SDN/NFV, Machine Learning, and Big Data Driven Network Slicing for 5G

Luong-Vy Le[1], Bao-Shuh Paul Lin[2,3], Li-Ping Tung[3], Do Sinh[2]
[1]College of Electrical and Computer Engineering, National Chiao Tung University, Hsinchu, Taiwan
[2]Deparment of Computer Science, National Chiao Tung University, Hsinchu, Taiwan
[3]Microelectronics & Information Research Center, National Chiao Tung University, Hsinchu, Taiwan
Email: leluongvy.eed03g@nctu.edu.tw, bplin@mail.nctu.edu.tw, lptung@nctu.edu.tw, dosinhuda.cs04g@nctu.edu.tw

***Abstract*—5G networks are expected to be able to satisfy a variety of vertical services for mobile users, business demands, and automotive industry. Network slicing is a promising technology for 5G to provide a network as a service (NaaS) for a wide range of services that run on different virtual networks deployed on a shared network infrastructure. Moreover, the SON (self-organizing network) in 5G is expected as a significant evolution to guarantee for full intelligence, automatic, and faster management and optimization. To deal with those requirements, recently, software-defined networking (SDN), network functions virtualization (NFV), big data, and machine learning have been proposed as emerging technologies and the necessary tools for 5G, especially, for network slicing. This study aims to integrate various machine learning (ML) algorithms, big data, SDN, and NFV to build a comprehensive architecture and an experimental framework for the future SONs and network slicing. Finally, based on this framework, we successfully implemented an early state traffic classification and network slicing for mobile broadband traffic applications implemented at Broadband Mobile Lab (BML), National Chiao Tung University (NCTU).**

Keywords— SDN/NFV; Machine Learning; Big Data; SON; 5G; Network slicing; Application identification.

I. INTRODUCTION

According to Third-Generation Partnership Project (3GPP) and the IEEE Computer Society, network slicing, SDN, NFV, IoT, Cloud, ML, and Big Data are among the most important key technologies to achieve the goals of 5G, such as the increasing demands from the end users, the reducing in capital expenditure (CAPEX) and operational expenditure (OPEX) (at least 20% compared to today) [1][2]. Base on those technologies, network operators expect a flexible and efficient framework to deploy their applications quickly and efficiently with various performance requirements, such as high data rate, low latency (less than 1ms), high reliability, and high mobility within a physical network infrastructure [3]. Recently, network slicing was considered as one of the most important and innovative evolutions in 5G to support various services, such as IoT, CDN (content delivery network), and healthcare, by providing end-to-end logical networks with a set of isolated virtual resources of one service from others [3][4][5][6]. However, as a new paradigm, besides providing a number of advantages, network slicing has raised many issues that open up innovation opportunities attracting a lot of researchers [7].

The approach of network slicing is closely related to network virtualization, it creates multiple logical end-to-end networks from an underlying physical network [8]. Moreover, recently, SDN/NFV, which have abilities to provide the full power of programmability for creating and managing quickly multiple virtual networks, have been exploited to implement on-demand functions of network components. For example, they enable network operators to flexibly allocate network resources, such as bandwidth and radio frequency, for virtual RANs [9][10]. However, Two studies [3] and [5] also showed many challenges of developing network slicing, such as network reconstruction, slices cooperation, and resource scheduling and optimization. Therefore, to deploy network slicing successfully, the SON in 5G will be significantly enhanced from the current in 4G. It must provide a better integration and more intelligent capabilities, such as the capability of self□configuration, self□optimization, and fault management, based on ML and big data applications [1][6][11][12]. For example, when a new slice is created, the SON first automatically configures network parameters (NPs) for network elements, and then it continuously adjusts the NPs based on network conditions to ensure that other slices are also operating at their peak performance.

