Towards Efficient and Scalable Machine Learning-Based QoS Traffic Classification in Software-Defined Network

M. Z. Fatimah Audah[1], Tan Saw Chin[1](✉),
Y. Zulfadzli[2], C. K. Lee[2], and K. Rizaluddin[3]

[1] Faculty of Computing and Informatics, Multimedia University,
Cyberjaya, Malaysia
sctanl@mmu.edu.my
[2] Faculty of Engineering, Multimedia University, Cyberjaya, Malaysia
[3] Telekom Malaysia Research & Development, Cyberjaya, Malaysia

Abstract. Internet Service Provider (ISP) has the responsibility to fulfill the Quality of Service (QoS) of various types of applications. The centralized network controller in Software Defined Networking (SDN) provides the chance to instil intelligence in managing network resources based on QoS requirements. A fined-grained QoS Traffic Engineering can be realized by identifying different traffic flow types and categorizing them according to various application/classes. Previous methods include port-based classification and Deep Packet Inspection (DPI), which have been found non-accurate and highly computational. Thus, machine learning (ML) based traffic classifier has gained much attention from the research community, which can be seen from an increase number of works being published. This paper identifies the issues in ML-based traffic classification (TC) in order to devised the best solution; i.e. the TC framework should be scalable to accommodate network expansion, can accurately identify flows according to their source applications/classes, while maintaining an efficient run-time and memory requirement. Therefore, based on these findings, this work proposed a TC engine comprises of Training and Feature Selection Module and Classifier Model, which is placed at the data plane. The training and feature selection will be done offline and regularly to keep the Classifier Model updated. In the proposed solution, the SDN switch forwards the packets the Classifier Model, which classify the packets with accurate applications and send them to the control plane. Finally, the controller will perform resource and queue management according to the labeled packets and updates the flow tables via the switch. The proposed solution will be the starting point in solving efficiency and scalability issues in SDN-ISP TC.

Keywords: Traffic Classification · Software Defined Networking · Machine Learning

1 Introduction

The biggest challenge for an Internet Service Provider (ISP) is to cope with the Quality of Service (QoS) requirements for increasing number and various types of applications. The term QoS refers to a level of assurance for a network element; e.g. router and

I. Awan et al. (Eds.): MobiWIS 2019, LNCS 11673, pp. 217–229, 2019.
https://doi.org/10.1007/978-3-030-27192-3_17

application, that it's traffic and service requirements can be satisfied [1]. Though the same links might be traversed by traffic of different applications, they will not be of the same priority, bandwidth and latency requirements. For example, video traffic may allow packet drops but requires low latency, which is the opposite of bulk transfer traffic. Voice over Internet Protocol (VoIP) traffic demands little bandwidth but sensitive to time delay. Surveillance video requires low latency and large bandwidth to keep it flowing. Heavy usage of multimedia applications such as video on demand or applications such as voice on IP have created a great challenge for ISP to ensure their subscribers have sufficient bandwidth for QoS provisioning. To accommodate the huge increase of internet traffic, ISPs have to provide more facilities to increase network throughput, especially during peak-hours. However, fluctuation of traffic demand between peak and off-peak period usually left a significant percentage of bandwidth unused [2]. Furthermore, solutions with an objective to offer more capacity will eventually fail as it will be used up by the ever increasing traffic.

Software-Defined Networking (SDN) carries new possibilities to provide intelligence in the network. By splitting the data plane and control plane of a network and communicate using the southbound API; i.e. OpenFlow protocol, more flexible infrastructure can be implemented. Therefore, ISPs can control the network more efficiently to provide the best services for their subscribers. The keystone of SDN is centralized and software-based network's control, allowing clear communication with network resources according to the applications requirements. The forwarding decisions in OpenFlow based network devices can be controlled by the software. Therefore, ISPs can improve traffic flows management with respect to their QoS demands.

In this work, we propose a QoS-aware SDN Internet traffic classification framework based on Machine Learning (ML), to classify traffic flows which can improve resources allocation through efficient traffic management. Besides being highly accurate, using ML-based classifier eliminates the need to examine packet content. In order to allocate resources efficiently, SDN controller need a quick and accurate identification of the network traffic flows. However, currently, SDN are unable to provide a detail and refined flow control [3]. Besides the information in Layer 1, 2, 3 and 4 of the Open Systems Interconnection (OSI) model, OpenFlow does not has access to the application layer information. In addition, flow table is being generated only based on the information derived from Ternary Content Addressable Memory (TCAM). TCAM offers fast lookup, thus is a preferred classification method. However, its drawbacks; i.e. limited memory, excessive power usage and complex conversion of rule ternary, will soon make it unreliable [4].

