Society for Science and Education
United Kingdom

TNC Transactions on Networks and Communications

Volume 6, No. 2
ISSN: 2054 -7420

Applying Big Data, Machine Learning, and SDN/NFV for 5G Early-Stage Traffic Classification and Network QoS Control

Luong-Vy Le[1], Bao-Shuh Paul Lin[2,3], Do Sinh[2]

[1]*College of Electrical and Computer Engineering, National Chiao Tung University, Hsinchu, Taiwan*
[2]*Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan*
[3]*Microelectronics & Information Research Center, National Chiao Tung University, Hsinchu, Taiwan*
leluongvy.eed03g@nctu.edu.tw, bplin@mail.nctu.edu.tw, dosinhuda.cs04g@nctu.edu.tw

ABSTRACT

Due to the rapid growth of mobile broadband and IoT applications, the early-stage mobile traffic classification becomes more important for traffic engineering to guarantee Quality of Service (QoS), implement resource management, and network security. Therefore, identifying traffic flows based on a few packets during the early state has attracted attention in both academic and industrial fields. However, a powerful and flexible platform to handle millions of traffic flows is still challenging. This study aims to demonstrate how to integrate various state-of-the-art machine learning (ML) algorithms, big data analytics platforms, software-defined networking (SDN), and network functions virtualization (NFV) to build a comprehensive framework for developing future 5G SON applications. This platform successfully collected, stored, analyzed, and identified a huge number of real-time traffic flows at broadband Mobile Lab (BML), National Chiao Tung University (NCTU). Moreover, we also implemented network QoS control to configure priorities per-flow traffic to enable bandwidth guarantees for each application by using SDN. Finally, the performance of the proposed models was evaluated by applying them to a real testbed environment. The powerful computing capacity of the platform was also analyzed.

Keywords: Traffic classification; Machine Learning; Big Data; SON; 5G; InfoSphere; Streaming.

1 Introduction

In 5G networks and IoT contexts, small cells, heterogeneous networks (HetNets), wireless sensor networks (WSNs) are deployed everywhere to bring enhanced mobile broadband services to users, such as Internet of Everything (IoE) paradigms, cloud services, and real-time video streaming services. However, the diversity of broadband services carries many challenges for traffic engineering to provide a technical solution to improve the network QoS and QoE (quality of experience). For example, video streaming is a time-sensitive service, so any unexpected delay may result in bad QoE. Moreover, understanding traffic flow behaviors of running applications in the network play a critical role for network operators to implement QoS and QoE policies, network efficiency, resource management, load balancing, energy saving [1].

Recently, the early-state traffic classification has become an important topic in communication, and it was explored in many studies [2][3][4][5][6][7][8]. Generally, a traffic classification model can be divided into

DOI: 10.14738/tnc.62.4446
Publication Date: 1st May 2018
URL: http://dx.doi.org/10.14738/tnc.62.4446

two steps: Firstly, traffic is divided into flows, and their headers are extracted to define useful features for the classification model such as packet number, packet size. Secondly, the classification model trains and classifies traffic flows into different types of applications. Studies [7][8] introduced several popular approaches for the early-stage traffic classification: Port-based classification, payload-based classification, protocol behavior or heuristics based classification, statistical analysis based classification, deep packet inspection (DPI) approaches, and packet size approaches. The port-based approach is simple and easy to implement, however, because nowadays the number of applications using dynamic port is increasing, this model becomes inaccurate. On the other hand, the payload-based approach investigates packet payload to determine the signatures of known applications; therefore, it can only classify traffic flows for which signatures are available, and it usually requires substantial computing power and storage capacity, besides, this method also violates the privacy laws. Another example, the DPI approach, which aims at identifying traffic protocol patterns in the packets of different applications, is too complicated with high computational overheads. Fortunately, the most current studies, such as [6][5][9][10], concluded that traffic classification models based on the packet sizes of some first packets were powerful enough for achieving high classification performance. Furthermore, research [6] found that the most efficient number of packets used for traffic flows identification is from 5 to 7 packets. That means too many packets and too few packets may reduce the model efficiency. In addition, ML algorithms play a significant role in deciding the success of the model. Those studies focused on applying powerful and well-known ML algorithms as classifiers, such as Naïve Bayes, support vector machine (SVM), Random Forest, logistic, Hidden Markov Model (HMM) etc. For example, research [6] investigated the performance of 11 well-known ML models. Especially, In the study [2], we proposed a traffic classification model using HMM to classify the internet traffic of mobile broadband applications based on the packet sizes and packet transmission directions. As a result, it achieved 99.17% accuracy for 6 types of mobile applications.

One of the most essential application of the early-state traffic flow classification is network QoS control. In research [1], the authors investigated and proposed a systematic design approach to support QoS-guaranteed chaining services with considering the effects of both data plane and control plan messages. The delay of the services and SDN were evaluated

This study focuses on enhancing the current architecture to propose a comprehensive framework of integrating big data, ML SDN/NFV, and cloud for collecting, transforming, extracting, and analyzing mobile traffic flows and then building powerful mobile traffic classification at the early stage. Furthermore, a new and effective network QoS control based on Open Flow Switch is presented. The experiments are deployed on the experimental 4G/LTE & beyond 4G network testbed, located at MIRC/BML (Microelectronics and Information Research Center/Broadband Mobile Lab) in the campus of National Chiao Tung University).

The remainder of the paper is organized as follows: Section 2 introduces the experimental architecture based on SDN/NFV, big data, and ML; Section 3 demonstrates Open-SON platform based on big data, ML, and SDN/NFV; Section 4 implements, evaluates, and analyzes the early-state traffic flow classification application; Section 5 proposes and implement per-flow traffic QoS control based on SDN, Section 6 concludes the present study.

2 Experimental Architecture Based on Big Data, ML and SDN/NFV at BML

Recently, ML, big data, cloud, and SDN/NFV have been applying in developing 5G networks to support computing, communication, programming abstractions, data analysis, and management services. Those technologies have been applied to the experimental 4G/LTE&5G network testbed, located at MIRC/BML. This platform integrates several big data and ML environments (e.g. Apache Spark, IBM InfoSphere) that works in collaboration with the advanced SDN/NFV technologies (e.g. ONOS, OpenDaylight, P4, Docker) to design various 5G applications. Based on the platform, many applications were introduced. For example, research [11] investigated the roles of 5G in the development of cloud, big data, SDN, and IoT, and then it proposed an integrated architecture of these technologies for 5G; research [12] addressed the technology outlook of computing power and system characteristics of 5G Mobile broadband system; research [13] described and implemented cloud computing for various Mobile Augmented Reality (MAR) applications with smart mobile devices; research [14] showed the challenges of applying SDN/NFV to 5G and IoT. [15][16] investigated various ML algorithms to cluster, forecast, and manage handover behaviors and traffic behaviors of a huge number of cells based on analyzing a huge amount of collected data. This study proposes a comprehensive architecture for traffic flow classification, the architecture and real devices is described as Fig.1.