In the past few years, we have been focusing on developing architecture and technology for collecting and analyzing a huge amount of mobile network data based on ML, big data, SDN/NFV, cloud, and IoT over the experimental 4G/LTE network testbed, located at MIRC/BML (Microelectronics and Information Research Center/Broadband Mobile Lab) in the campus of NCTU [13][11]. Many applications were introduced. For example, research [2] addressed the importance of cloud computing in 4G/LTE for Mobile Augmented Reality (MAR) with smart mobile devices; research [11] investigated various ML algorithms to manage and forecast handover behaviors of a huge number of cells; and research [13] proposed a Hidden Markov Model (HMM) to classify traffic flows of mobile broadband applications with high accuracy. In this study, the authors utilize the result of those studies to develop a comprehensive architecture for network slicing. We first propose an approach for clustering and classifying mobile applications at an early stage, then each application is preserved a relevant bandwidth controlled by an SDN controller to guarantee that the network is working at high efficiency.

The remainder of the paper is organized as follows: Section II describes network slicing architecture based on SDN/NFV, cloud, ML, and big data; Section III introduces our experimental platform; Section IV discusses the review of mobile traffic identification approaches; Section V introduces and implements network slicing for mobile traffic applications; finally, Section VI concludes the present study.

II. NETWORK SLICING ARCHITECTURE BASED ON SDN/NFV, ML, BIG DATA, AND CLOUD FOR 5G

In research [1], the authors examined the role of SDN/NFV, big data, ML, IoT, and cloud for 5G. In research [14], we proposed a Full-SDMN (software define mobile network) architecture with full of programmable capability on every segment and connection through access and core mobile network. This study enhances that architecture to make it more powerful by integrating the current architecture with ML and Big data framework as shown in Fig. 1, and its main components and functions for 5G are described below.

Firstly, all network functions such as the home subscriber server (HSS), policy and charging rules function (PCRF), Authorization and Accounting (AAA), mobility management entity (MME), serving gateway (S-GW), packet gateway (P-GW), and all signaling connections are installed and run on commodity servers in Data Center Network (DCNs). Furthermore, data planes are decoupled from control planes, and they are deployed in DCNs, which contain programmable Full-nodes (defined in [15]). Those components are managed by SDN/NFV orchestrations to create and control multiple logical network functions quickly and easily.

Secondly, to support various types of Radio Access Networks (RANs), such as 3GPP (e.g. LTE-E-UTRAN, UMTS-UTRAN, GPRS-GERAN), non-3GPP (e.g. WiMAX, CDMA2000), and Wireless Sensor Network (WSN), Access Gateways (AGWs) are deployed at the edge of DCNs. In this scenario, an AGW plays the role of both the S-GW and P-GW to communicate between VRANs (virtual-RANs) and VCNs (Virtual core networks) using IEEE 802.1ad known as Q-in-Q VLAN tunneling method.

Thirdly, the control plane components, such as Full-SDMN orchestration, NFV orchestration, and SDN controller, cooperate with one another to generate flows so that they can manage and control the network components of the physical infrastructure to form logical networks that meet the requirements of slices.

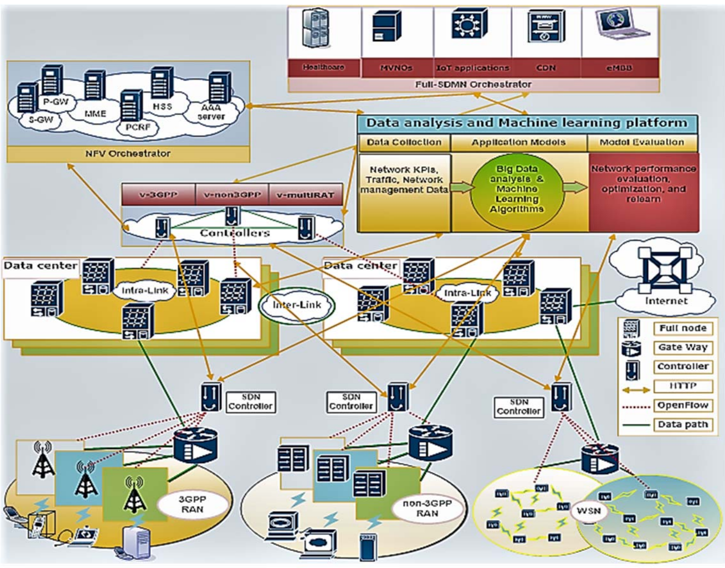

Figure 1. Proposed Network Architecture and Technologies for 5G

Finally, the data mining and ML components work as the brain of the architecture. They are exhibited as breakthrough approaches deciding the success of the network slicing concept in solving new problems for network operations in terms of coordination, configuration, management, and optimization to keep the network operating smoothly at high performance and providing better services. In this architecture, big data and ML can be deployed at any segment of the network, they can interact with the SON, the SDN controllers in both Core and Access networks. The detail of big data and ML platform will be described in next section.

