SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TNC TRANSACTIONS ON NETWORKS AND COMMUNICATIONS

VOLUME 6, NO. 5
ISSN: 2054 -7420

Big Data and Machine Learning Driven Open5GMEC for Vehicular Communications

Luong-Vy Le[1], Do Sinh[2], Bao-Shuh Paul Lin[2,3], Li-Ping Tung[3]
[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; dosinhuda.cs04g@nctu.edu.tw; bplin@mail.nctu.edu.tw; lptung@nctu.edu.tw

ABSTRACT

Mobile Edge Computing (MEC) is an emerging technology and an essential component of 5G networks to bring cloud services closer to users. That means data collection, storage, processing, computing, communication, and network control are implemented at network edges. MEC is expected to be able to satisfy a variety of delay-sensitive services and applications. On the other hand, the development of vehicles to everything (V2X) communication brings many requirements to future networks to guarantee full intelligence, automatic, and faster computation, management, and optimization to fulfill network QoS (quality of service) and QoE (quality of experience). To deal with those requirements, recently, software-defined networking (SDN), network functions virtualization (NFV), big data, and machine learning (ML) have been proposed as emerging technologies and the necessary tools for MEC and vehicular networks. This study aims to integrate those technologies to build a comprehensive architecture and an experimental framework for future 5G MEC called Open5GMEC. Moreover, the authors analyzed challenges and proposed relevant solutions for future vehicular communications in 5G networks. Finally, based on this framework, we successfully implemented several powerful ML-based applications for V2X such as object detection, network slicing, and migration services, which are executed at Broadband Mobile Lab (BML), National Chiao Tung University (NCTU).

Keywords: 5G, Vehicular communication, V2X; Automotive driving, SDN/NFV; Machine Learning; Big Data; MEC.

1 Introduction

In Mobile Edge Computing (MEC) has recently been proposed as a promising paradigm to overcome the requirements of future networks that enabling a wide range of benefits such as high bit rate, high availability, low latency (less than 1ms), and high mobility in heterogeneously converged connectivity environments by shifting computational efforts from the centralized cloud computing to edge servers. The servers are usually deployed and co-hosted at base stations or near mobile users to eliminate and reduce a tremendous amount of data routing through the core network. As a result, the core network can be simplified, and the end-to-end (E2E) latency is reduced [1][2]. Furthermore, due to the exponential

DOI: 10.14738/tnc.65.5410
Publication Date: 31st October, 2018
URL: http://dx.doi.org/10.14738/tnc.5410

increase of IoT applications and the massive deployment of new vertical business services, MEC is expected to be a flexible and efficient framework, in which network service providers can deploy their applications quickly and efficiently. With particular reference to vehicular communication networks, it is expected to meet various communication requirements of future Intelligent Transportation Systems (ITS) such as low latency, location awareness, and real-time response applications (e.g., driving safety applications and real-time warning on the road). Moreover, the IEEE vehicular communication standard also provided many requirements for vehicular communications like congestion control mechanisms, fairness in accessing resources, and the availability of infotainment services [3]. On the other hand, vehicle-to-everything (V2X) is a promising solution to improve road safety, traffic efficiency, and meet various QoS requirements in different application scenarios [4]. Therefore, recently, 5G-based V2X has been actively conducted by the Third Generation Partnership Project (3GPP) to provide solutions for vehicular communications [5][6]. For example, research [6] explored MEC for 5G-enabled software defined vehicular networks where SDN was exploited and combined with MEC to strengthen vehicular systems. However, MEC for vehicular communications is still in its early stage with many unresolved challenges ranging from reliability, flexibility, scalability architecture to data management and integration, even security issues and so on.

This study, firstly, explores various emerging technologies, such as big data, ML, SDN/NFV, and cloud computing, and then integrates them to propose a comprehensive platform called Open5GMEC for 5G and vehicular networks. SDN/NFV are considered as the most critical technologies for 5G networks to provide the full power of programmability, interoperability, agility, short time to market, and low-cost solution by virtualizing network components and creating multiple logical end-to-end networks [7]. Moreover, recently, ML and big data have been exploited as the key technologies to empower computing components in 5G networks. For example, they make the 5G MEC and SON better integration and more intelligent capabilities [8] [9][10]. Therefore, they are considered as promising solutions deciding the success of 5G-based vehicular communication in term of network reconstruction, virtual-network cooperation, and resource optimization [11]].

The remainder of the paper is organized as follows: Section II reviews V2X communication and the experimental platform based on Big Data, SDN/NFV, and ML; Section III proposes Open5GMEC experimental architecture; Section IV introduces and analyzes ML-based applications for V2X communication; Section V introduces and applies ML and Open5GMEC for V2X applications; Section VI proposes service migration for V2X; finally, Section VII concludes the present study.