Internet traffic classification has been rigorously studied by researches, where various approaches have been developed and proposed to address issues imposed by the techniques. Traffic classification of internet traffic offers a fined-grained QoS Traffic Engineering by identifying different traffic flow types and categorizing them according to various application/classes [3]. It is also the answer to network management problems for ISPs and their equipment vendors, where the outcome is a more efficient network resources allocation. By implementing traffic classification techniques in ISP, traffic patterns; i.e. the time and endpoints in which packets being exchange, and classes of applications can be identified [5].

In SDN, by having a centralized controller, decision making process is done solely by the controller, while the switches become programmable simple forwarders. This architecture is a perfect platform for improvements in networking as it is highly manageable, adaptable, flexible and simple. By taking advantage of these properties, traffic flows can be managed based on QoS requirements and existing resources. Unlike conventional best effort network service, which lack traffic control, SDN controller can offer guaranteed applications-specific QoS [6]. However, in order to deliver this, efficient and flexible traffic management, routing and flow scheduling are needed to manage packets of various applications. Some works have explored this issue through flow balancing [7], fair queuing scheme [8] and workload merging scheme [9]. In this paper, we intend to address the problem by proposing an ML-based framework for Internet traffic classification, which then can be applied for traffic management, routing and flow scheduling to satisfy QoS requirements.

The remaining of the paper is organized as follows. Related works on ML-based Traffic Classification is presented in Sect. 2, while Sect. 3 described the Overview of SDN Traffic Classification Framework. The future direction of this research is given in Sect. 4. Finally, the paper concludes in Sect. 4.

2 Related Works

2.1 ML-Based Traffic Classification

Traffic classification (TC) techniques commonly inspects packets' content on the network to identify the types of classes or the source application of the packet. Packets with similar 5-tuple; i.e. protocol, source and destination address, source and destination port, belong to the same flow which its application is being identified. The approach used for traffic classification includes port-based [10], Deep Packet Inspection (DPI) [11–13] and statistical-based [14–27]. Port-based classification is among the earliest techniques and is simple and fast. It uses the ports in the TCP/UDP header to map applications with renowned TCP/UDP port numbers. Unfortunately, this approach is no longer accurate since dynamic port numbers being used for many applications or transported via end-to-end encrypted channels [5]. DPI method examines packet contents to search for data that is application-specific. It identifies traffic flow's application by matching its payload with predefined patterns. Though it can accurately classify flows, it has a few disadvantages [15, 16]. First, this technique assumes that packet contents is visible for inspection, which relates to data privacy. Second, the increasing number of applications not only complicates pattern updates, but impractical. Third, DPI is high in computational cost as more patterns need to be compared with increasing applications. Fourth, maintaining a database containing all applications is costly. Finally, encrypted traffic is impossible to be classified by DPI.

On the other hand, statistical-based method can classify traffic flows without the need of deep inspection of packet contents. This technique assumes that the statistical properties of each traffic at the network layer are similar for applications with the same QoS requirements. Thus, different source applications can be distinguished from one another. By recognizing statistical patterns in the flows' features, such as arrival times of inter-packet, the first few packets' size, IP address, packet length, flows' duration, round trip time and source/destination ports [17], the technique can classify them into groups with similar patterns. Apart from some customized algorithms, many researchers are looking at Machine Learning (ML) techniques as an alternative to DPI. ML based traffic classification is seen as the future as it has much lower computational cost and can recognize encrypted traffic.

Traffic classification with ML technique requires a large number of traffic flows to be collected. The flows' features or attributes are then being used as the input data to train ML algorithm and classify them into predefined traffic classes. Finally, the trained model can be used in real-time to classify unknown traffic flows using the learned rules. This work concentrates on the development of ML classifier to classify internet traffic in SDN. Acquiring the traffic flows and analyzing them can be carried out in the control plane because the controller in SDN is centralized and has the overall network view.