III. EXPERIMENTAL PLATFORM FOR DEPLOYMENT OF BIG DATA AND ML ALGORITHMS FOR NETWORK SLICING

Recently, the utilization of ML and big data for empowering the SON in 5G has been studied extensively. For example, the research [11] introduced the possible ML algorithms that have been taken or could be taken in designing SON functions. It also illustrated the framework to empower the SON using big data and ML, which were applied in the network of Industrial Technologies Research Institute (ITRI) and NCTU as shown in Fig.2 and described below:

First, the model collects necessary network KPIs (Key Performance Indicators) [11], traffic, and other data from all sources of the network for management and optimization purposes. Next, based on the application purpose, the data preparation step extracts all necessary data attributes from the collected data. Then, they are applied to big data and ML models. Besides, it is important to set expected KPI targets that are used to evaluate the success of the model. Finally, the application model is applied to the SON, and then the expected KPIs are analyzed to determine whether new NPs match for the network performance or not. If the network behavior achieves the expected goal, the new NPs will be applied. Otherwise, identify problems, relearn, and update the behavior of the model.

In this architecture, the most important component deciding the power of the framework is technologies for data analysis and ML whose detail framework is illustrated in Fig.3. As can be seen, it is an open platform, which can integrate various state-of-the-art software and programming languages such as Apache Kafka, ZooKeeper, HBase, Apache Spark, InfoSphere, Matlab, and R for data collection, data transformation, data storage, data analysis, visualization, and ML applications. For example, Spark, a distributed computing framework, is emerging as a perfect solution for big data analysis and ML on clusters due to its powerful library (MLlib) for data processing, ML, Streaming (online learning), performance evaluation, and data visualization with high computing performance (100x faster than Hadoop MapReduce in memory). Generally, this system comprises of multiple software components running independently and concurrently on multiple physical machines by using Docker