2 Overview of V2X Communication and Experimental Network

2.1 Overview V2X communication

V2X is considered as a crucial service of 5G networks to support a variety of ITS applications, each with a specific set of requirements about data rates, latency, mobility, ubiquity, and reliability. A vehicular communication scenario usually consists of a vast number of smart vehicles and roadside units. Generally, an intelligent vehicle integrates many data generators, storage and communication components for real-time sensing and computing (e.g., cameras, radars, and GPS). Roadside units are commonly deployed along the roads to collect and process data sent by smart vehicles. Generally, V2X can be divided into four main categories: Vehicle-to-Vehicle (V2V), Vehicle-to-Pedestrian (V2P), Vehicle-to-Infrastructure (V2I),

and Vehicle-to-Network (V2N) communications [4] [12]. They include a massive number of connections among a large number of sensors to offer a wide variety of versatile IoT services for ITS consumers, ranging from accident-free transportation, road safety, parking infrastructure, and greener transport to mobile broadband services like video streaming. They also integrate various communication technologies and protocols. For example, V2V can directly communicate with one another to exchange their information through 5G-based networks or new D2D (device to device) communication interface known as PC5 for sharing safety and critical information; V2N can communicate through heterogeneous networks such as WSNs, non3GPPRANs, 3GPPRANs; V2I communication allows vehicles to communicate with eNodeBs to provide various traffic efficiency and entertainment services.

2.2 Experimental platform

In the past few years, big data, ML, SDN/NFV, and cloud have been integrated into the experimental network testbed for 4G/LTE and beyond 5G, located at MIRC/BML (Microelectronics and Information Research Center/Broadband Mobile Lab) in the campus of National Chiao Tung University to build an open architecture

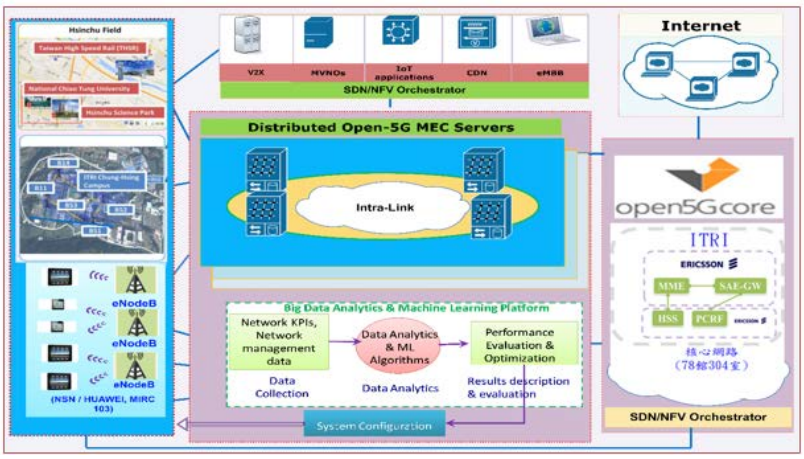

Figure 1. Experimental architecture of 5G network at BML

for developing future network applications [7][11][13][14][15]. For example, the integrated architecture of SDN/NFV, cloud, IoT, and big data in 5G and their roles were introduced in [15]; in studies [8][9][16], we explored and implemented many big data and ML algorithms for 5G applications. Most recently, the collaboration between NCTU and Open Networking Lab (ON.Lab) has accomplished an SDN-IP global peering deployment and established primitive primary CORD (Central Office Re-architected as a Datacenter). Besides, we have also focused on developing P4, ONOS, and CORD applications that are considered as the essential solutions of SDN/NFV technologies for 5G.

Fig.1 describes the abstract architecture and physical components of the current developed 5G testbed at BML. In general, it involves four main parts, the RANs, Open5GCore, Open5GMEC, and applications. The RAN of 5G integrates new technologies such as massive MIMO and optical fiber to support high-speed connections for wide-area wireless connectivity to various types access devices, such as 3GPP (e.g., LTE-E-UTRAN), non-3GPP (e.g., WiMAX), Wireless Sensor Network (WSN). The Open5GCore component was described in the study [7]. Its elements such as S-GW, P-GW, home subscriber server (HSS), and mobility management entity (MME) are virtualized and run on commodity data center under the control of SDN/NFV orchestrations. SDN/NFV orchestrations are also used to manage 5G-based services and

Copyright © Society for Science and Education, United Kingdom

applications, such as network management as a service, V2X, IoT applications, etc. These applications are considered as virtual entities deployed in Docker containers, and they are created and controlled by SDN/NFV applications. The Open5GMEC component will be introduced in the next section.

