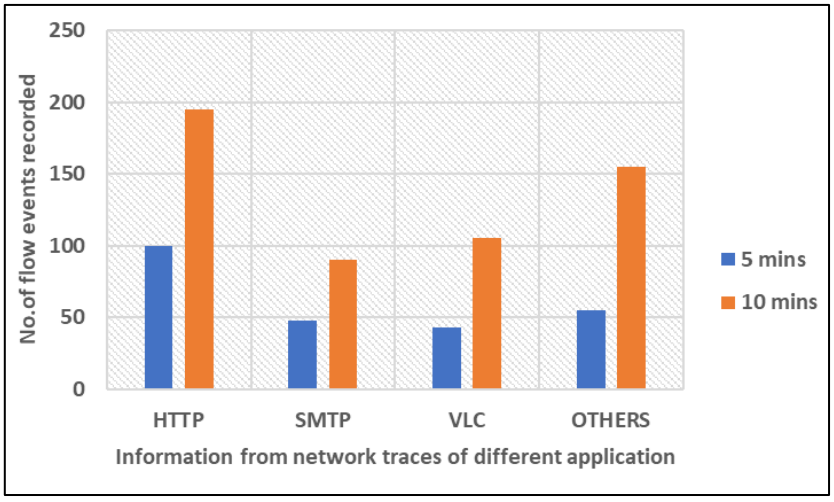


Figure 1. Flow occurrence

The accuracy and training error of the three distinct supervised ML models for the SDN platform's data traffic classification are shown in Figure 2. SVM achieved 93.2% accuracy, NB 97.69%, and the closest centroid 92.01%. The live network data traffic acquisition and application classification in the SDN platform provide obstacles. The SVM has the lowest training error, whereas the closest centroid learning model has a 6% error and NB has a 2% error.

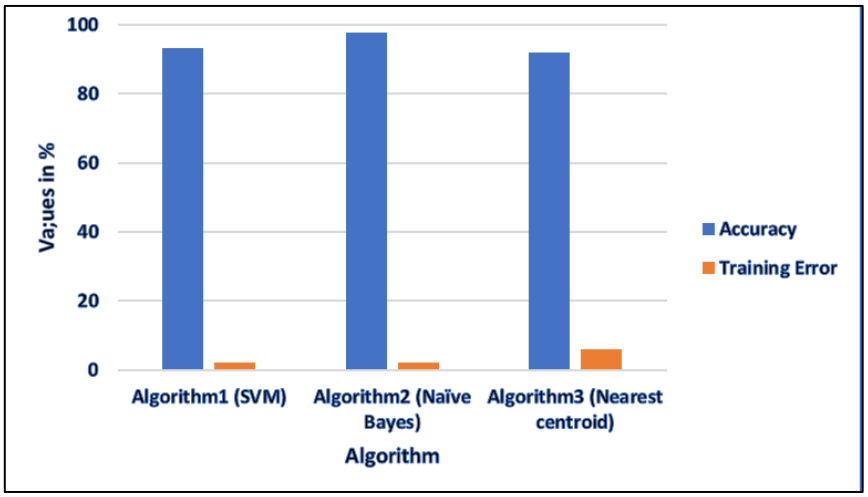


Figure 2. Effectiveness of categorizing network traffic

Figure 3 shows a comparison of the supervised learning models with 150 and 999 as the number of flow examples (n). The graph analysis shows that after 150 flow cases, the precision and recall are reduced. Figure 4 shows training and testing times for classifying network traffic using SVM, which takes 30 ms to train, and NB and NC, which takes 2 ms to train. The testing time in each of the three scenarios is one millisecond. Figure 5 shows a comparison of the various supervised learning models' performances. The NB model yielded 97.69% accuracy.