2.1 Physical Devices

- ✓ UEs: some modern UEs such as iPhone 5s, iPhone 6s, Zenphone 3 are used to open broadband services
- ✓ eNodeB: both indoor and outdoor RRUs are used to connect with Nokia BBUs located inside BML lab. Furthermore, we can implement experiment with multi-vendors (NSN (Nokia), Huawei) and multi-modes (TDD and FDD).
- ✓ EPC: (Evolve packet core): The BBUs are connected to 2 cores through a switch, the main core is located at ITRI and the open core (open5G core) is located in BML for developing 5G applications.
- ✓ SDN/NFV environment: SDN and NFV are recognized as key technologies studied for enhancing mobile network applications by providing the capacity of programmability for control and data plane of both RAN and EPC’s elements.

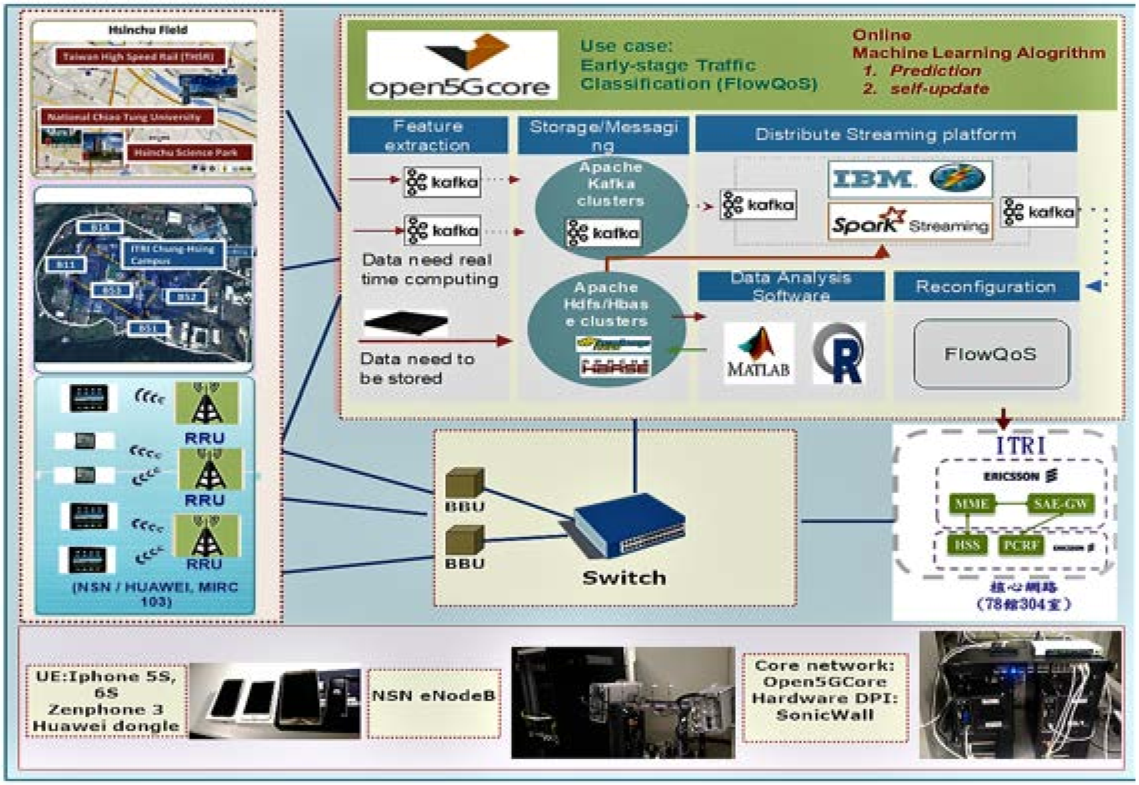


Figure 1. Experimental architecture and devices of BML

2.2 Experimental Process

Figure 1. demonstrates the process of data collection, data transformation, data storage, data analysis, visualization, and ML applications for both statistical models and online/streaming models in this study. Generally, the common process for applying ML and data mining to mobile networks usually involves 4 steps:

Step 1: Collect and storage mobile traffic and other data that necessary for the application from all sources of the network.

Step 2: Using data mining and machine learning to develop optimization models by extracting and analyzing the collected data at the open5Gcore

Step 3: Apply the models and optimization parameters to network components such as the SON in the main EPC, SDN controller, and eNodeB

Step 4: Analyze and evaluate the network performance through network KPIs (Key performance indicators) to determine whether the optimization model meets the expected results. If the network behavior achieves the expected performance, the new network parameters (NPs) will be applied. Otherwise, we need to identify problems such as change machine learning models or learning parameters.

3 Big Data and Machine Learning Platform for Empowering 5G SON

The SON of 5G consists of three main functions: self-configuration, self-optimization, and self-healing. Self-configuration provides plug-and-play functionality for both eNodeBs and EPC, for instance, when a new network element is added to the RAN such as a BBU or RRU, it will automatically download the necessary software as well as configure basic network parameters, such as the neighbor list, the radio parameters, etc. Self-Optimization continuously optimizes and controls the network parameters to respond the real-time network states to ensure that the network is working efficiently at the peak performance. For example, load-balancing algorithms are used to optimize RAN traffic, energy-saving algorithms turn RAN elements on or off due to traffic load [15]. The self-healing responds to a failure or a malfunction in the network with two steps: The cell outage detection and the cell outage compensation. It involves remote diagnosis, abnormal detection, failure prediction, and compensation to keep the network operating smoothly by detecting problems, and then automatically changes the network parameters of the active elements. Therefore, it requires a powerful, cost-effective, and autonomous SON in 5G with full intelligence to meet the requirement of both users and operators.

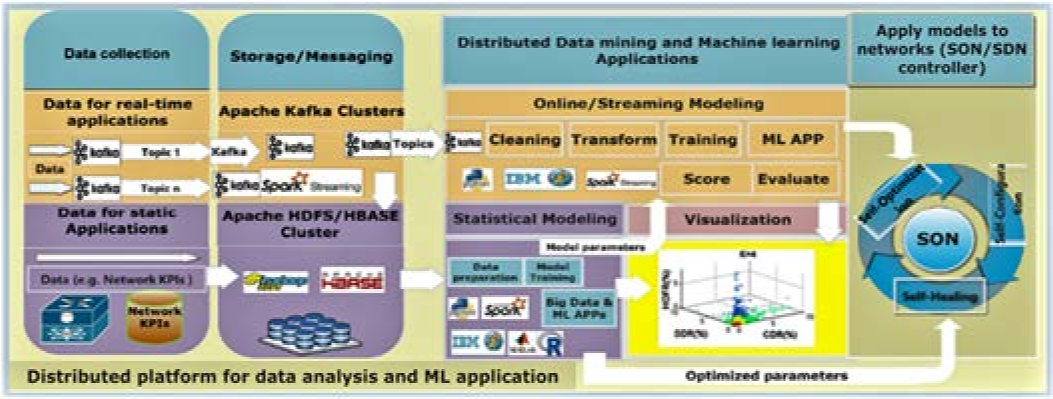

Data collection
Storage/Messaging
Distributed Data mining and Machine learning Applications
Apply models to networks (SON/SDN controller)
Data for real-time applications
Apache Kafka Clusters
Online/Streaming Modeling
Topic 1
Kafka
Topics
Cleaning
Transform
Training
ML APP
Data
Topic n
Score
Evaluate
Data for static Applications
Apache HDFS/HBASE Cluster
Statistical Modeling
Visualization
Model parameters
Data (e.g. Network KPIs)
Data preparation
Model Training
Network KPIs
Big Data & ML APPs
SON
Self-Healing
Distributed platform for data analysis and ML application
Optimized parameters