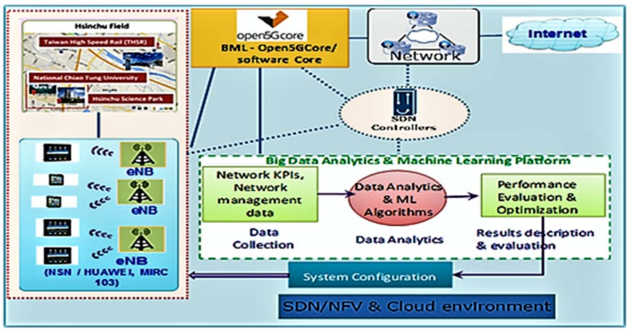

Figure 2. Experimental Platform

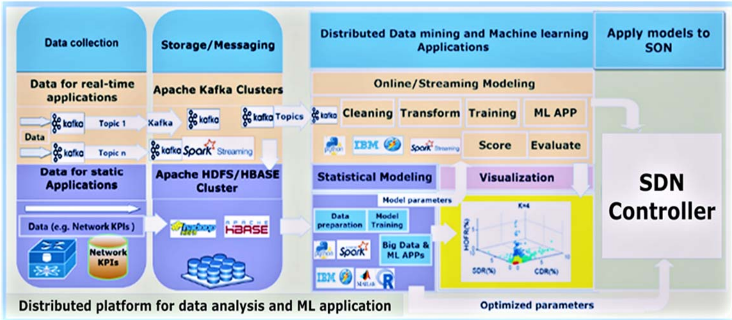

Figure 3. Data analysis and ML framework

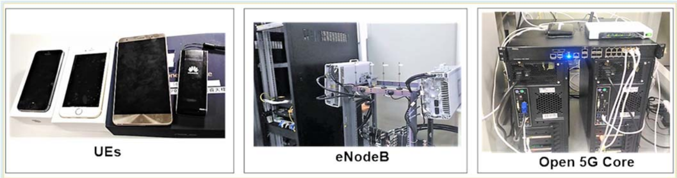

Figure 4. Real experimental equipment

containers; therefore, practically, it is fast and robust with full of automation, flexibility, scalability, and reliability. Consequently, it may be considered as a significant solution for mobile Edge Computing (MEC), a key emerging technology in 5G, which is expected to support a powerful computing performance and low-latency communication by moving computing, storage, and networking resources from remote public clouds closer to the edge of the network. In other words, by extending the computing to the edge of the network, network providers can create virtual slices within the access network, and then they can experience a low end-to-end delay, which is a crucial requirement for 5G applications. Moreover, this platform can be applied to both statistical models and online/streaming models [6]. Finally, Fig. 4 shows some real equipments of the experimental network using in this study.

IV. Overview of Early Application Identification

Early application detection, an essential step for traffic engineering and network slicing, has been explored in many studies [13] [15] [16] [17]. There are three common approaches for early mobile traffic classification: port-based approaches, deep packet inspection (DPI) approaches, and packet size approaches. However, those studies concluded that traffic classifications based on the packet sizes of some first packets are effective enough to achieve ideal identification performances, and they are much better than port-based and DPI approaches. Specifically, research [16] proposed model to characterize and classify both TCP and UDP flows from the application layer perspective, and the results showed that this method significantly improved classified accuracy than doing so from the transport layer perspective with a minimal 15–30% of improvement of overall accuracy. Furthermore, research [16] found that the most efficient number of packets using for the classification model is from 5 to 7 packets after they applied their traffic classification models with several ML algorithms, such as Naïve Bayes, SVM, Random Forest, logistic, etc. That means too many packets and too few packets will reduce the model accuracy. Especially, in the study [15], we proposed an HMM-based to classify the internet traffic flow of mobile broadband application using the packet size and packet transmission direction with 99.17% accuracy for 6 types of mobile applications. In the following section of this study, the authors propose a comprehensive process of using their platform to slice the network based on an early application classification model.

V. Network Slicing Based on Early Mobile Traffic Applications Identification

The proposed practical process for network slicing illustrated in Fig.5 involves 4 main steps. Firstly, in the data preparation step, necessary data features are collected and extracted. Next, a clustering model groups applications into different clusters so that all applications in a cluster have similar traffic patterns. Then, a classification model is built based on the clustering model result. Finally, new coming traffic flows are classified and mapped to relevant slices under the control of an SDN controller.

A. Data preparation

Features selection is a critical step deciding the accuracy of the classification model and clustering model; therefore, input features must accurately represent characteristics of applications. As discussed in the previous section, it is possible to identify an application by observing the size and the direction of its first few packets, which characterize for different negotiation stages that are usually defined uniquely for each application. In this study, the features were extracted from both application layer and transport layer perspective. As described in the researches [16] and [13], when an application session starts, it has several negotiation stages between client sides and server sides. This process consists one or more continuous interaction rounds, each round contains one or multiple

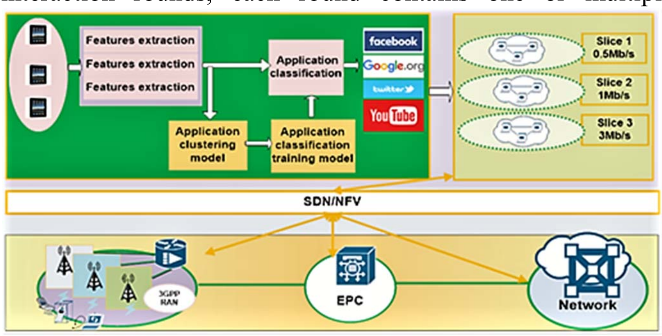

Figure 5. Network slicing framework for mobile broadband

continuous application messages (layer 7 messages) as shown in Fig.6. After that, those messages are segmented and encapsulated at the transport layer, and then, they are encapsulated into packets at the network layer. Here, the number of the packets and their sizes of the first interaction round were used as the features of the classifying model. Moreover, the sizes of the first 5 packets of each TCP/UDP flow and the source port, and the destination port of the connection are used. Table 1 is an example of some samples of several flows, each data sample contains 13 features.