3 Open5GMEC Architecture based on SDN/NFV, Big Data, and ML

Fig.1 and Fig.2 illustrate the Open5GMEC in the network architecture, in which big data, ML, cloud computing, and SDN controller applications work on NFV environments, and they are considered as the brain of Open5GMEC. As can be seen in Fig.1, Open5GMEC can interact with both the RANs and the SON to move the computing functions to the proximity of UEs and eNodeBs. Therefore, it is considered as a new intermediate layer responsible for data collection, transformation, filtering, aggregation, and processing and then building both online and offline applications for the SON or RAN, even for third-party applications and services. Furthermore, the integration of those state-of-the-art technologies exhibits as breakthrough approaches and promising solutions that deciding the success of the MEC concept in solving new requirements for 5G and V2X networks regarding network coordination, configuration, management, and optimization to support various new services. For example, to build a V2X application, the Open5GMEC firstly collects necessary data such as real-time GPS to determine the position of the vehicle, traffic on the road, and other necessary data from all sources of the network. Next, it extracts and preprocesses the collected data for applying to big data and ML models. And then, the

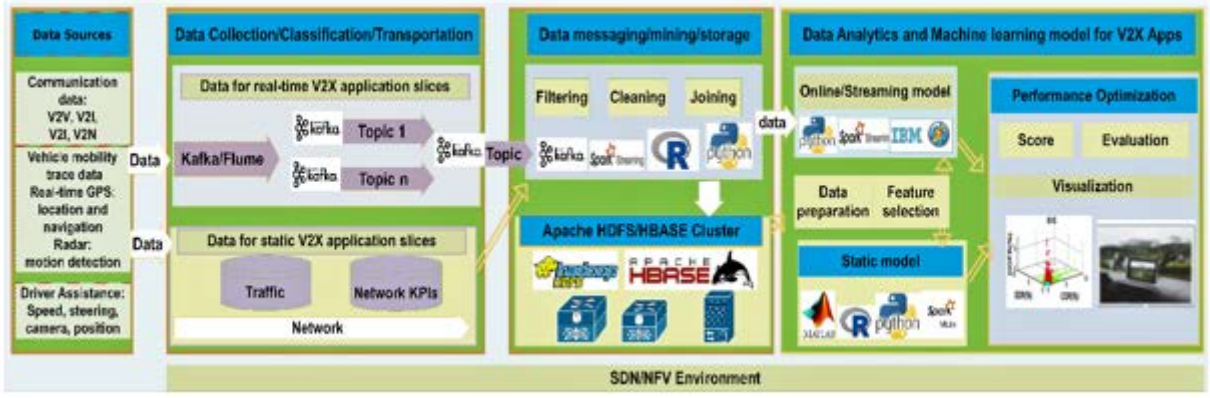

Figure 2. Open5GMEC computing platform for V2X applicationsation intelligently detect which

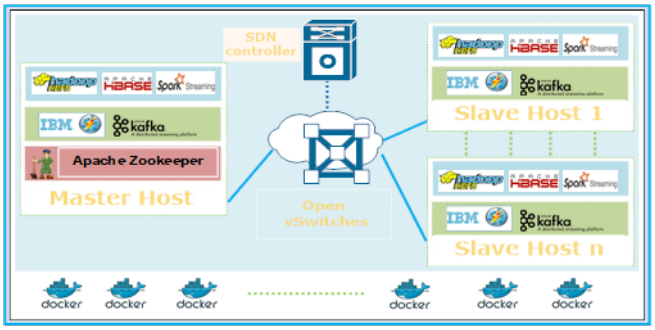

Figure 3. Distributed computing platform

application model is built and sent the result to the SON or RAN. Finally, the network performance is analyzed, evaluated. The following subsection introduces several essential characteristics of the Open5GMEC platform.

Open platform: this is a crucial requirement in deciding the success of Open5GMEC-based ecosystems for IoT and V2X communications. As described in Fig.3, the platform is opened for different state-of-the-art technologies and software can be easily integrated into it for data processing and ML algorithms. For example, Kafka, Flute, and Python are used to collect and transform data from different resources such as network KPIs, sensors of smart vehicles; programming platforms such as Matlab, R, Spark, and InfoSphere can be used for data analytics, visualization, data mining, and ML algorithms.

Distributed platform: Open5GMEC is a distributed computing system in which a host works as the master, and multiple hosts work as workers or executors as illustrated in Fig.3. Each host (master/slave) is a virtual machine working on SDN/NFV environment, and its components are Docker containers running on Linux systems and sharing the OS' kernel; therefore, these containers can be created and started instantly while consuming small resources. These software components usually run independently and concurrently on multiple virtual or physical machines. As a result, it is easy to deploy and upgrade Open5GMEC components. Moreover, in this framework, SDN controller applications are used to manage and control connections among virtual hosts through configuring OpenvSwitch.

Reliability: In the Open5GMEC platform, each component can utilize different methods such as partition, replication, and fault tolerance to improve the reliability of the whole system. For example, in a ZooKeeper cluster, when the primary master is a failure, the backup master takes over the role of primary master. Another example, Kafka stores critical information such as information about topics, consumer offsets, and brokers in the Zookeeper, which generally replicates this data across its ensemble. As a result, failure of Kafka broker or Zookeeper does not affect their clusters and achieve zero data loss, zero downtime. That means failure of a single or a few parts does not affect the system.

Flexibility: An application can be submitted their jobs to the master or any worker in the system. Once the master receives a job, it distributes the workload to its executors, optimizes and controls the number of executors based on the job computing load and the available worker resources.