ML based traffic classification can be carried out in different perspective; i.e. application-aware [4, 14, 18–20] and QoS-aware [6, 16, 21–26]. The former objective is to classify traffic flows' applications; i.e. Youtube, Facebook, Spotify, Tumblr, Skype, etc., while the latter is to classify traffic flows' classes; i.e. Audio, Browsing, Chat, Mail, Video, etc. In QoS-aware traffic classification, several applications might have the same QoS requirement, hence falls under the same QoS class. Some argues that classifying the flows based on their classes is more effective because it is almost impossible to identify all applications on the Internet as the number is growing exponentially. Nevertheless, by identifying either the application or classes of a traffic flow, an efficient route can be chosen in order to ensure its QoS is met.

In [24], QoS-aware traffic classifications are proposed using DPI and semi-supervised ML. DPI is used to identify the flows and were tagged with their applications, forming a partially labelled dataset. Then, using this dataset, a classifier is trained and different application flows are sorted accordingly into 4 QoS categories; i.e. Voice, Video, Bulk Transfer and interactive data. The frameworks use DPI to maintain a partially labelled database that is dynamically updated to retrain the classifier. Therefore, it will be able to recognize new applications which is rapidly deployed from time to time. However, maintaining a database in the control plane leads to scalability issue in the future.

In [26], both DPI and ML algorithm are used as the classifier. For every flow, ML will be used first as the classifier and its result will be accepted or rejected based on a threshold. The threshold can be dynamically changed according to classifier's accuracy. DPI will be used to classify the flow if the ML result is rejected. The authors intend to use ML as it is fast, but because DPI accuracy in classifying flows are higher, it is used as a check and balance. However, the threshold which determines the acceptance of ML result might be set too low, which leads to many DPI classifications to be done, hence taking longer time and more computing resources. Similar to [26], both DPI and ML algorithm are used as the classifier in [25]. While [26] will use either DPI or ML as the classifier, [25] uses both. Once an elephant flow is detected, the applications will be identified first by DPI, then classified into classes using Support Vector Machine (SVM). Though the accuracy exceeded 90%, the use of both DPI and ML as classifiers incur high computational cost.

The classification scheme proposed in [6] uses decision tree to classify incoming traffic flows based on features such as port number, order index and packet size. Priority number is assigned for each classes; i.e. Voice: High, Video: Medium and Data: Lowest, which will be used for a queuing model that is designed to manage the waiting time of the packets. An experimental SDN setup consisting of four nodes is then used to test the proposed queuing scheme with respect to FIFO. The delay measured in the proposed scheme is 93% better than FIFO scheme, which does not implement traffic classification. This has proved the significant of traffic classification in traffic flow management. The authors of [19] have shown that the collection of traffic data using OpenFlow protocol can be done with a single OpenFlow switch deployed in hybrid-SDN enterprise network. The classifier used Random Forest, Stochastic Gradient Boosting and Extreme Gradient Boosting to classify the network traffic according to 8 applications; i.e Bittorent, Dropbox, Facebook, HTTP, LinkedIn, Skype, Vimeo and Youtube. Results show that classifiers' accuracy in classifying Vimeo and LinkedIn is only 71% to 76%, compared to other applications which exceed 83%. This might be because these applications are similar to other application in their classes; i.e. Vimeo is similar to Youtube and LinkedIn is similar to Facebook, which has similar statistical patterns.

An application-aware classification framework for mobile applications, called Atlas [18], used crowd sourcing to gather training data from mobile devices. A decision tree is trained and the model is used to classify the top 40 applications on Google with over 94% accuracy. Although the proposed framework is simple and produce reliable ground truth data, the actual implementation is very limited as personal mobile devices need to be accessed in order to collect data from them.

The works in [14] developed a traffic classification framework for an SDN home gateway (HGW). Three classifiers are used to classify encrypted traffic according to applications; i.e. multilayer perceptron (MLP), stacked autoencoder (SAE), and convolutional neural networks (CNN). To provide real-time network management for SDN-HGW, the classification must be carried out at the data plane, with limited CPU power and memory. Though the classifiers' accuracy exceeded 95%, CPU usage is more than 60%, leaving little spare capacity for other programs.