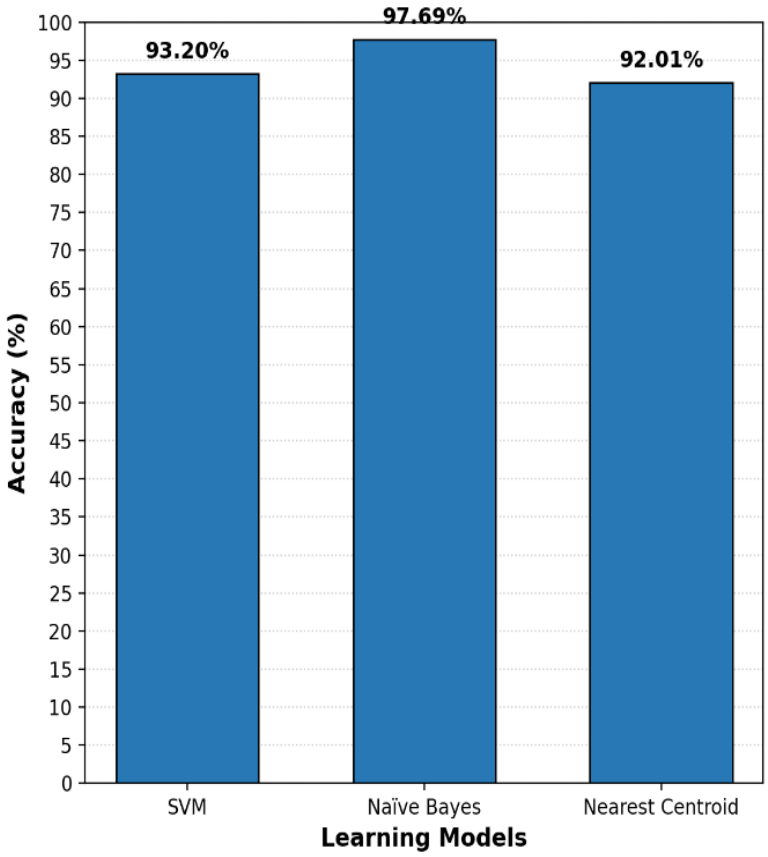


Figure 3. Evaluation of ML algorithms' performances

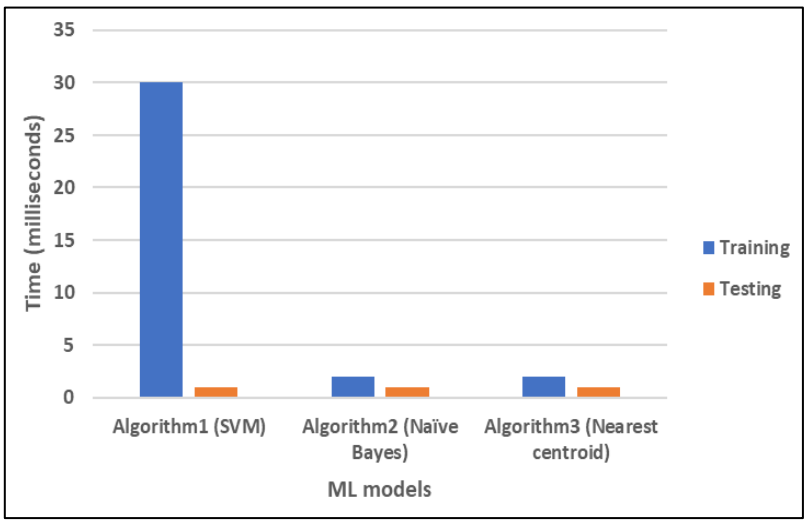


Figure 4. Training and testing time

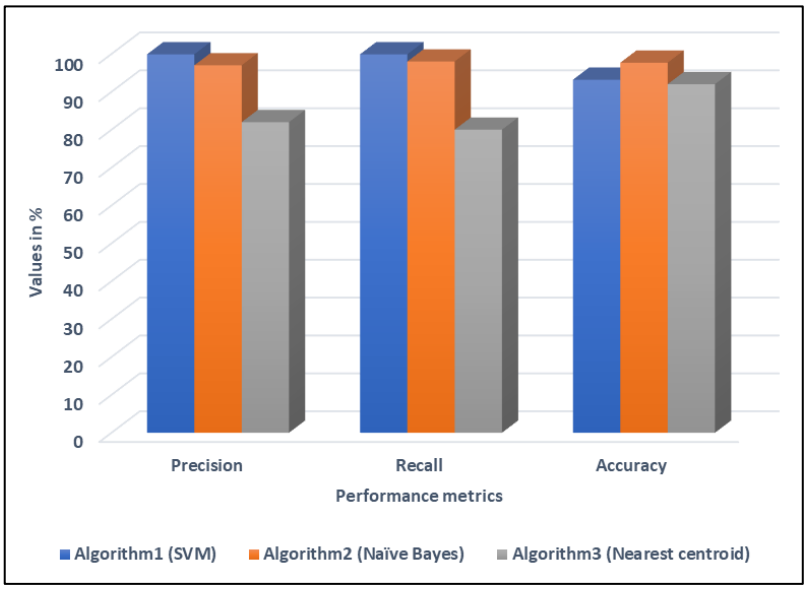


Figure 5. Comparison of precision, recall, and accuracy of the supervised learning models

5. CONCLUSION

This study presented a supervised ML–based approach for network traffic classification in SDN environments. The proposed framework utilizes flow-level traffic features extracted from packet capture files using the NetMate tool, which converts raw network traffic into structured flow records for analysis. The dataset was categorized into three major application classes: email, streaming, and browsing traffic, allowing the system to learn traffic patterns associated with different user activities. Three supervised learning algorithms—SVM, NB, and NC—were implemented and evaluated using the Scikit-learn framework. The experimental results demonstrated that all three models achieved classification accuracy above 90%, indicating their effectiveness in identifying network traffic types in an SDN-based environment. Among the evaluated algorithms, the NB classifier achieved the highest accuracy of 97.69%, while also requiring minimal training time compared with the other models. The SVM model showed strong classification capability with lower training error, whereas the NC algorithm provided competitive performance with very low computational complexity. The results highlight the potential of ML techniques for efficient traffic classification in modern networks experiencing rapid growth in data traffic. In future work, the proposed system can be extended by incorporating DL models and network slicing mechanisms to improve scalability, adaptability, and intelligent resource allocation in next-generation network infrastructures.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Gunavathie Mariappan	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	✓
Jacophine Susmi Sunny Nayagam		✓			✓			✓	✓	✓	✓	✓	✓	
Priya Jayaraman			✓	✓	✓				✓	✓				
Kavya Sabapathy	✓	✓				✓			✓	✓		✓		✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] T. J. Parvat and P. Chandra, "A Novel Approach to Deep Packet Inspection for Intrusion Detection," *Procedia Computer Science*, vol. 45, pp. 506–513, 2015, doi: 10.1016/j.procs.2015.03.091.