B. Clustering model for mobile traffic applications

In real mobile networks, there are numerous types of mobile broadband applications, and they are increasing rapidly. Therefore, it is impossible that a single classification model can obtain fully labeled training data for all new applications. Moreover, providing individual QoS policy or slice for each application is inefficient, event impacts to the accuracy of the classification model. To overcome these challenging issues, we first propose a clustering model to group and label applications having similar traffic behavior into a group. Therefore, in the classification step later, a new or unknown application will be classified into a group in which its behavior is the most similar to other applications in the group.

1) Features selection

Features selection directly influences the performance of traffic clustering model. In this experiment, the input features were extracted and summarized from the data set described in the previous Subsection (Subsection A). Each data sample consists of 22 features, which represent the average value and the mean absolute deviation value of its total data samples. Table 2 is an example of input features of SSL Google, both real values and normalized values (will be illustrated in next subsection) are shown.

2) Machine learning algorithm

This experiment uses K-means as an unsupervised learning algorithm for clustering mobile applications. Statistically, K-means is the most famous practical approach for clustering model because it is very effective and scalable. It automatically explores and detects complex patterns of regularities in traffic flows to find applications that have similar traffic behavior. Since K-means is a distance-based technique, all input attributes need a comparable range. Here, we use min-max normalization method to normalize all data features into the range [0,1] interval as shown in Table 2. In addition, a de-normalization can be used to recover the original values if necessary.

3) Clustering result

The model set 3 as the number of clusters associating with 3 slices (0.5Mbps, 1Mbps, 3Mbps). The result of the clustering model for 21 applications is summarized in Table 3. As can be seen, the number of application in cluster 2 is the most. In the future, when the number of applications is huge, the number of clusters should be optimized to select the best value of K by evaluating several metric as described in the research [15].

C. Classification models for mobile traffic applications

In this subsection, we investigate several popular ML algorithms to classify new coming traffic flows into 3 distinct application categories.

1) Machine learning algorithms

Naïve Bayes is a simple probabilistic classifier, which makes classifications by using Bayes' theorem to compute the probability of a data sample that belongs to a given class. It assumes that all features are conditional independence one another, this algorithm is easy to train but can provide impressive performance in solving many problems effectively.

Support Vector Machine(SVM):
SVM is a powerful and popular ML algorithm for patterning recognition. It is used as a supervised learning of finding the largest margin for linear and non-linear classifiers. It makes predictions for a new input only depend on calculation on a subset of training data points known as support vectors; therefore, SVM has an advanced solution for large-scale and complex problems.

Neural network (NN): NN is a non-linear algorithm, representing the state-of-the-art technology, providing the best solution to many regression and classification problems, and pattern recognition. Recently, there were many applications use NN for prediction and forecasting applications [11].

Gradient Boosted Tree (GBT): GBT, an ensemble of decision trees, supports for both regression and classification problems. It iteratively trains multiple decision trees to minimize the cost function.

Random Forest (RF): RF is known as an ensemble of decision trees, it consists of a chosen number of decision trees and can support multiclass classification. To classify a data point, each tree in the forest is learned on a different set of data points and predicts a class on its own. Finally, the output of the model describes a random forest, which predicts the final class as the one that has majority votes.

2) Experimental setup

A classification model is a form of supervised learning; therefore, its process includes training and testing phases. For

Table 1 Features extraction

Connection	Packet_1_size	Packet_2_size	Packet_3_size	Packet_4_size	Packet_5_size	No_Packet_A	No_Packet_B	No_Packet_A&B	Packets_A (Byte)	Packet_B (Byte)	Packet_A&B (Byte)	Dport	Sport	Label
10.100.11.90:49324 > 17.154.66.43:443	1125	85	437	165	53	13	2	15	8469	218	8687	443	49324	SSL.AppleiTunes
10.100.11.90:49688 > 74.125.204.155:443	581	1397	1397	1397	581	14	1	15	10844	1397	12241	443	49688	SSL.Google
10.100.11.67:44411 > 54.230.215.75:80	416	1448	1448	1448	1448	2	5	7	1864	5274	7138	80	44411	SSL.Amazon
10.100.11.67:58202 > 31.13.95.48:443	1176	1398	131	1398	1223	25	2	27	28074	637	28711	443	58202	SSL.Instagram