The authors in [20] argues that QoS-aware classifications unable to satisfy QoS requirement of multimedia applications. For example, both Netflix and Livestream generate video streaming, but Netflix has larger streaming buffer than Livestream, thus better adaptability to bandwidth fluctuation. In addition, the same mobile application can generate multiple types of flows, e.g. voice, video and file sharing can be done with Skype. Therefore, a more detailed traffic classification that can identify the applications as well as flow types is needed. The proposed framework first gathers ground-truth training data available from mobile devices. Then, the application name is identified by a decision tree classifier, and finally a k-NN classifier will classify the flow type/class. The accuracy of application identifier is 95.5%, while 98% for classes classifier. Combining both application and QoS-aware classification could be the best method for traffic classification in Internet traffic, as it provides fine-grained classification which can lead to better routing and resources management. Table 1 provides the summary of the works discussed above.

Table 1. ML-based traffic classification in SDN

Ref.	Learning model	Accuracy	Advantages	Shortcomings	TC placement
[23]	Semi-supervised learning	>90%	Only require partially labelled training dataset	Maintaining a database in the control plane leads to scalability issue in the future	Control Plane
[26]	ML model	>85%	DPI is used as check and balance	Long run-time and high computational cost	Control Plane
[25]	Semi-supervised learning	>90%	Identifies the applications and classes in elephant flows	The use of both DPI and ML as classifiers incur high computational cost	Control Plane
[19]	Random Forest	86.4%	Ground-truth data obtained from a hybrid-SDN	Focusing only on TCP traffic	Control Plane
[18]	Decision Tree	94%	Crowd sourced ground-truth data from mobile devices	Classify only 40 most popular applications	Control Plane
[20]	Decision Tree	95.5%	Classified the flows according to applications and classes	The proposed framework assumes that end users' devices can receive command from the controller	Control Plane
	k-NN	>90%			
[14]	Deep Neural Network	>95%	Classified encrypted data	Real-time classification will cause high CPU usage	Data Plane

Based on the works described above, a number of issues have been identified. First, a traffic classification framework should be scalable to accommodate network expansion. In [23], although the use of DPI produces accurate ground-truth training data, maintaining a database of the patterns for every application will result to storage problem on the control plane. Hence, there should be an alternative to DPI for supervised and semi-supervised ML algorithm to run effectively.

Second, traffic classification framework should be able to accurately identify flows according to their source applications/classes. The authors of [21] and [22] have shown that the classification accuracy is highly dependent on the volume and dimension of the training datasets. Pre-processing of the training data, which involves feature selection method and the acceptable number of each application/classes data, will determine the overall accuracy of an algorithm, as well as per applications/classes accuracy.

The authors in [18] have taken advantage of device management software agents on enterprises' employees mobile devices to collect ground-truth data, while [19, 27], have collected data on their network campus over a period of time, while [14, 22] used published datasets to train their model. Therefore, it is essential to obtain a dataset which has a balance number of all applications/classes that we intend to classify, and formulate an effective feature selection method to achieve maximum accuracy.

Finally, the run-time and memory requirement for a proposed algorithm should not cause resource exhaustion. As mentioned above, [26] will take longer time to classify the flows if the threshold is set too low. Meanwhile, though deep learning is more effective in classifying high dimensional data, it consumes lots of computing resources as shown in [14] and [16]. Thus, the proposed method should aim for accuracy without neglecting the amount of computing resources that can be used to run traffic classification without crashing the system.

2.2 SDN Traffic Classification Framework

The works in [3, 6, 18–23, 25] and [4] suggested that, since the controller has a global network view, traffic collection and analysis can be carried out in the control plane. For every new flow, its statistics will be extracted and passed from the data plane to the control plane where its application/class will be identified. However, it has been proven in [28] that the performance of a controller degraded with high rates of new flows; i.e. for 125 new flows per second (NFPS), the CPU controller usage reaches 100%. Note that, the first n packets of each new flow need to be sent to the controller for classification process, causing resource exhaustion of the control plane. In addition, as the network size expands, so as the interconnection with network elements, thus increasing the demands on the controller capacity and efficiency. Therefore, placing the classifier in the control plane might cause the controller to eventually fail in delivering its tasks.

In contrast, the works in [14, 16] and [28] locates TC engine in the data plane. Both [16] and [14] used deep learning as their classifier, while [28] does not used ML-based classifier. The accuracy of deep learning is highly dependent on the volume and dimension of the training data. Its potential will be affected if the sampling data are subjected to bandwidth constraints. In SDN, even though packets sampling can be done by the controller, a large amount of sampled data will occupy the limited bandwidth, which might lead to communication problem between controller and the switches. Therefore, the authors in [16] have proposed to use Virtualized Network Function (VNF) to deploy TC using Deep Neural Network (DNN). The VNF runs in virtual machines on top of the hardware infrastructure of the data plane. For data sampling, packets can be forwarded to the VNF without interrupting the control channel. Meanwhile, once packets have been identified by the DNN model, it will be sent to the controller for flow table updates. The benefits of having a VNF to run TC is that the network functions will not be affected if the VNF crashes. However, there is no evaluation on the memory requirement and the run-time of the classifier on the VNF. Hence, no further conclusion can be made on the network performance.