Security: Security and privacy of the IoT services are critical challenges in the current study. A common threat is cyber- attacks, such as distributed denial of service (DDoS), Jamming attacks, privacy leakage, and man-in-the-middle. In the case of MEC, the data processing and computing are performed at the edge of a network close to the data source and mobile subscribers. Therefore, it bridges the gap between remote data centers and IoT devices to enable a wide range of security advantage. For example, ML can be utilized to detect abnormal attacks in networks such as learning-based IoT malware detection and learning-based authentication.

4 Applying Machine Learning to Open5GMEC and Potential V2X Applications

Recently, ML and big data have been utilized for empowering the SON of 5G, future MEC, and cloud computing to push their performances to the next level of full intelligence and automation [8][9][10][17]. The LTE/4G&5G network testbed, located at MIRC/BML has integrated big data and ML as the vital enabler to develop 5G applications. Generally, there are four categories of ML algorithms, namely, supervised learning, unsupervised learning, reinforcement learning, and deep learning.

Copyright © Society for Science and Education, United Kingdom

4.1 ML categories

Supervised learning, the majority type of practical ML methods used in most current research, is a type of learning that requires a supervisor to learn the model parameters. Its models use a labeled dataset, which contains both input and output information, called training samples to find the relationship that maps from the input attribute space to the labels. As a result, the model gives the expected output for new coming input.

Unsupervised learning, on the other hand, is given an unlabeled input dataset. That means it does not have a supervisor. Therefore, it must investigate the similarity and the relationship among the unlabeled data samples, and then groups them into different clusters.

Reinforcement learning is an area of ML, where an agent learns the optimal behaviors in a trial-and-error manner by interacting with its environment, senses its current state and the state of the surroundings, and chooses an action to achieve a goal. It learns how to map from situations to actions, and therefore, the learner must identify which operation obtains the most reward.

Deep learning is a state-of-the-art and powerful algorithm with the sophistication of self-learning capability. It provides a significant improvement in various fields such as object detection, speech recognition, computer vision, and vehicle trajectory. In general, it is considered as a more in-depth version of neural networks (NN), which consist a series of multiple layers of neurons, the input layer, the hidden layers, and the output layer. Deep learning usually breaks down a very complicated problem into several simple issues to provide more accurate and faster processing.

4.2 Potential ML applications and implementation platforms

Fig.4 summarizes the most popular ML-based applications and the appropriate ML algorithms that can be exploited to empower the Open5GMEC with a full capacity of intelligence and automation.

Prediction and forecasting models are used in dynamic systems to precisely predict and estimate trends of events such as mobile traffic, vehicle mobility, vehicle tracking, and then, the system can keep track of those event's behaviors in changing environments. The suitable ML algorithms for these applications are Hidden Markov Models (HMMs), Linear/Non-linear Dynamical Systems.

Clustering is the most popular and powerful application of unsupervised learning to group a set of more similar data samples together, such as cluster traffic density, V2V neighbor, and abnormal detections. Typical ML algorithms for clustering model are K-means, Mixtures of Gaussians, Automatic relevance determination (ARD), etc.

Classification is the most typical ML models used in vehicular applications, for example, in this study, deep learning is used to classify and detect the object on the roads; Random Forest, SVM, NN, etc. are used to classify mobile broadband applications at the early state.

Diagnosis and decision making are used to analyze the current condition of a system or individual element in a network or vehicle to take timely controlling actions to ensure these components working at their peak performance even under complex situations. Moreover, rapid and relevant controls are essential for vehicular systems to develop safety and driver-assistance applications like power control, adjustment car

direction, car speed. Applicable ML algorithms for these applications are Bayesian networks (BN), Reinforcement learning, logistic regression, decision tree, Gaussian Processes.

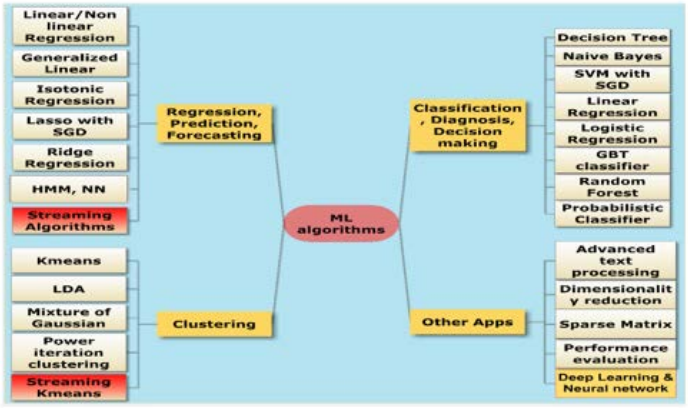

Figure 4. ML algorithms and Applications

Since the majority models of ML application in 5G and IoT are stream mining and distributed computing forms, it is necessary to build ML and big data on powerful platforms. Fortunately, emerging platforms like InfoSphere and Apache Spark are considered as comprehensive platforms supporting most of the popular ML algorithms and tools for big data analytics with various capacities