Table 2 Input features selection for clustering model

Features	packet.1.size	packet.2.size	packet.3.size	packet.4.size	packet.5.size	No_packet_A	No_packet.B	No_packet.A&B	size_A (byte)	size_B (byte)	size_A&.B (byte)	Application label
Mean_value (real value)	1201.97	1044.23	1059.84	1101.41	938.61	19.50	2.25	21.75	18093.08	1770.46	19863.55	SSL.AppleiTunes
Mean_abs_deviation (Real value)	201.56	441.17	400.38	354.25	509.12	7.69	1.56	7.65	10273.06	1998.43	10343.90	
Mean_value (Normalized value)	0.99	0.74	0.75	0.79	0.67	0.26	0.03	0.25	0.20	0.03	0.20	
Mean_abs_deviation(Normalized value)	0.39	0.78	0.77	0.64	0.77	0.11	0.03	0.09	0.10	0.03	0.09	

Table 3 Cluster result

Cluster 1	Cluster 2	Cluster 3
SSL.Amazon	SSL.facebook	Twitch
SSL.Google	SSL.Apple	SSL.YouTube
POPS	GMail	SSL.Skype
SSL.Wikipedia	SSL.Instagram	TeamViewer
SSL.MSN	SSL.Twitter	SSL.Apple ITunes
	Google Drive	Spotify
	SSL.Deezer	
	SSL.Snapchat	
	WeChat	
	WhatsAPP	

each experiment, training dataset of 20000 data samples was chosen randomly from the data of different applications collected from different connections. Also, 4000 testing samples were chosen randomly and different from the training dataset to evaluate the performance of each ML algorithm. The input attributes of each training sample are the 13 features extracted from subsection B (Table 1) and the output is a cluster number.

3) Experimental result

Table 4 summarizes the classification results, it is noticeable that all the algorithms achieve high accuracy. Naïve Bayes and SVM give quite similar performance, NN obtains slightly better accuracy, while two major ensemble algorithms: GBT and RF give the best performance that they can exactly classify traffic flows (nearly 100%). Fig.7 and Table 5 show the classification accuracy for each cluster and each type of applications, respectively. It is obvious that Naïve Bayes, SVM, and NN give different performances for each cluster and application. For example, in case of identifying cluster 1, SVM and NN achieved higher accuracy than Naïve Bayes model, while Naïve Bayes model's performance was better for cluster 3. However, the difference was not significant. We also found that some applications, such as SSL.Google, SSL.Wikipedia, and SSL.Instagram, are more difficult to identify than others due to the fact that some of their flow features (e.g. the first packet size) vary significantly among flows of different connections. However, GBT and Ensemble can exactly classify them into their relevant clusters.

Table 4 Classification results

Machine Learning Algorithm	Accuracy (%)
Naïve Bayes	96.62
SVM (Support Vector Machine)	97
Neural network	98.34
Tree Ensemble	99.68
Random Forest	99.68