[2] N. Williams, S. Zander, and G. Armitage, "A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 5, pp. 5–16, Oct. 2006, doi: 10.1145/1163593.1163596.
[3] H. Cui, Y. Zhu, Y. Yao, L. Yufeng, and Y. Liu, "Design of intelligent capabilities in SDN," in *2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE)*, IEEE, May 2014, pp. 1–5, doi: 10.1109/VITAE.2014.6934459.
[4] S. Jain *et al.*, "B4," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 3–14, Sep. 2013, doi: 10.1145/2534169.2486019.
[5] C.-Y. Hong *et al.*, "Achieving high utilization with software-driven WAN," in *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, New York, NY, USA: ACM, Aug. 2013, pp. 15–26, doi: 10.1145/2486001.2486012.
[6] N. Handigol, S. Seetharaman, M. Flajslik, N. McKeown, and R. Johari, "Plug-n-Serve: Load-Balancing Web Traffic using OpenFlow," *ACM Sigcomm Demo*, vol. 4, no. 5, p. 6, 2009.
[7] CISCO, "Cisco Annual Internet Report (2018–2023)," Digital Transformation powered by Value Engineering, 2020, [Online]: Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.
[8] 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*, New York, NY, USA: ACM, Aug. 2003, pp. 3–10, doi: 10.1145/863955.863957.
[9] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," in *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015, doi: 10.1109/JPROC.2014.2371999.
[10] G. Kakkavas, A. Stamou, V. Karyotis, and S. Papavassiliou, "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, Mar. 2021, doi: 10.1109/MCOM.001.2000458.
[11] IANA, "Service Name and Transport Protocol Port Number Registry." [Online]: Available: https://www.iana.org/assignments/service-names-port-numbers.
[12] 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)*, IEEE, Jul. 2012, pp. 1–9, doi: 10.1109/ICCCN.2012.6289207.
[13] Z. A. Qazi, J. Lee, T. Jin, G. Bellala, M. Arndt, and G. Noubir, "Application-awareness in SDN," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 487–488, Sep. 2013, doi: 10.1145/2534169.2491700.
[14] L. Vigneri, N. Liakopoulos, G. Paschos, and S. Vassilaras, "Model-Driven Artificial Intelligence for Online Network