4.3 ML-based applications for vehicular communication

This subsection identifies the possible ML-based V2X applications in 5G networks. Fig.5 summarizes new applications and services that enable higher mobility, better coordinated, more enjoyable driving experience, more reliable for vehicular communication. For example:

Traffic prediction aims to accurately predict or forecast the future traffic of a road, road sections, even for an area. It has significant roles in improving traffic control, management and other ITS applications, such as traffic congestion avoidance, public vehicle deployment, and road hazard warning. Traffic prediction model usually uses real-time data collected by various roadway sensors or cameras. The prevalent model for traffic prediction is Time Series models, which are based mainly on the historical traces of traffic to forecast the future one. The suitable ML algorithms are dynamic models like HMMs, Deep Learning, and Gaussian Process (GP) [8][9]. LOBECOM 2011), 2011 IEEE, pp. 1–6, 2011.

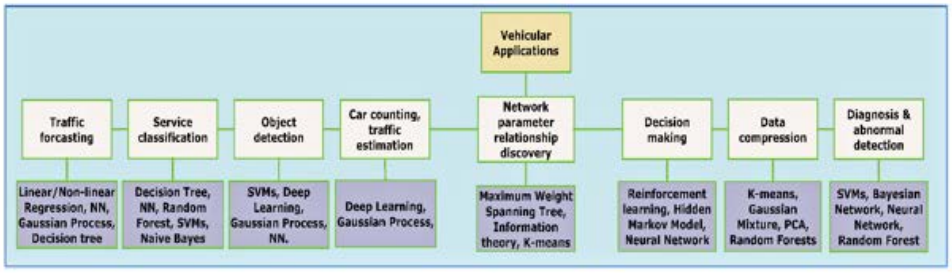

Figure 5. ML-Based Applications for V2X Communication

Vehicle trajectory prediction is crucial for developing advanced driver-assistance systems and autonomous vehicles by providing a better understanding of the traffic environment to perform various functions such as criticality assessment in advance, collision avoidance, trajectory planning, and vehicle

Copyright © Society for Science and Education, United Kingdom

tracking [18]. Moreover, it also plays a vital role in mobile network planning and optimization, such as handover policies; therefore, it has recently received extensive research interest from both academia and industry. For example, research [19] introduced two approaches for vehicle trajectory prediction, the physics-based motion models and maneuver-based methods. These methods observed several running parameters of the vehicles, for example, acceleration, velocity, and direction rate to predict driving behavior using a dynamic Bayesian network. The most suitable ML algorithms are dynamic ML algorithms like KF, dynamic Bayesian networks, HMM, and deep learning.

5 V2X communication Application based on ML and open5G MEC

MEC for vehicular communication applications at BML/NCTU is described in Fig.6. MEC servers were deployed in BML of MIRC building, near the eNodeBs to reduce the delay caused by propagation for real-time applications, like MAR (mobile augmented reality)[14][20]. Smart devices in the vehicle collect and send data directly to the MEC servers through the 5G RAN. The MEC servers process and send the result to their clients to display [14]. The following section analyzes several V2X applications deployed in our platform.

5.1 Computer vision for V2X Based on Deep Learning

Vehicular applications based on computer vision techniques such as object identification and classification, lane-change detection, object density estimation, and vehicle-trajectory prediction have essential roles in developing Intelligent Transportation Systems. These applications enable vehicles to understand the road conditions and the vehicle's precise position; therefore, they can be applied for different purposes like obstacle avoidance, situational awareness, driving safety, roads maintenance, etc. The traditional ML algorithms for object classification are SVM, dynamic Bayesian networks, and K-Nearest Neighbor Classifier (KNN). However, recently, they have gradually been replaced by DL models (e.g., region-based convolutional neural networks (R-CCNs)) to provide real-time detectors satisfying the requirements of autonomous driving and V2X communications [21][22]. For example, research [22] proposed a high-accuracy model using DL for counting the objects in the crowded scene like a number of the car in traffic jam scenarios; research [21] applied the faster R-CNN for real-time object detection. Notably, there are available datasets for current object detection such as

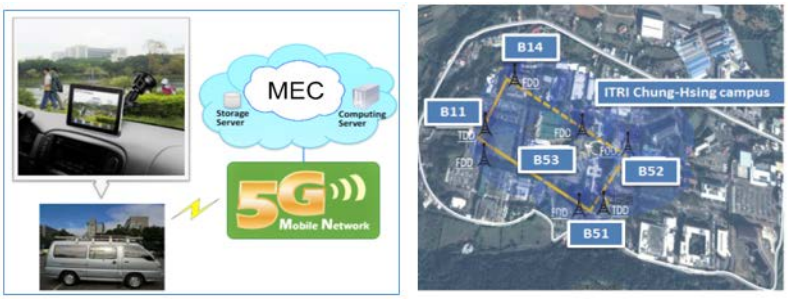

Figure 6. Open-5G MEC for V2X Application at BML/NCTU

Figure 7. Object detection using deep learning at NCTU

CNN for real-time object detection. Notably, there are available datasets for current object detection such as Microsoft COCO datasets [23], which contain large-scale object detection with more than 91 common object categories, each category has more than 5,000 labeled instances. Up to the year 2018, the dataset has 2,500,000 labeled instances in 330,000 images.