On the other hand, in [14], the deep learning training and model updates are being carried out in the application plane, which has more computing resources than the data plane. Though the authors do not specify the source and procedure of data sampling, decoupling this from the data plane increases the network scalability. In this work, the computational performance analysis for running deep learning classifiers have been reported. The CPU usage of classifying 150,000 packets are between 7%–9% for SAE, 19%–22% for MLP and 58%–65% for CNN classifier. Therefore, it can be concluded that the choice of ML classifier is relatively important towards the scalability of the whole network.

The authors of [28] has prototyped an SDN TC architecture on OpenFlow1.3 in the data plane and have shown improvements on network's performance and scalability. A device called Data Plane Auxiliary Engine (DPAE) has been introduced as a network element to offload TC workload from control plane to data plane. Upon flow classification, relevant information will be sent to controller via a new protocol, namely DPAE to Controller communications (D-CPI). D-CPI is proposed to automate the configuration of DPAE and communicate TC results with the controller. However, note that the classifier used is not ML-based and the accuracy has not been discussed. Two classifiers have been implemented, namely Link Layer Discovery Protocol (LLDP) identity classification and statistical classification. The LLDP identity can be identified by matching against LLDP features defined as a regular expression or using Python's in operator. Meanwhile, the statistical classifier is to match the packets as a flow according to their TCP port numbers and IP addresses. The proposed architecture resulted to a linear increase of DPAE CPU usage and recorded only 50% of CPU usage against 500 NFPS. This has significantly improved over their earlier findings with TC located at the control plane; i.e. 100% CPU usage for 125 NFPS. Therefore, it can be concluded that placing the TC engine on the data plane has higher chance of having a scalable network architecture.

To the best of our knowledge, there is no study that has carried out an experimental evaluation on the scalability of ML-based TC in SDN. However, based on the works of [14, 16] and [28], we can conclude that both the types of algorithm and TC engine placement contributes to the performance and scalability of TC in SDN. Therefore, we intend to focus on these properties in proposing the framework of TC solution. In terms of accuracy, the works by other researchers have proven to produce accurate classifier, but only [26] and [14] evaluate the run-time and memory requirement needed to finish the classification task. However, [26] recorded low throughput with low accuracy of ML classifier, making the need to run DPI increases, while [14] consumes lots of computing resources to run deep learning classifier. Therefore, we will focus on the efficiency of the classifier in terms of its run-time and memory requirement without degrading its accuracy.

3 Scalability-Aware SDN Internet ML-Based QoS-Aware Traffic Classification Framework

According to the issues of ML-based TC in SDN that have been highlighted in Sect. 2, this work outlines two objectives to be achieved. First, to propose a traffic classifier model with high classification accuracy, short run-time and less memory requirement. To achieve this, a quality dataset with balance number of applications/classes should be acquired and an effective feature selection method should be devised.

This work will use Naïve Bayes algorithm as it is possible to be trained with a small training dataset without overfitting. The classifier has faster convergence time than discriminative model, such as logistic regression. Since our focus is on the efficiency and accuracy of the classifier, choosing this algorithm will ensure fast and light computing requirement. The model will be evaluated based on how accurate the model's decisions on previously unseen data. Three evaluation metrics will be used; i.e. Accuracy, Recall and Precision. Accuracy is defined as the percentage of correctly classified flows among all flow, while Recall and Precision are measured on per-class basis. Recall is a measure of the model's ability to identify all classes of interest in the dataset, as below:

$$Recall = \frac{Number\ of\ True\ Positive}{Number\ of\ True\ Positive + Number\ of\ False\ Negative} \tag{1}$$

where True Positive is the number of flows correctly labelled as X, while False Negative is the number of flows belong to X but incorrectly labelled as other than X.