Figure 2 Open-SON for 5G platform

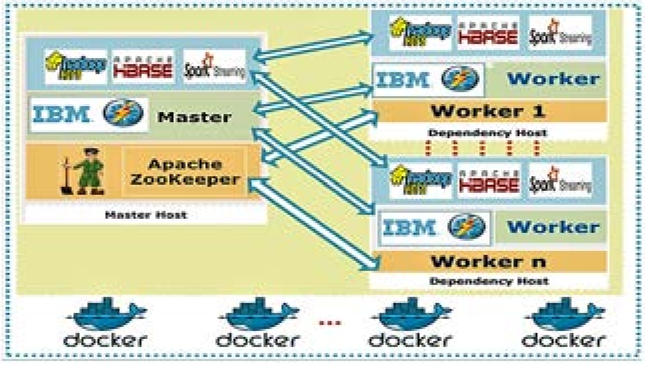


Figure 3. Distributed computing platform

3.1 Open Platform for 5G SON

Virtual SON (VSON), which integrates lately developed technologies such as software-defined wireless networking (SDWN), NFV, big data, ML, has been recently introduced as a prime proposal for future SON [17]. In this subsection, we propose and analyze an Open-SON for 5G in which we can apply data analytics and ML algorithms to develop variety SON applications. The big data and ML framework for the Open-SON shown in Fig. 2 composes four components: data collection; data storage; data analytics and ML applications; network configuration and optimization.

The platform is open to different types of technologies that can be easily integrated and deployed to build both online and offline applications. For example, Kafka, Flute, and Python are used to collect data from different data resources; programming languages such as R, Matlab, Spark, and InfoSphere can be used for analyzing data and building various ML algorithms. Especially, these software platforms are compatible with one another, for example, Spark can support various languages: R, Python, Java, Scala. Moreover, to satisfy the flexibility and scalability requirements for 5G applications, the Open-SON must work in distributed computing system in which a host works as the master and multiple hosts work as workers or executors. Fig 3 shows that all components in the master and workers are deployed in Docker containers, this helps the deployment of computing application becomes easier, quicker, and more efficient. In this framework, ZooKeeper manages and controls all workers so that a distributed application can run on multiple executors in a cluster simultaneously. The executors coordinate among themselves to handle a specific task in a fast and efficient way. In other words, the software components such as Kafka, HBase, Spark, InfoSphere run concurrently and independently on multiple physical machines. This makes the system becomes more powerful fully with intelligence and automation. Finally, the distributed computing system are deployed and controlled based on SDN/NFV environments.

3.2 Machine Learning Algorithms for Empowering SON

Fig. 4 summarizes ML algorithms that are used to reinforce the Open-SON talent in building various applications such as clustering, classification, prediction, and forecasting.

Clustering applications usually utilize unsupervised algorithms such as K-means and Mixtures of Gaussians to group and identify a set of similar network parameters together, such as clustering network parameters (NPs), coverage area of a cell, traffic density, data compression, and abnormal detections. Especially, in research [15], we used K-means to cluster the handover behaviors of 2000 cells and then extracted the handover characteristics of each cluster and all cells in a cluster.

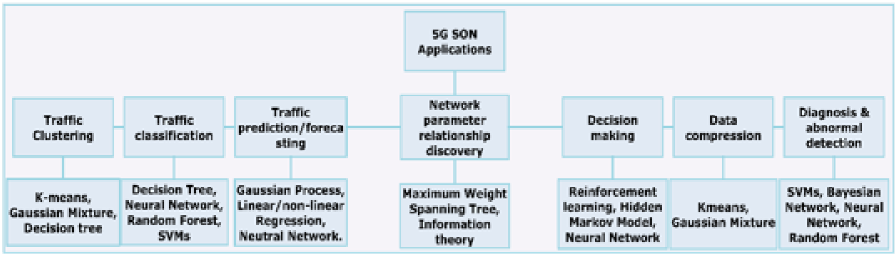

5G SON Applications
Traffic Clustering
Traffic classification
Traffic prediction/forecasting
Network parameter relationship discovery
Decision making
Data compression
Diagnosis & abnormal detection
K-means, Gaussian Mixture, Decision tree
Decision Tree, Neural Network, Random Forest, SVMs
Gaussian Process, Linear/non-linear Regression, Neutral Network.
Maximum Weight Spanning Tree, Information theory
Reinforcement learning, Hidden Markov Model, Neural Network
Kmeans, Gaussian Mixture
SVMs, Bayesian Network, Neural Network, Random Forest

Figure 4. SON applications and ML algorithms

Forecasting and prediction applications aim to identify and predict precise trends of the network parameters, network events while operating. This helps the SON keeps track of network parameters and deploys further optimization applications. For example, in studies [15] and [16], we proposed several models to accurately and efficiently forecast future handover numbers and the traffic of a huge number of cells by using several ML algorithms: Neural network (NN), Gaussian process (GP), and linear regression. Other examples of typical prediction and forecasting applications that can be applied to 5G networks are subscriber tracking, HO trend prediction, power control, antenna adjustment (e.g., tilt, azimuth, transmitted power), and load balancing. Typical dynamic ML algorithms can be used for those applications are Kalman Filter, Random Forest, HMM, NN, Linear Dynamical Systems, and GP.

Classification applications can use both unsupervised and supervised algorithms to classify network parameters into the relevant groups based on some significant features. For example, the following section will classify mobile traffic applications using several popular ML algorithms such as Random Forest, Decision Trees, support vector machine (SVM), etc.

4 Early-State Traffic Flow Classification

The process of the early-state traffic flow classification, which implemented on the platform of BML, can be divided into 2 steps: feature extraction and classification as shown in Fig. 5. The first step extracts and defines useful features for the classification model from the collected traffic flows of each application. It involves several pre-processing data processes, such as feature collection, data cleaning, and data transforming, to create a training dataset for the classifier. The second step builds classification models using different ML algorithms in InfoSphere, Spark, R, Matlab, etc. environments to implement the classification task. Finally, the classifiers will be applied to the SON and then new coming traffic flows will be classified for further processes such as network QoS control for each application.

4.1 Feature Extraction

Feature extraction and selection are essential steps deciding the accuracy and efficiency of classification models; therefore, the observed features must accurately represent the traffic behaviors of each application. As discussed in the introduction section, many studies such as [7] [9] proved that packet size and the number of packets carry enough information for the early-stage traffic classification. Generally, mobile applications are based on SSL/TLS built on the top of the TCP/IP protocol for security purpose [17].