In the past few years, computer vision has been applied to the 4G/LTE network testbed at NCTU; many applications were introduced [14][24][25]. For example, in research [24], we proposed a platform to recognize tactic patterns in broadcast basketball videos. This system used Kalman filter to automatically detects the court lines, tracks the players and ball, captures and analyzes ball trajectory, calibrates the players' positions to the real-world, etc. In research [25], we developed a preliminary system called YogaST, which utilized C++ with OpenNI 1.5.4.0 and OpenCV to assist the Yoga practitioner in self-training. It aims at instructing him/her to adjust his/her posture correctly and prevents injury caused by improper postures. Notably, research [14] performed MAR for outdoor vehicular navigation applications, such as corner detection, after that it also addressed the real-time challenge and estimate the performance of 4G/LTE and 5G based on outdoor navigation system at BML. This study applies R-CCN, which is one of the most popular DL models comprising many pooling and convolutional layers that resembles the human visual system, into object detection of autonomous vehicles. Fig.7 shows a result of object detection implemented at NCTU campus. As can be seen, it identified precisely cars and pedestrians on the road, the confidences of the detected objects were also shown.

In summary, R-CCN is robust algorithms, enabling computer vision to support many useful applications for automotive driving with high accuracies like object recognition, car counting, and vehicle tracking under different traffic conditions.

5.2 Two stages slicing for V2X applications

This section analyzes and applies network slicing, which can be considered as one of the most significant innovations and evolutions in 5G architectures, to support various V2X services by providing flexible subscription models and multiple end-to-end virtual networks that share the resources of a network operator. Generally, smart devices in a vehicle usually work as multi-slice devices, that means they can simultaneously attach to multiple slices for different purposes. For example, a driver could use a self-driving service controlled by an autonomous driving slice based on V2I and V2V communications. In the meanwhile, he/she opens an HD streaming of a

Copyright © Society for Science and Education, United Kingdom

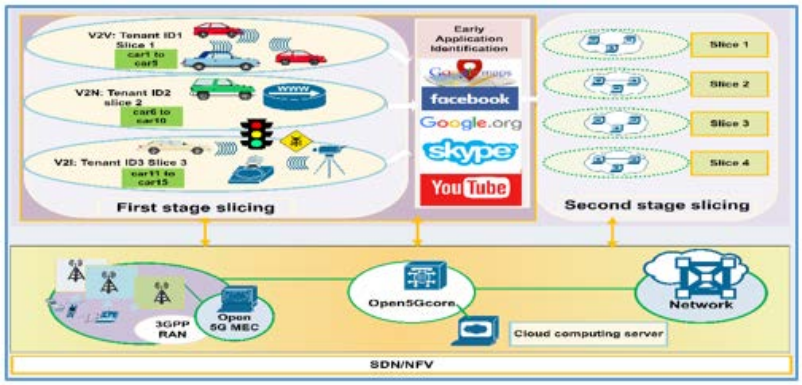

Figure 8. Two stages network slicing for V2X

different purposes. For example, a driver could use a self-driving service controlled by an autonomous driving slice based on V2I and V2V communications. In the meanwhile, he/she opens an HD streaming of a football game offered by an infotainment slice. The following subsection proposes a comprehensive framework called two-stage network slicing for V2X applications as shown in Fig.8. The first stage slices the network for different tenants or purposes, the second stage implements network slicing for various applications with different QoS requirements.

5.2.1 The first stage network slicing

Multi-tenancy is typical characteristics of V2X ecosystems, different tenants representing different services so that they should be mapped into different slices, for example, the slice for tele-operated driving and the slice for autonomous driving. These slices may be offered and managed by different service providers with diverse sets of performances and service requirements. For example, autonomous driving slices require latency less than 1ms, data rate 10Mb/s, and reliability nearly 100%, while vehicular internet and infotainment slices require latency around 10ms for web browsing, data rate 25Mb/s for UHD video streaming, and they do not concern about reliability.

In this study, we assume that there are 3 slices representing for V2V, V2N, and V2I communications working on 5G network infrastructure under the control and management of open5Gcore and open5GMEC as described in Fig.8. For example, when a vehicle detects a dangerous obstacle on the road, it will send the warning information and the location information towards all the cars around it. Firstly, the car can directly send this information to cars very nearby using V2V communication (PC5 interface). In the meantime, it passes the information through eNodeB to MEC, and then, the SDN controller in MEC platform will flood the information to cars inside its slice. In this case, the slice is defined and managed based on the distance between cars and the obstacle. That means when a car moves into the range, it will be added to the slice's list or become a slice's member until it moves out of the range.