On the other hand, Precision is a measure of the model's ability to correctly identify all flows in the dataset. The formula for Precision is given as below:

$$Precision = \frac{Number\ of\ True\ Positive}{Number\ of\ True\ Positive + Number\ of\ False\ Positive} \tag{2}$$

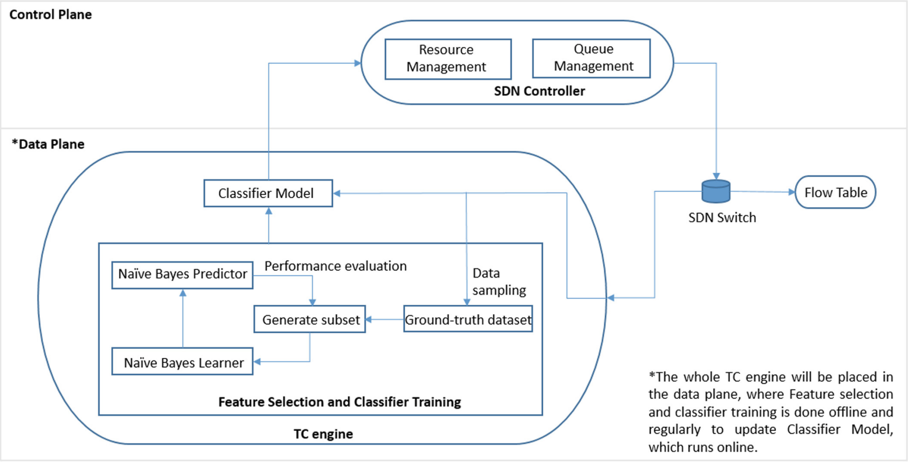
where False Positive is the number of flows incorrectly labelled as X.

To achieve high percentage of these metrics, redundant and irrelevant features should be removed. Dataset with only the most relevant features not only improves model accuracy, but also reduces model complexity, thus requires less computational cost. There are two types of feature selection; i.e. filter and wrapper method. The filter method has lesser computational cost, but the wrapper method, results in higher learning performance [29]. In this work, forward feature selection with Naïve Bayes learner and predictor will be implemented. Subsets of features will be used to train the model and the decision to add or remove a feature is based on their implications on the trained model. The process begins without a feature in the model and adding ones from the original set in each iteration. Subsequently, the features that improves the model performance will be kept in the list and the process stops once the addition of a new feature does not contribute to the model performance. Finally, only the features in the list will be used as inputs to the classifier.

Naïve Bayes classifier with various feature selection technique has been widely used in research works to classify Internet traffic in traditional network architecture.

However, apart from [15], to the best of our knowledge, there is no other works using this type of classifier in SDN architecture. The work in [15] proposed a classifier based on a modified Naïve Bayes, where the algorithm drops the assumption of attributes' independence. The accuracy of the algorithm is first compared to Naïve Bayes' accuracy which recorded slight difference; i.e. 98.8% for the proposed algorithm and 85.25% for Naïve Bayes. The authors then applied feature selection method on both algorithms which resulted to an improved accuracy of Naïve Bayes; i.e. 90.05%, while no improvement seen on their algorithm. The proposed algorithm also has shorter run-time, both before and after feature selection is done, which is crucial in video streaming. It is well known that the prediction accuracy of Naïve Bayes algorithm suffers from the existence of irrelevant and redundant features. Hence, a feature selection method must be devised to improve the learning performance. However, the method used in [15] was only based on filter method, which depends on the features' scores in several statistical test, without involving any learning algorithm. Realizing the importance of feature selection method, we suggest wrapper methods based on greedy search algorithm, where Naïve Bayes performance will be evaluated for every possible combinations of features. This method tends to find the best features that match the learning algorithm; i.e. Naïve Bayes, resulted to improved learning performance. Furthermore, the works in [15] focused only on classifying video streaming traffic; i.e. NetFlix and YouTube which is far than enough looking at the continuous increment of the number of internet applications. Furthermore, though the run-time has been compared to Naïve Bayes, the memory requirement of the classifier has not been evaluated, which is important to ensure scalable solution. In [15], only 11 features used in the classification, while in our work, more features will be included in the dataset and the importance of each feature will be evaluated by the forward feature selection technique. By doing this, the application types can be narrowed down which can further contribute to network resource management.