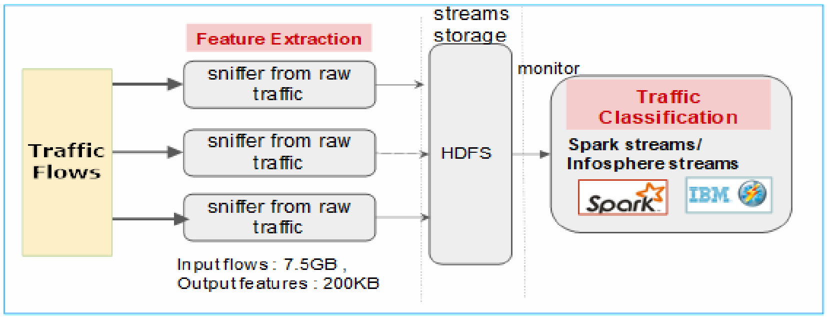
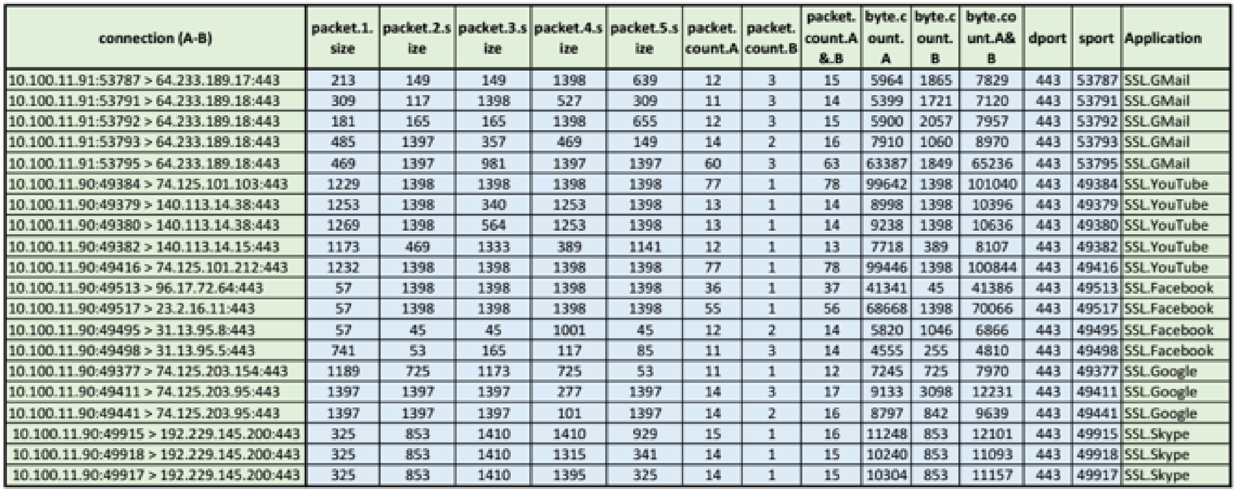


Figure 5. Traffic flow classification process

Table 1. Feature extraction of several traffic flows



connection (A-B)	packet.1.size	packet.2.size	packet.3.size	packet.4.size	packet.5.size	packet.count.A	packet.count.B	packet.count.A&.B	byte.count.A	byte.count.B	byte.count.A&B	dport	sport	Application
10.100.11.91:53787 > 64.233.189.17:443	213	149	149	1398	639	12	3	15	5964	1865	7829	443	53787	SSL.GMail
10.100.11.91:53791 > 64.233.189.18:443	309	117	1398	527	309	11	3	14	5399	1721	7120	443	53791	SSL.GMail
10.100.11.91:53792 > 64.233.189.18:443	181	165	165	1398	655	12	3	15	5900	2057	7957	443	53792	SSL.GMail
10.100.11.91:53793 > 64.233.189.18:443	485	1397	357	469	149	14	2	16	7910	1060	8970	443	53793	SSL.GMail
10.100.11.91:53795 > 64.233.189.18:443	469	1397	981	1397	1397	60	3	63	63387	1849	65236	443	53795	SSL.GMail
10.100.11.90:49384 > 74.125.101.103:443	1229	1398	1398	1398	1398	77	1	78	99642	1398	101040	443	49384	SSL.YouTube
10.100.11.90:49379 > 140.113.14.38:443	1253	1398	340	1253	1398	13	1	14	8998	1398	10396	443	49379	SSL.YouTube
10.100.11.90:49380 > 140.113.14.38:443	1269	1398	564	1253	1398	13	1	14	9238	1398	10636	443	49380	SSL.YouTube
10.100.11.90:49382 > 140.113.14.15:443	1173	469	1333	389	1141	12	1	13	7718	389	8107	443	49382	SSL.YouTube
10.100.11.90:49416 > 74.125.101.212:443	1232	1398	1398	1398	1398	77	1	78	99446	1398	100844	443	49416	SSL.YouTube
10.100.11.90:49513 > 96.17.72.64:443	57	1398	1398	1398	1398	36	1	37	41341	45	41386	443	49513	SSL.Facebook
10.100.11.90:49517 > 23.2.16.11:443	57	1398	1398	1398	1398	55	1	56	68668	1398	70066	443	49517	SSL.Facebook
10.100.11.90:49495 > 31.13.95.8:443	57	45	45	1001	45	12	2	14	5820	1046	6866	443	49495	SSL.Facebook
10.100.11.90:49498 > 31.13.95.5:443	741	53	165	117	85	11	3	14	4555	255	4810	443	49498	SSL.Facebook
10.100.11.90:49377 > 74.125.203.154:443	1189	725	1173	725	53	11	1	12	7245	725	7970	443	49377	SSL.Google
10.100.11.90:49411 > 74.125.203.95:443	1397	1397	1397	277	1397	14	3	17	9133	3098	12231	443	49411	SSL.Google
10.100.11.90:49441 > 74.125.203.95:443	1397	1397	1397	101	1397	14	2	16	8797	842	9639	443	49441	SSL.Google
10.100.11.90:49915 > 192.229.145.200:443	325	853	1410	1410	929	15	1	16	11248	853	12101	443	49915	SSL.Skype
10.100.11.90:49918 > 192.229.145.200:443	325	853	1410	1315	341	14	1	15	10240	853	11093	443	49918	SSL.Skype
10.100.11.90:49917 > 192.229.145.200:443	325	853	1410	1395	325	14	1	15	10304	853	11157	443	49917	SSL.Skype

When an application session starts, there are several negotiation stages between the client side and the server side. For example, each TCP flow begins with the TCP three-way handshake. This process consists of several continuous interaction rounds, which contain one or multiple messages (layer 7 messages). Those messages are segmented, in other words, the TCP layer receives encrypted data from the above layer and adds a TCP header to create TCP segments. After that, each TCP segment is encapsulated into IP packets and exchanged with a peer. In TCP connection, TCP packets do not include a session identifier so that both endpoints identify the TCP session through the client's IP address and the port number. Here, the system captures incoming IP packets and parses traffic flows to extract the headers of IP packets. A traffic flow is defined as a bi-directional ordered sequence of packets, which consists of the same 5-tuple: source IP, destination IP, source port, destination port, and transport layer protocol. In this study, the features are extracted from the application layer and transport layer perspectives [5]. They consist of the number of the packets and the packet sizes of the first interaction round, the size of the first 5 packets of each TCP/UDP flow, the source port and the destination port of the connection. Table 1 is an example of some samples of several flows, each input data sample contains 13 features.

4.2 Machine Learning Algorithms for traffic flow classification

This section briefly introduces characteristics of several state-of-the-art ML algorithms that are used in this study

Naïve Bayes is a straightforward and powerful probabilistic classifier, which computes the probability of a data sample that belongs to each class by using Bayes' theorem. It assumes that all features are conditional independence with one another. That means the presence of one feature does not affect the presence of others. As a result, this model is easy to train and can provide impressive performance, even if it is working on a data set with millions of data samples.