Experimental implementation

In this experiment, we assume that there are 16 cars, car1 to car15 belong to 3 slices, each slice contains 5 cars, and car16 does not belong to any slice. They are connected together by Open vSwitches controlled by the SDN controller. An SDN application was built on the controller to handle the first stage slicing that allows only cars in the same slice to communicate with one another. The SDN application has two main functions:

The primary function is to build the network topology, calculate and install all the shortest paths between all pair of sources and destinations in the network. For example, Fig.9 shows several couples of switches in the network and the shortest paths between them. Two hosts in the network can communicate through a random route in the shortest paths or by applying a load balancing algorithm.

The second function applies network slicing policies to isolate a slice from others by checking the source and destination IP addresses of the communicating packets. If they belong to hosts (cars) of the same slice (tenant),

```
There is exactly one shortest path from switch 3 to switch 2:
3, 2,
There are 10 shortest paths from switch 3 to switch 4:
3, 1, 4,
3, 2, 4,
3, 7, 4,
3, 8, 4,
3, 9, 4,
3, 10, 4,
3, 11, 4,
3, 12, 4,
3, 13, 4,
3, 14, 4,
```

Figure. 9 Example shortest paths of couples of switches

```
*** Ping: testing ping reachability
car1 -> car2 car3 car4 car5 X X X X X X X X X
car2 -> car1 car3 car4 car5 X X X X X X X X X X
car3 -> car1 car2 car4 car5 X X X X X X X X X X
car4 -> car1 car2 car3 car5 X X X X X X X X X X
car5 -> car1 car2 car3 car4 X X X X X X X X X X
car6 -> X X X X X car7 car8 car9 car10 X X X X X X
car7 -> X X X X X car6 car8 car9 car10 X X X X X X
car8 -> X X X X X car6 car7 car9 car10 X X X X X X
car9 -> X X X X X car6 car7 car8 car10 X X X X X X
car10 -> X X X X X car6 car7 car8 car9 X X X X X X
car11 -> X X X X X X X X X X car12 car13 car14 car15 X
car12 -> X X X X X X X X X X car11 car13 car14 car15 X
car13 -> X X X X X X X X X X car11 car12 car14 car15 X
car14 -> X X X X X X X X X X car11 car12 car13 car15 X
car15 -> X X X X X X X X X X car11 car12 car13 car14 X
car16 -> X X X X X X X X X X X X X X X
*** Results: 75% dropped (60/240 received)
```

Figure 10. Ping result after applying slicing for V2X

destination IP addresses of the communicating packets. If they belong to hosts (cars) of the same slice (tenant), they are allowed to communicate. Otherwise, the packets are dropped. Fig.10 shows the ping result after acting the slicing function. As can be seen, only the cars in the same tenant can communicate with others, and since car16 does not belong to any tenant, the communicating packets between it and others are dropped. This function works as a firewall to logically isolate network functions and resources among slices.

5.2.2 The second stage network slicing

The second stage of network slicing is to guarantee QoS, QoE, user experience, and network resource efficiency for different V2X services. For example, in the slice of vehicular infotainment services, each application requires a QoS guarantee. Generally, a high-definition video streaming requires up to gigabytes per second peak data rate, small delay and jitter to provide seamless communication for end devices in a vehicle with high mobility, while other data services like web browsing, Gmail, and Google are more sensitive to packet loss. Hence, the network operator must provide different policies for them, for instance, preserve a relevant bandwidth for each application. To solve this problem, in previous research

Copyright © Society for Science and Education, United Kingdom

[17], we proposed and implemented a QoS control model for different mobile broadband applications. The model's process is described in Fig.8 and Fig.11; it involves two steps, namely, the early-state traffic flow classification and network QoS control for traffic flow using an SDN application.

Firstly, we applied several state-of-the-art classification algorithms like Naïve Bayes, Gradient Boosted Tree (GBT), Random Forest, SVM, and NN to identify traffic flows at the early stage [17]. Moreover, in this study, we applied several methods: cross-validation, parameter optimizations, and prediction fusion that combines SVM classifier and Naïve Bayes classifier to improve the performance of classification models. The result shows that all models achieved high accuracy to identify traffic flows at the early-state as summarized in Table 1.