The second objective is to propose a scalable TC framework that can accommodate network expansion. Based on the studies in [28], it is evident that the TC engine should be placed in the data plane to avoid performance degradation of the controller. The high-level TC framework is shown in Fig. 1. The whole TC engine will be placed in the data plane, where Feature selection and classifier training is done offline and regularly to update Classifier Model, which runs online. By having the TC engine placed in the data plane, the possibility of overloading control plane's resources can be avoided. Looking at the current network expansion rate, the demands on the controller capacity and efficiency rapidly increases, which can cause controller failure. In our proposed solution, instead of sending the packets directly to the control plane, the switch forwards the packets through the transport network; e.g. Ethernet, to the Classifier Model. Once a packet has been identified, it will be labelled with certain application/class and sent to the controller, which is made possible with the programmability property of the SDN switches. In the control plane, the controller performs resource and queue management according to the labelled packets and updates the flow tables via the switch.



*The whole TC engine will be placed in the data plane, where Feature selection and classifier training is done offline and regularly to update Classifier Model, which runs online.

Fig. 1. The proposed software framework of Naïve Bayes Traffic Classification in SDN-ISP

4 Conclusion

SDN architecture has a promising future for a guaranteed application-specific QoS. Unlike conventional best effort network services, SDN improves traffic engineering by taking over the control of the entire network through a centralized controller. However, to realize this, the controller needs a quick and accurate identification of the network traffic flows to enhance traffic routing, resource management and flow scheduling. Current practice in traffic classification requires packet inspection which is not feasible due to privacy issues as well as high power and memory consumption. On the other hand, statistical-based method, particularly ML-based can be trained to formulate accurate traffic classifiers by recognizing statistical patterns in the flows. This work has pointed out the gaps in the previous works on SDN traffic classification; i.e. efficient and scalable TC engine. Hence, an ML-based QoS-aware Internet traffic classification should be devised to equip service providers with enhanced traffic engineering. Realizing the issues, this work proposes a Naïve Bayes traffic classifier which is part of the TC engine placed at the data plane. The proposed solution will be the starting point in solving efficiency and scalable issue in SDN-ISP TC.

Acknowledgments. This research work is fully supported by the research grant of TM R&D and Multimedia. University, Cyberjaya, Malaysia. We are very thankful to the team of TM R&D and Multimedia University for providing the support to our research studies.

References

1. Nina, K., Anastasia, K.: Quality of services evaluation method in next generation networks. In: 2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET), Lviv-Slavske, pp. 1055–1058 (2018)
2. Budiman, E., Wicaksono, O.: Measuring quality of service for mobile internet services. In: 2016 2nd International Conference on Science in Information Technology (ICSITech), Balikpapan, pp. 300–305 (2016)
3. Yu, C., Lan, J., Guo, Z., Hu, Y., Baker, T.: An adaptive and lightweight update mechanism for SDN. IEEE Access **7**, 12914–12927 (2019)
4. Guerra Perez, K., Yang, X., Scott-Hayward, S., Sezer, S.: A configurable packet classification architecture for software-defined networking. In: 2014 27th IEEE International System-on-Chip Conference (SOCC) (2014)
5. Nguyen, T.T.T., Armitage, G.: A survey of techniques for internet traffic classification using machine learning. IEEE Commun. Surv. Tutorials **10**(4), 56–76 (2008)
6. Aujla, G.S., Chaudhary, R., Kumar, N., Kumar, R., Rodrigues, J.J.P.C.: An ensembled scheme for QoS-aware traffic flow management in software defined networks. In: 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, pp. 1–7 (2018)
7. Sood, K., Yu, S., Xiang, Y., Cheng, H.: A general QoS aware flow-balancing and resource management scheme in distributed software-defined networks. IEEE Access **4**, 7176–7185 (2016)
8. Vasiliadis, D., Rizos, G., Vassilakis, C.: Class-based weighted fair queuing scheduling on dual-priority delta networks. J. Comput. Netw. Commun. **27**(5), 435–457 (2012)
9. Tajiki, M.M., Akbari, B., Shojafar, M., Mokari, N.: Joint QoS and congestion control based on traffic prediction in SDN. Appl. Sci. **7**(12), 1265 (2017)
10. Madhukar, A., Williamson, C.: A longitudinal study of P2P traffic classification. In: Proceedings of the 14th IEEE International Symposium on the Modeling, Analysis, Simulation, pp. 179–188, September 2006
11. Jeong, S., Lee, D., Choi, J., Li, J., Hong, J.W.: Application-aware traffic management for OpenFlow networks. In: 2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS), Kanazawa, pp. 1–5 (2016)
12. Sanvito, D., Moro, D., Capone, A.: Towards traffic classification offloading to stateful SDN data planes. In: 2017 IEEE Conference on Network Softwarization (NetSoft), Bologna, pp. 1–4 (2017)
13. Lee, S., Park, J., Yoon, S., Kim, M.: High performance payload signature-based Internet traffic classification system. In: 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS), Busan, pp. 491–494 (2015)
14. Wang, P., Ye, F., Chen, X., Qian, Y.: Datanet: deep learning based encrypted network traffic classification in SDN home gateway. IEEE Access **6**, 55380–55391 (2018)
15. Dias, K.L., Pongelupe, M.A., Caminhas, W.M., de Errico, L.: An innovative approach for real-time network traffic classification. Comput. Netw. **158**, 143–157 (2019). https://doi.org/10.1016/j.comnet.2019.04.004
16. Xu, J., Wang, J., Qi, Q., Sun, H., He, B.: Deep neural networks for application awareness in SDN-based network. In: 2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP), Aalborg, pp. 1–6 (2018)
17. Ibrahim, H.A.H., Aqeel Al Zuobi, O.R., Al-Namari, M.A., MohamedAli, G., Abdalla, A.A.A.: Internet traffic classification using machine learning approach: datasets validation issues. In: 2016 Conference of Basic Sciences and Engineering Studies (SGCAC), Khartoum, pp. 158–166 (2016)