Gradient Boosted Tree (GBT): GBT, an ensemble of decision trees, combines simple parameterized functions to achieve high accuracy for prediction and classification models. In other words, it iteratively trains multiple decision trees to minimize the cost function and provide more accurate prediction model by changing complex interactions in a simple fashion.

Random Forest (RF): RF is also known as an ensemble of decision trees computed in parallel fashion on a dataset by using random subsets to improve the multiclass classification rate and to overcome the over-fitting problem. To classify a data sample, different trees in the forest will learn on different subsets of data, and then they make classification on their own. Finally, the final class of the testing data sample is assigned to a class that has majority votes.

SVM is a popular supervised ML algorithm for pattern recognition, such as classification, regression, and abnormal detection. It analyzes training data to find the largest margin for linear and non-linear classifiers. While many ML algorithms are memory-based methods, that means the kernel function, which implicitly maps their inputs into high-dimensional feature space, must be calculated for all pairs of training points. As a result, it needs a huge amount of calculation during the training stage so that they easily lead to excessive computation and computing time, SVM, in another way, is a decision algorithm that classifies a new sample only depend on calculating a subset of the training data.

Neural network (NN): NN is a non-linear algorithm in which the computational scheme is based on the structure and functions of biological neural networks. It represents for popular technologies to provide the best solution for many problems in which relationships between multiple input and output variables are complex. Recently, NN has been applied in many fields, including pattern recognition, speech recognition, image recognition, and natural language processing.

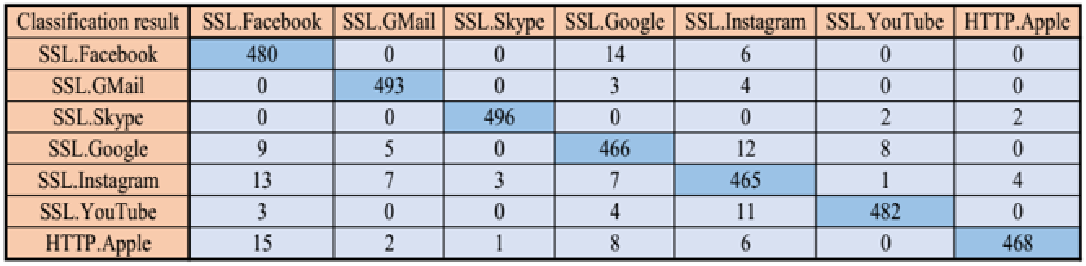
4.3 Experimental implementation

This subsection evaluates and compares the classification performance of several state-of-the-art ML algorithms to find out relevant algorithms for traffic classifications. Generally, classification models are form of supervised learnings, which include two phases, training phase and testing phase. For each experiment, a training dataset of 21000 traffic flow samples of 7 applications, 3000 flows for each application, was chosen randomly from the collected data from different connections. The input of each data sample consists of 13 features and the output is an application label as described in Table 1. Moreover, since the common values of the features are in different ranges, they need to be normalized into relevant ranges by applying a z-score normalization on all data columns. Finally, to evaluate the performance of each ML algorithm, a testing dataset of 3500 testing flow samples (500 flows for each application) was chosen randomly, and it must differ from the training dataset.

4.4 Experimental result

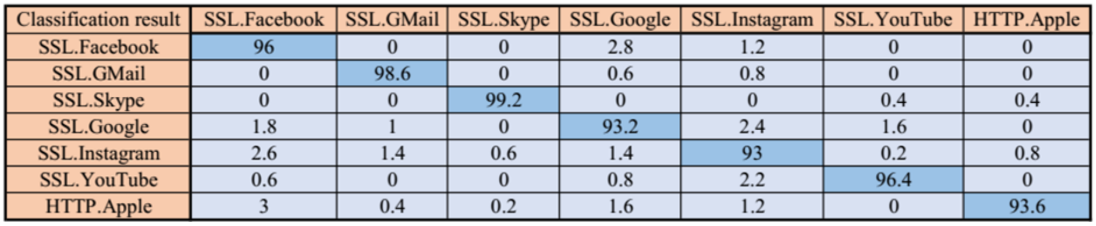
Firstly, we analyze the model performance through the confusion matrix of classification results. Table 2 & 3 show the classification result of SVM algorithm representing by number and percentages, respectively. For example, the number of 500 Facebook flows that are classified as Facebook, Gmail, Skype, Google, Instagram, YouTube, and Apple are 480, 0, 0, 14, 6, 0, 0, respectively. In other words, 96% Facebook flows were identified accurately, and the remainder were identified inaccurately, Google (2.8%) and Instagram (1.2%). That means some of Facebook flows have quite similar characteristics to those of Google and Instagram.

Table 2. Confusion matrix (number) of SVM Classification result



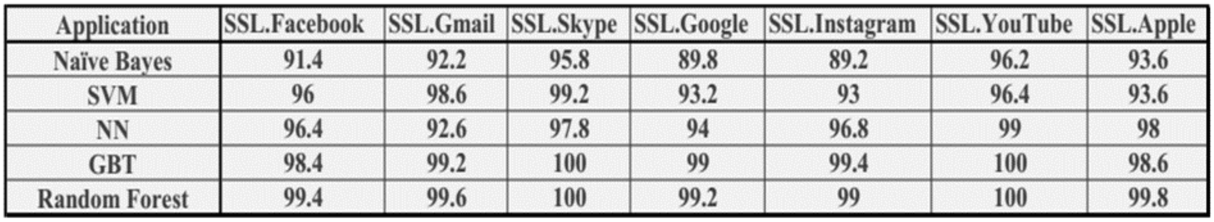
Classification result	SSL.Facebook	SSL.GMail	SSL.Skype	SSL.Google	SSL.Instagram	SSL.YouTube	HTTP.Apple
SSL.Facebook	480	0	0	14	6	0	0
SSL.GMail	0	493	0	3	4	0	0
SSL.Skype	0	0	496	0	0	2	2
SSL.Google	9	5	0	466	12	8	0
SSL.Instagram	13	7	3	7	465	1	4
SSL.YouTube	3	0	0	4	11	482	0
HTTP.Apple	15	2	1	8	6	0	468

Table 3. Confusion matrix (percentage %) of SVM Classification result



Classification result	SSL.Facebook	SSL.GMail	SSL.Skype	SSL.Google	SSL.Instagram	SSL.YouTube	HTTP.Apple
SSL.Facebook	96	0	0	2.8	1.2	0	0
SSL.GMail	0	98.6	0	0.6	0.8	0	0
SSL.Skype	0	0	99.2	0	0	0.4	0.4
SSL.Google	1.8	1	0	93.2	2.4	1.6	0
SSL.Instagram	2.6	1.4	0.6	1.4	93	0.2	0.8
SSL.YouTube	0.6	0	0	0.8	2.2	96.4	0
HTTP.Apple	3	0.4	0.2	1.6	1.2	0	93.6