Optimization," *arXiv Preprint*, 2018, doi: 10.48550/arXiv.1805.12090.
[15] A. Sabbeh, Y. Al-Dunainawi, H. S. Al-Raweshidy, and M. F. Abbod, "Performance prediction of software defined network using an artificial neural network," in *2016 SAI Computing Conference (SAI)*, Jul. 2016, pp. 80–84, doi: 10.1109/SAI.2016.7555965.
[16] 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," in *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, IEEE, Nov. 2016, pp. 1–5, doi: 10.1109/ICNP.2016.7785327.
[17] Z. Arslan, A. Alemdaroglu, and B. Canberk, "A traffic-aware controller design for next generation software defined networks," in *2013 First International Black Sea Conference on Communications and Networking (BlackSeaCom)*, IEEE, Jul. 2013, pp. 167–171, doi: 10.1109/BlackSeaCom.2013.6623403.
[18] J. Zhang, Y. Xiang, W. Zhou, and Y. Wang, "Unsupervised traffic classification using flow statistical properties and IP packet payload," *Journal of Computer and System Sciences*, vol. 79, no. 5, pp. 573–585, Aug. 2013, doi: 10.1016/j.jcss.2012.11.004.
[19] 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, doi: 10.1109/ACCESS.2018.2872430.
[20] 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," *Applied Sciences*, vol. 9, no. 12, pp. 1-16, Jun. 2019, doi: 10.3390/app9122550.
[21] 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, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, 2018, pp. 41–45, doi: 10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00022.
[22] M. Demirci and S. Abdulazzaq, "A Deep Learning Based System for Traffic Engineering in Software Defined Networks," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 8, no. 4, pp. 206–213, Dec. 2020, doi: 10.18201/ijisae.2020466314.
[23] C. Zhang, X. Wang, F. Li, Q. He, and M. Huang, "Deep learning–based network application classification for SDN," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 5, May 2018, doi: 10.1002/ett.3302.
[24] M. Reza, M. Javad, S. Raouf, and R. Javidan, "Network Traffic Classification using Machine Learning Techniques over Software Defined Networks," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 7, pp. 1-7, 2017, doi: 10.14569/IJACSA.2017.080729.
[25] Z. Fan and R. Liu, "Investigation of machine learning based network traffic classification," in *2017 International Symposium on Wireless Communication Systems (ISWCS)*, IEEE, Aug. 2017, pp. 1–6, doi: 10.1109/ISWCS.2017.8108090.
[26] 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 Computer Science*, vol. 171, pp. 2750–2759, 2020, doi: 10.1016/j.procs.2020.04.299.
[27] F. A. M. Zaki and T. S. Chin, "FWFS: Selecting Robust Features Towards Reliable and Stable Traffic Classifier in SDN," *IEEE Access*, vol. 7, pp. 166011–166020, 2019, doi: 10.1109/ACCESS.2019.2953565.
[28] A. Malik, R. de Frein, M. Al-Zeyadi, and J. Andreu-Perez, "Intelligent SDN Traffic Classification Using Deep Learning: Deep-SDN," in *2020 2nd International Conference on Computer Communication and the Internet (ICCCI)*, IEEE, Jun. 2020, pp. 184–189, doi: 10.1109/ICCCI49374.2020.9145971.
[29] M. Amiri, H. Al Osman, and S. Shirmohammadi, "Game-Aware and SDN-Assisted Bandwidth Allocation for Data Center Networks," in *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, IEEE, Apr. 2018, pp. 86–91, doi: 10.1109/MIPR.2018.00023.
[30] J. Xu, J. Wang, Q. Qi, H. Sun, and B. He, "Deep neural networks for application awareness in SDN-based network," in *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, IEEE, Sep. 2018, pp. 1–6, doi: 10.1109/MLSP.2018.8517088.
[31] A. I. Owusu and A. Nayak, "An Intelligent Traffic Classification in SDN-IoT: A Machine Learning Approach," in *2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, IEEE, May 2020, pp. 1–6, doi: 10.1109/BlackSeaCom48709.2020.9235019.
[32] P. K. Mondal, L. P. A. 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, Jul. 2021, doi: 10.1080/09540091.2020.1870437.
[33] 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)*, IEEE, Oct. 2016, pp. 37–46, doi: 10.1109/MASS.2016.016.