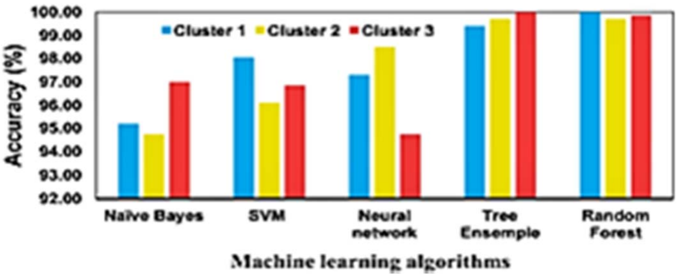

Figure 7. Classification accuracy (%) of clusters

D. Network Slicing based on application classification

After an application is classified, all of its flows will be assigned a relevant bandwidth value using flow tables and meter tables of OpenFlow Switch as shown in Fig.8 for the uplink direction. Firstly, the SDN controller setups flow table entries for each connection based on its classification number to instruct how the flows (packets) are executed. Secondly, a meter is attached directly to flow entries to measure and control the data rate of packets. As can be seen, each meter table consists of several meter entries, which are defined for traffic flows. Here, the meter has 3 bands corresponding to 3 target data rates. Once the band is chosen, the meter applies the QoS processing to packets of the flow. Finally, SDN controller also uses flow's information to install flow entries for downlink direction packets. In this experiment, our UE opened a high-quality video on Youtube, Fig.9 and 10 show the experimental results before and after applying the model, respectively. In the first case, the mobile was provided the default bandwidth (1Mbps), not enough for playing video smoothly, even dropped frames. In the second case, the model identified and classified Youtube traffic belong to cluster 3, and provided 3Mbps for its flows, as the result, the UE played the video smoothly without dropped frames.

E. Evaluation computing performance of the system

This subsection evaluates the computing capacity of the framework by running an example in it with a specific configuration. Here, only one computing Worker, which has 8 cores and 3.9 GB memory, of the distributed computing framework, was activated for computing purpose as shown in Fig.11. We first randomly generated a stream of 50.000.000 data samples for classification testing data, then Kafka received the data, created topic, and produced data to Apache Spark. After that, Spark classified the incoming flows by its SVM classifier, which was built from Spark MLlib 2.0. In online

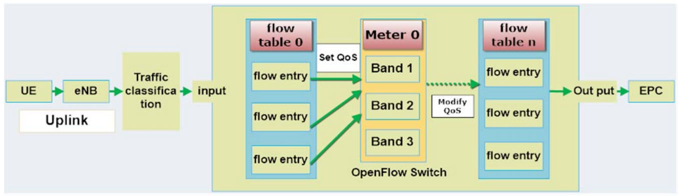

Figure 8. Network slicing based on traffic classification

Figure 9. Video quality of model without network slicing

Figure 10. Video quality of model with network slicing

Figure 11. A computing worker configuration

Workers

Worker Id	Address	State	Cores	Memory
worker-20171011083212-192.168.156.98-36925	192.168.156.98:36925	ALIVE	8 (8 Used)	3.9 GB (1024.0 MB Used)

learning, a streaming model continually records, updates, and processes its state when a new sample is coming. However, in this study, to achieve better performance, we used the mini-batch method. That means the stream is discretized into a sequence mini-batches which called sequence of Resilient Distributed Datasets (RDD). A mini-batch combines a number of records, and its length can be either by time duration or number of records. This study used the time duration method, therefore, each mini-batch has the same time interval (3 seconds), but different numbers of records with others. Fig. 12 summarizes the input data rate, the processing time of Kafka. Fig. 13 shows the time, size, scheduling time, processing delay, and total delay of each batch. In summary, it took 5.7 minutes for the Worker to classify all the data samples smoothly with a small time for scheduling and processing.

VI. CONCLUSION

Network slicing for 5G will be the combination of many upgraded technologies to meet the diverse requirements for establishing the lower latency, more transparent, and more coordinated network. This Study demonstrated the architecture and the powerful experimental framework for 5G and network slicing. Furthermore, we also proposed and implemented the comprehensive process for slicing mobile broadband traffic with high accuracy. The initial experimental results also open up many interesting ideas of using those technologies to build more complex network slicing applications. Our future work will focus on the areas of using the framework for mobile Edge Computing, CORD applications and SDN-based (E-CORD, R-CORD, M-CORD) for 5G.