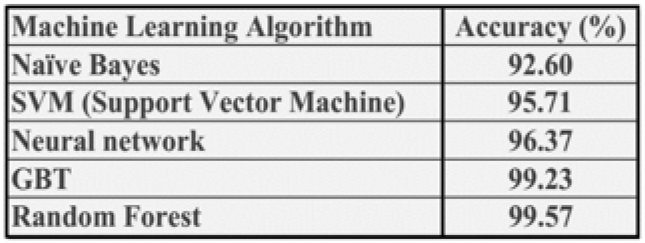
Table 4. Classification result of different applications and ML algorithms



Application	SSL.Facebook	SSL.Gmail	SSL.Skype	SSL.Google	SSL.Instagram	SSL.YouTube	SSL.Apple
Naïve Bayes	91.4	92.2	95.8	89.8	89.2	96.2	93.6
SVM	96	98.6	99.2	93.2	93	96.4	93.6
NN	96.4	92.6	97.8	94	96.8	99	98
GBT	98.4	99.2	100	99	99.4	100	98.6
Random Forest	99.4	99.6	100	99.2	99	100	99.8

Table 4 summarizes and compares the classification performance of different ML algorithms for different applications. It is noticeable that all the algorithms can achieve high accuracy for identifying those applications, however, different models give different performances for different applications. In general, Table 5 summarizes the performance of each ML algorithm, as can be seen, Naïve Bayes gets worse performance, SVM and Neural Network give a quite similar performance, and they are better than Naïve Bayes, while GBT and RF give the best performance for all traffic flows.

Table 5. Average classification performance of ML algorithms



Machine Learning Algorithm	Accuracy (%)
Naïve Bayes	92.60
SVM (Support Vector Machine)	95.71
Neural network	96.37
GBT	99.23
Random Forest	99.57

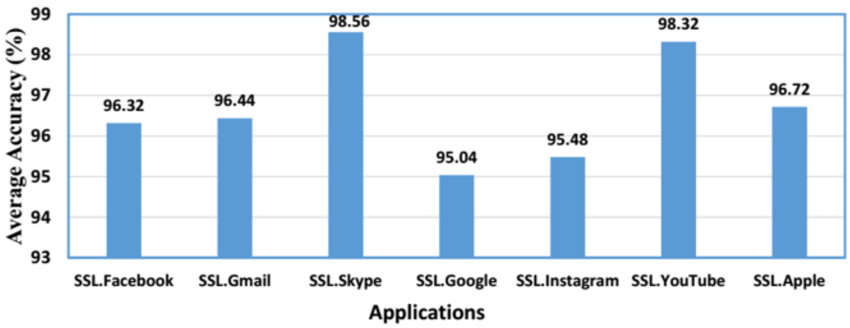
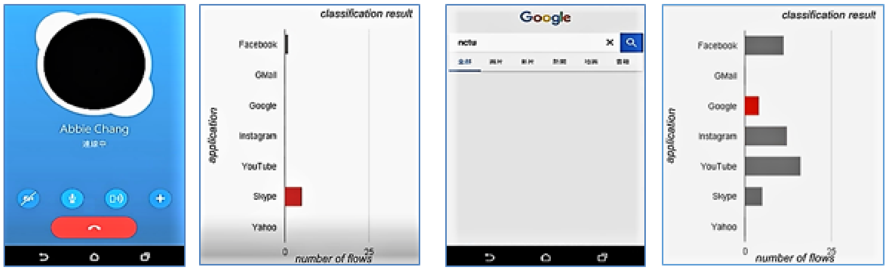


Figure 6. Average classification accuracy of applications



(a) Opening Skype (b) Identifying Skype flows (c) Opening Google (d) Identifying Google flows

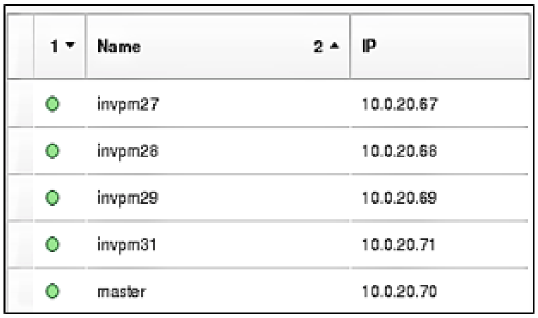
Figure 7. Online classification results

Moreover, during analyze the results, we noticed that some applications such as Google and Instagram are more difficult to identify than for others due to the fact that some of their flow features (e.g. the first packet size) vary significantly among flows of different connections and different user's actions on UEs (e.g. send an email, open an email), besides, they are also easy to confuse with those of other traffic flows. For more details, Fig.6 shows the average classification accuracy of each type of traffic flows, it is clear that YouTube and Skype are easier to identify and get the highest classification accuracies.

Online classification: this experiment uses SVM algorithm to identify online traffic flows of mobile applications. A Streaming application is usually described as a directed graph composing individual computing entities that interconnect and operate on a platform; therefore, it often integrates a monitoring application for scheduling and managing purposes. In this test, a UE was used to access mobile applications, and their traffic flows were classified into relevant applications. Moreover, with each application, we played different actions that may be sensitive the traffic flow behavior. For example, we analyzed typical action on Gmail such as send an email, send a reply, open an email, open chats, etc. The number of traffic flows of each application was also accumulated. Fig. 7 (a) illustrates that Skype application is opening on a UE, 7(b) shows the identification result and the flow accumulation of each application. Similarly, Fig. 7(c)&(d) show the result when the user is opening Google application. In summary, the classification model is able to classify online or streaming traffic flows with high accuracies.

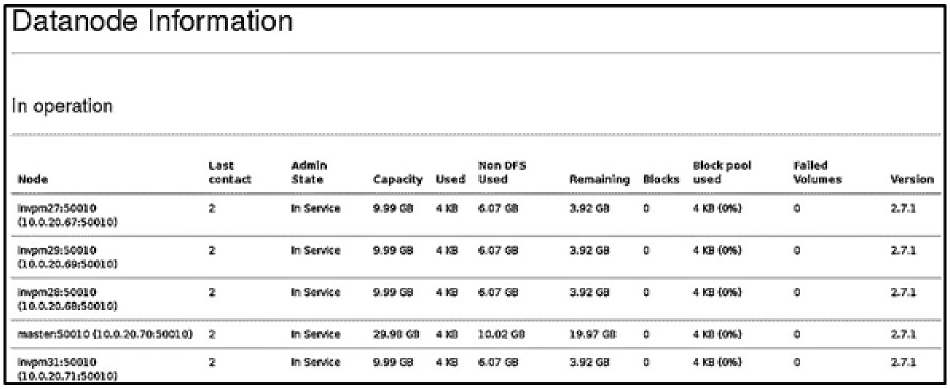
4.5 Computing Performance of InfoSphere Cluster

This subsection evaluates the computing performance of classification system. An online classification model was deployed in the InfoSphere cluster [18], which consists of one computing master and 4 slaves, their names and IP address are shown in Fig.8. In this experiment, we randomly generated a stream of 10.000.000 data samples as classification testing dataset, then Kafka received the data, created topics, and produced data to InfoSphere, after that, InfoSphere classified the incoming flows by its SVM classifier. Fig. 9 shows the computing state of the InfoSphere cluster in which computing job is equally distributed for all the slaves under the control of the master. Fig. 10 summarizes the computing performance, it shows the input flow rate, output flow rate, and the latency. As can be seen, with different the input speed, the cluster classified all the data samples smoothly with a small time for scheduling and processing, the maximum speed is around 90.000 flows/second with low total latency (average about 10ms).