BIOGRAPHIES OF AUTHORS



Dr. Gunavathie Mariappan received B.Tech. degree in IT from Mepco Schlenk Engineering College, Sivakasi, Affiliated to Anna University, Chennai in the year 2010. She received her M.Tech. in IT from Madras Institute of Technology, Anna University, Chennai in the year 2012. She received her Ph.D. in the Information Communication Engineering from Anna University in 2024. She is currently working as an assistant professor in Easwari Engineering College, Ramapuram, Chennai. Her research interests include networks, network security, and software defined networking. She can be contacted at email: gunavathie.ap@gmail.com.



Dr. Jacophine Susmi Sunny Nayagam received B.E. degree in Computer Science and Engineering from Prince Shri VenketeshwaraPadmavathy Engineering College, Affiliated to Anna University, Chennai in the year 2006. She received her M.E. in Computer Science and Engineering from M. N. M. Jain Engineering College, Affiliated to Anna University, Chennai in the year 2008. She received her Ph.D. in the Information Communication Engineering from Anna University in 2016. She is currently working as an assistant professor Selection Grade in University College of Engineering Tindivanam, A Constituent College of Anna University, Chennai. Her research interests include data mining and data warehousing, artificial intelligence and machine learning, and clinical applications. She can be contacted at email: jacophine.susmi@gmail.com.



Dr. Priya Jayaraman received her B.E. degree in Computer Science and Engineering from Madras University, Chennai, India, in 2004, and her M.E. degree in Computer Science and Engineering from Anna University, Chennai, India, in 2010. She received her Ph.D. in Information Communication Engineering from Anna University in 2025. Currently, she is working as an assistant professor at Easwari Engineering College, Chennai, India. She has more than ten years of teaching experience. She has published many research papers in international journals and at international conferences. Her research interests include computer vision, medical image processing, artificial intelligence, and machine learning. She can be contacted at email: pria.manikandan@gmail.com.



Kavya Sabapathy received her bachelor of Technology degree in Information Technology from Dr Mahalingam College of Engineering and Technology, Anna University, Coimbatore in 2011 and her master of Engineering degree in Computer Science from Easwari Engineering College, Anna University, Chennai in 2020. From January 2023, she has been a Ph.D. scholar in Information and Communication Engineering at Anna University, Chennai. Currently, she is working as an assistant professor in the Department of Computer Science Engineering (Cyber Security) at Easwari Engineering College, Chennai, India. Her main research interests include wireless body area networks, wireless sensor networks, IoT, and artificial intelligence. She can be contacted at email: kavyasabapathy@gmail.com.