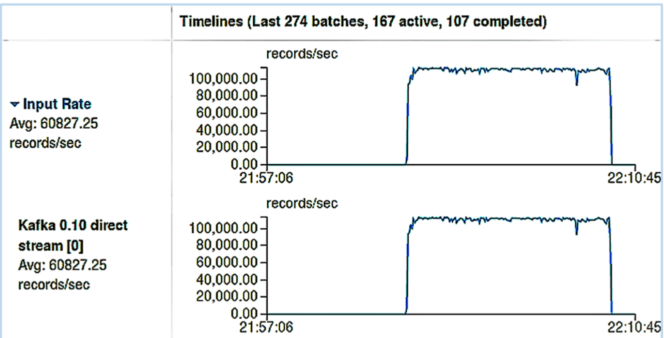

Figure 12. Input data and Kafka processing timelines

Batch Time	Input Size	Scheduling Delay (?)	Processing Time (?)	Total Delay (?)
2017/10/11 22:08:36	277296 records	4 ms	1 s	1 s
2017/10/11 22:08:33	326904 records	0.6 s	1 s	2 s
2017/10/11 22:08:30	334854 records	2 s	1 s	3 s
2017/10/11 22:08:27	334854 records	0 ms	5 s	5 s
2017/10/11 22:08:24	329448 records	2 ms	1.0 s	1.0 s
2017/10/11 22:08:21	335490 records	1 ms	1 s	1 s
2017/10/11 22:08:18	334059 records	2 ms	0.9 s	0.9 s
2017/10/11 22:08:15	337239 records	2 ms	2 s	2 s

Figure 13. Mini-batch timeline and Computing capacity

ACKNOWLEDGMENT

This work was financially supported by the Center of Open Intelligent Connectivity from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan. It was also supported in part by the Ministry of Science and Technology of Taiwan and National Chiao Tung University under Grants: MOST 106-2221-E-009-008.

REFERENCE

[1] 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. 1, pp. 9–21, 2016.

[2] 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,” *IEEE 7th Int. Symp. Serv. Syst. Eng. SOSE 2013*, pp. 561–566, 2013.

[3] X. Li *et al.*, “Network Slicing for 5G: Challenges and Opportunities,” *IEEE Internet Comput.*, vol. 21, no. 5, pp. 20–27, 2017.

[4] B. Han, S. Tayade, and H. D. Schotten, “Modeling profit of sliced 5G networks for advanced network resource management and slice implementation,” *Proc. - IEEE Symp. Comput. Commun.*, pp. 576–581, 2017.

[5] K. Katsalis, N. Nikaein, E. Schiller, A. Ksentini, and T. Braun, “Network Slices toward 5G Communications: Slicing the LTE Network,” *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 146–154, 2017.

[6] H. Zhang, N. Liu, X. Chu, K. Long, A. Aghvami, and V. C. M. Leung, “Network Slicing Based 5G and Future Mobile Networks: Mobility, Resource Management, and Challenges,” *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 138–145, 2017.

[7] P. Rost *et al.*, “Network Slicing to Enable Scalability and Flexibility in 5G Mobile Networks,” *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 72–79, 2017.

[8] L. Velasco *et al.*, “An Architecture to Support Autonomic Slice Networking,” *J. Light. Technol.*, vol. 36, no. 1, pp. 135–141, 2018.

[9] Y. Peng, X. Huang, and J. Qi, “Cross-media shared representation by hierarchical learning with multiple deep networks,” *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2016–Janua, no. August, pp. 3846–3853, 2016.

[10] C. Bouras, A. Kollia, and A. Papazois, “SDN & NFV in 5G: Advancements and challenges,” *Proc. 2017 20th Conf. Innov. Clouds, Internet Networks, ICIN 2017*, pp. 107–111, 2017.

[11] 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), Helsinki, 2017*, 2017, pp. 214–219.

[12] 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,” in *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2018, pp. 1–4.

[13] 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.

[14] D. Sinh, L. Le, L. Tung, and B. P. Lin, “The Challenges of Applying SDN / NFV for 5G & IoT,” *14th IEEE - VTS Asia Pacific Wirel. Commun. Symp. (APWCS), Korea, Sep 2017.*

[15] L. Bernaille, R. Teixeira, and K. Salamatian, “Early application identification,” *Proc. 2006 ACM Conex. Conf.*, p. 6:1--6:12, 2006.

[16] 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.

[17] Y. Wang, Y. Xiang, and J. Zhang, “Network traffic clustering using Random Forest proximities,” *IEEE Int. Conf. Commun.*, pp. 2058–2062, 2013.