1 ▾	Name 2 ▴	IP
○	invpm27	10.0.20.67
○	invpm28	10.0.20.68
○	invpm29	10.0.20.69
○	invpm31	10.0.20.71
○	master	10.0.20.70

Figure 8. InfoSphere computing cluster



Datanode Information

In operation

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
invpm27:50010 (10.0.20.67:50010)	2	In Service	9.99 GB	4 KB	6.07 GB	3.92 GB	0	4 KB (0%)	0	2.7.1
invpm29:50010 (10.0.20.69:50010)	2	In Service	9.99 GB	4 KB	6.07 GB	3.92 GB	0	4 KB (0%)	0	2.7.1
invpm28:50010 (10.0.20.68:50010)	2	In Service	9.99 GB	4 KB	6.07 GB	3.92 GB	0	4 KB (0%)	0	2.7.1
master:50010 (10.0.20.70:50010)	2	In Service	29.98 GB	4 KB	10.02 GB	19.97 GB	0	4 KB (0%)	0	2.7.1
invpm31:50010 (10.0.20.71:50010)	2	In Service	9.99 GB	4 KB	6.07 GB	3.92 GB	0	4 KB (0%)	0	2.7.1

Figure 9. Computing state of InfoSphere cluster

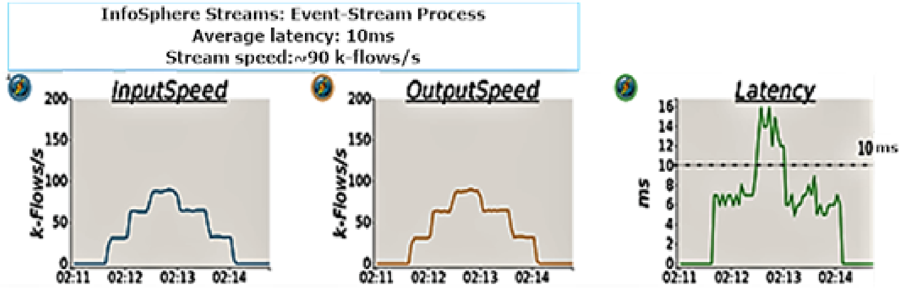


Figure 10. Computing speed and latency

In summary, the platform is powerful in supporting high computing capacity with low-latency. Moreover, it also provides an environment for collecting, transforming, and processing data of multiple streams from in inside and outside of the system. Therefore, it is relevant and powerful enough to be applied for the industrial case.

5 Network QoS Control for traffic applications Based on SDN

QoS control is a significant concept in mobile networks to prevent one mobile broadband application from degrading overall performance when it shares bandwidth with others. Therefore, how to manage QoS for multiple

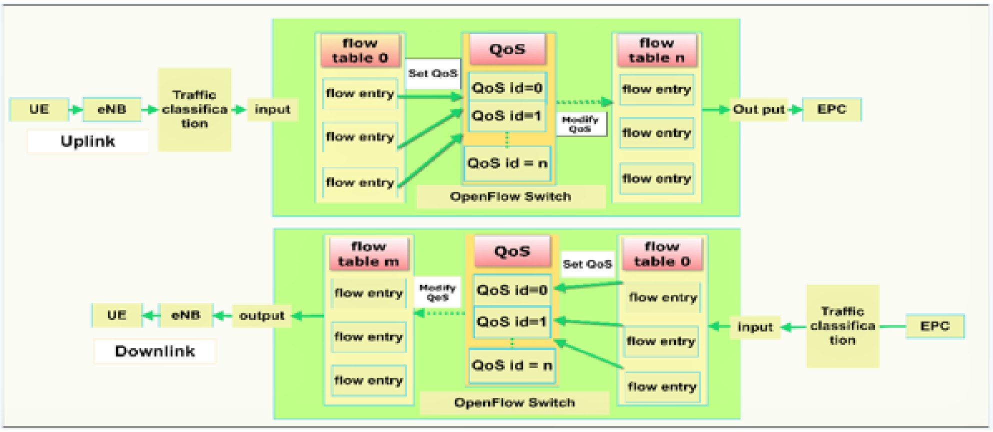
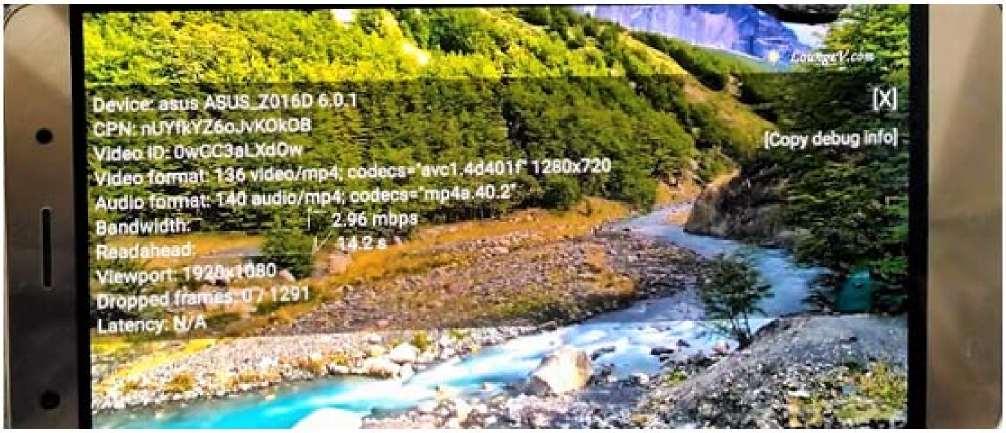


Figure 11. QoS control for traffic flows based on traffic classification

traffic flows is important to network operators in improving both user experience and network efficiency. Each application requires a QoS guarantee, for example, Skype and video streaming require small delay and jitter, while other data communications like Gmail, Google are more sensitive to packet loss. Hence, the network operator must preserve a relevant bandwidth for each application. The traditional technologies for network QoS control are Integrated Service (IntServ) and Differentiated Service (DiffServ). However, the former is too complex and not scalable, On the other hand, the latter is less complex, but does not provide strong QoS guarantees. Fortunately, SDN/NFV are emerging technologies that play important roles in enabling new approaches for QoS control such as the utilization of meter tables, ingress policing, Queue management, Virtual Network Embedding, etc. [1][19][20]. This section integrates the traffic flow identification system with a QoS management system based on SDN as described in Fig. 11. After a traffic flow is identified, it will be assigned a relevant bandwidth value by using one of above technologies. Among those methods, using meter tables and QoS queue in the flow table entries of OpenFlow Switch are the most relevant and effective approaches for controlling the QoS of traffic flows. The SDN controller utilizes the result of the classification application to setups flow table entries for each connection to instruct how the flows (packets) are executed. For example, in the meter table case, it uses meters, which are attached directly to flow entries, to measure and control the data rate of packets. Each meter table consists of multiple meter entries, which can support for different applications with different QoS policies. On other hand, the QoS queue method assigns a QoS ID for each application in the flow table entries. Fig. 11 describes the abstract process of implementing network QoS control in the mobile network for uplink and downlink directions. In this framework, the classification model is deployed close to Base Stations (edge network), this is important to support a variety of innovative applications and services quickly with very low latency like mobile edge computing (MEC) and Fog computing. Moreover, once the controller receives a traffic flow of an identified application, it will store the basic information of the connection, such as source IP address, destination IP address, application type, etc. Finally, based on these information, the SDN controller installs flow table entries and QoS policies to control the flows for both uplink and downlink directions. For example, in our case study, we set a bandwidth 3Mbps for YouTube application, then a UE was used to open YouTube video.