18. Qazi, Z.A., Lee, J., Jin, T., Bellala, G., Arndt, M., Noubir, G.: Application-awareness in SDN. In: Proceedings of the ACM SIGCOMM 2013, Hong Kong, China, pp. 487–488 (2013)
19. Amaral, P., Dinis, J., Pinto, P., Bernardo, L., Tavares, J., Mamede, H.S.: Machine learning in software defined networks: data collection and traffic classification. In: Proceedings of the IEEE ICNP 2016, Singapore, November 2016, pp. 1–5 (2016)
20. Uddin, M., Nadeem, T., TrafficVision: a case for pushing software defined networks to wireless edges. In: Proceedings of the IEEE MASS 2016, Brasilia, Brazil, October 2016, pp. 37–46 (2016)
21. Lashkari, A.H., Draper Gil, G., Mamun, M., Ghorbani, A.: Characterization of tor traffic using time based features. In: Proceedings of the 3rd International Conference on Information Systems Security and Privacy, ICISSP, vol. 1, pp. 253–262 (2017). ISBN 978-989-758-209-7
22. Fan, Z., Liu, R.: Investigation of machine learning based network traffic classification. In: 2017 International Symposium on Wireless Communication Systems (ISWCS), Bologna, pp. 1–6 (2017)
23. Yu, C., Lan, J., Xie, J., Hu, Y.: QoS-aware traffic classification architecture using machine learning and deep packet inspection in SDNs. Proc. Comput. Sci. **131**, 1209–1216 (2018)
24. Yu, C., Lan, J., Guo, Z., Hu, Y., Baker, T.: QoS-aware traffic mechanism for SDN. IEEE Access **7**, 12914–12927 (2019)
25. Wang, P., Lin, S.C., Luo, M.: A framework for QoS-aware traffic classification using semi-supervised machine learning in SDNs. In: Proceedings of the IEEE SCC 2016, San Francisco, CA, USA, June 2016, pp. 760–765 (2016)
26. Li, Y., Li, J.: MultiClassifier: a combination of DPI and ML for application-layer classification in SDN. In: The 2014 2nd International Conference on Systems and Informatics (ICSAI 2014), Shanghai, pp. 682–686 (2014)
27. Lashkari, A.H., Draper-Gil, G., Mamun, M.S.I., Ghorbani, A.A.: Characterization of encrypted and VPN traffic using time-related features. In: Proceedings of the International Conference on Information Systems Security and Privacy (ICISSP), pp. 407–414, February 2016
28. Hayes, M., Ng, B., Pekar, A., Seah, W.K.G.: Scalable architecture for SDN traffic classification. IEEE Syst. J. **12**(4), 3203–3214 (2018)
29. Tsamardinos, I., Borboudakis, G., Katsogridakis, P., et al.: A greedy feature selection algorithm for big data of high dimensionality. Mach. Learn. **108**, 149 (2019). https://doi.org/10.1007/s10994-018-5748-7