The result in Fig. 12 shows that the bandwidth provided for YouTube is 2.96 Mbps (around 3 Mbps).



Device: asus ASUS_Z016D 6.0.1
CPN: nUYfkYZ6oJvKOkOB
Video ID: 0wGC3aLXdOw
Video format: 136 video/mp4; codecs="avc1.4d401f" 1280x720
Audio format: 140 audio/mp4; codecs="mp4a.40.2"
Bandwidth: 2.96 mbps
Readahead: 14.2 s
Viewport: 1920x1080
Dropped frames: 0 / 1291
Latency: N/A

Figure 12. QoS control result for YouTube

6 Conclusion

This study proposed a comprehensive platform based on big data, ML, and SDN/NFV to empower the SON of 5G. Moreover, the process of building SON applications, such as data collection, storage, analytics, and virtualization were also introduced . Specifically, in the case study, we applied various state-of-the-art ML algorithms to classify accurately mobile applications at an early stage, then traffic flows of each application were preserved a relevant bandwidth controlled by the network QoS control by using SDN controller. This is crucial to the SON in ensuring that an application can work at right function, the network resources are utilized effectively, and user experience is guaranteed. Especially, both offline and online learning models were considered and implemented successfully. In the future, the authors focus on utilizing the results of the study to develop a comprehensive architecture for 5G SON and 5G MEC based on P4, ONOS, and CORD (Central Office Re-architected as a Datacenter) platforms, which are considered as the key elements and solutions of SDN/NFV technologies for 5G.

ACKNOWLEDGMENT

This paper is particularly supported by "Aiming for the SPROUT Project - Center for Open Intelligent Connectivity" of National Chiao Tung University and Ministry of Education, Taiwan, R.O.C. and the Ministry of Science and Technology of Taiwan under Grants: MOST 106-2221-E-009-008

REFERENCES

[1] Y. J. Chen, L. C. Wang, F. Y. Lin, and B. S. Lin, “Deterministic Quality of Service Guarantee for Dynamic Service Chaining in Software Defined Networking,” *IEEE Trans. Netw. Serv. Manag.*, vol. 14, no. 4, pp. 991–1002, 2017.

[2] I. C. Hsieh, L. P. Tung, and B. S. P. Lin, “On the classification of mobile broadband applications,” *IEEE Int. Work. Comput. Aided Model. Des. Commun. Links Networks, CAMAD*, pp. 128–134, 2016.

[3] J. Zhang, C. Chen, Y. Xiang, W. Zhou, and Y. Xiang, “Internet traffic classification by aggregating correlated naive bayes predictions,” *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 1, pp. 5–15, 2013.

[4] W. Ke, Y. Wang, X. Lei, and B. Wei, “Spark-Based Feature Selection Algorithm of Network Traffic Classification,” *2017 13th Int. Conf. Comput. Intell. Secur.*, pp. 140–144, 2017.

[5] N. F. Huang, G. Y. Jai, H. C. Chao, Y. J. Tzang, and H. Y. Chang, “Application traffic classification at the early stage by characterizing application rounds,” *Inf. Sci. (Ny).*, vol. 232, no. 22, pp. 130–142, 2013.

[6] L. Peng, B. Yang, Y. Chen, and T. Wu, “How many packets are most effective for early stage traffic identification: An experimental study,” *China Commun.*, vol. 11, no. 9, pp. 183–193, 2014.

[7] G. Aceto, D. Ciuonzo, G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, “Traffic Classification of Mobile Apps through Traffic Classification of Mobile Apps through,” no. September, pp. 2–7, 2017.

[8] R. Alshammari and A. N. Zincir-Heywood, “Identification of VoIP encrypted traffic using a machine learning approach,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 27, no. 1, pp. 77–92, 2015.

[9] M. Shafiq, X. Yu, and D. Wang, "Robust Feature Selection for IM Applications at Early Stage Traffic Classification Using Machine Learning Algorithms," *2017 IEEE 19th Int. Conf. High Perform. Comput. Commun. IEEE 15th Int. Conf. Smart City; IEEE 3rd Int. Conf. Data Sci. Syst.*, pp. 239–245, 2017.

[10] B. Hullár, S. Laki, and A. György, "Efficient methods for early protocol identification," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 10, pp. 1907–1918, 2014.

[11] B. P. Lin, F. J. Lin, and L. Tung, "The Roles of 5G Mobile Broadband in the Development of IoT, Big Data, Cloud and SDN," *Commun. Netw.*, vol. 8, no, no. February, pp. 9–21, 2016.

[12] B. P. Lin, L. Tung, F. Tseng, I. Hsieh, Y. Wang, and S. Chou, "Performance Estimation of MAR for Outdoor Navigation Applications based on 5G Mobile Broadband by using Smart Mobile Devices."

[13] B. S. P. Lin, W. H. Tsai, C. C. Wu, P. H. Hsu, J. Y. Huang, and T. H. Liu, "The design of cloud-based 4G/LTE for mobile augmented reality with smart mobile devices," *Proc. - 2013 IEEE 7th Int. Symp. Serv. Syst. Eng. SOSE 2013*, pp. 561–566, 2013.

[14] D. Sinh, L. Le, L. Tung, and B. P. Lin, "The Challenges of Applying SDN / NFV for 5G & IoT," in *The 14th IEEE - VTS: Asia Pacific Wireless Communications Symposium (APWCS), Incheon, Korea, August 2017*.

[15] Le Luong Vy; Li-Ping Tung; Bao-Shuh Paul Lin, "Big data and machine learning driven handover management and forecasting," in *IEEE Standards for Communications and Networking (CSCN), 2017 IEEE Conference on*, 2017, pp. 214–219.

[16] L. Le, D. Sinh, L. Tung, and B. P. Lin, "A Practical Model for Traffic Forecasting based on Big Data , Machine-learning , and Network KPIs," pp. 3–6, 2018.

[17] M. Conti, L. V. Mancini, R. Spolaor, and N. V. Verde, "Analyzing Android Encrypted Network Traffic to Identify User Actions," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 1, pp. 114–125, 2016.

[18] B. S. Lin *et al.*, "The design of big data analytics for testing & measurement and traffic flow on an experimental 4G/LTE network," *2015 24th Wirel. Opt. Commun. Conf. WOCC 2015*, pp. 40–44, 2015.

[19] D. L. C. Dutra, M. Bagaa, T. Taleb, and K. Samdanis, "Ensuring End-to-End QoS Based on Multi-Paths Routing Using SDN Technology," *GLOBECOM 2017 - 2017 IEEE Glob. Commun. Conf.*, pp. 1–6, 2017.

[20] H. Krishna, N. L. M. Van Adrichem, and F. A. Kuipers, "Providing bandwidth guarantees with OpenFlow," *2016 IEEE Symp. Commun. Veh. Technol. Benelux, SCVT 2016*, pp. 0–5, 2016.