Next, the second step utilizes the result of the traffic flow classification to build the SDN application for QoS control. Currently, SDN/NFV have been considered as emerging technologies enabling new robust methods for QoS control, such as queue management, the utilization of meter tables, ingress policing, etc. These approaches are more powerful, flexible, and easier than the traditional technologies like Integrated Service (IntServ) and Differentiated Service (DiffServ). The IntServ method is too complicated and not scalable, while DiffServ is less complicated but it does not provide strong QoS guarantees. Moreover, among the SDN-based approaches, using meter tables and QoS queue is the most popular methods. They install flow table entries for each connection to instruct Open vSwitches how to execute the flows (packets). For example, in the QoS queue approach, which is an egress packet queuing mechanism in Switch ports, each application is assigned a QoS ID in which we can set both min data rate and max data rate. Also, we can easily create, modify, delete a queue through the Open vSwitch configuration protocol like OF-Config (or ovs-vsctl command). On the other hand, the meter table method allows monitoring the ingress rate of flows by meter tables, which are attached directly to flow table entries, to measure and control the data rate of packets. Fig.11 illustrates the abstract process of implementing the second stage network slicing for uplink directions. In this framework, the classification model is deployed on the Open5GMEC close to the eNodeBs to identify and label traffic flows. When the SDN controller receives an identified traffic flow, it extracts and stores the necessary information of the connection, such as application label or type, source IP address, and destination IP address. After that, it installs flow table entries and QoS policies on OpenFlow Switch to guarantees bandwidth for that connection on both uplink and downlink directions

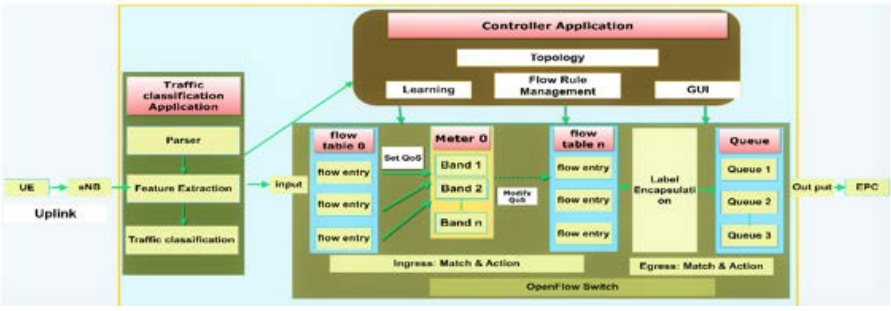

Figure 11. The second stage network slicing process

Figure 12. QoS control based on application identification

In our experiment, YouTube was preserved a bandwidth of 3Mbps in the QoS control, and then we used a UE to play a video from YouTube. Fig 12 shows the result of the test. It is clear that YouTube traffic flows were identified and classified precisely at the early state, and the bandwidth for YouTube at that time was 2.88Mbps. This bandwidth guarantees for QoS of the video to play smoothly without any dropped frame. In summary, SDN/NFV have crucial roles in creating and managing on-demand and wide-range network slicing services for V2X and 5G to build multiple logical end-to-end networks.

6 SDN/NFV Driven Service Migration for V2X

V2X communication is characterized by highly dynamic network topologies providing services for vehicles while they are moving quickly in and out of cells. The critical requirements of V2X services are not only about real-time services but also the transparent of connections on the IP layer, and TCP sessions, which are unable to be solved by applying fast-handover approaches. Moreover, a real network always encounters abnormal problems that may affect network QoS while operating due to equipment, nodes, and links failures. It becomes a big challenge for modern V2X communications because their services usually require incredibly fault-tolerant systems that can guarantee fast recovery. This section proposes a practical solution called service migration based on SDN aspects to deal with those challenges by moving a running application from a server to another one without any disruption. The process of service migration can be divided into three steps The first step: once the controller application receives a handover request from a service due to the movement of a user or due to a failure at a server called the old server, it triggers the service migration process to select an available server called migration server by using an optimization algorithm (e.g., based on load balancing).

Table 1. Accuracy of the early-state traffic classification models (%)

Application	SSL.Gmail	Amazon	WhatsApp	Facebook	Snapchat	Skype	Google	Instagram	YouTube	Apple
Naïve Bayes	90.1	89.5	93.4	91.6	87.8	94.8	85.8	87.9	94.1	93.4
SVM	93.6	91.3	95.7	94.9	92.6	97.2	90.6	92.8	97.8	94.8
SVM+Naïve Bayes	94.5	92.1	96.2	96.8	94.3	98.4	92.1	93.7	98.5	96.4
NN	93.4	94.5	97.9	95.1	96.4	97.2	93.8	97.5	99.2	98.5
GBT	98.6	98.7	99.5	98.1	99.2	99.8	99.4	99.6	100	99.4
Random Forest	98.9	99.2	99.6	98.4	99.8	100	99.6	99.2	100	99.7

After that, it calculates all the shortest paths between the client and migration servers (this function was described in the first-stage slicing section).

The second step: Delete all routing flows or table entries that relate to the old server and the current client in Open vSwitches. So that, when a switch receives a new packet from the user sent to the old server, it will ask the controller through Packet-In messages.

Copyright © Society for Science and Education, United Kingdom

The third step: Once the SDN controller receives the Packet-In message sending to the old server, it sends Packet-Out messages to set rules for redirecting all flows between the client and the old server on both directions: Modify the destination IP and MAC of flow packets that are sending from the client to old server to the IP and MAC addresses of the migration server; modify the source IP and MAC addresses of flow packets that are sending from the migration server to the client to the IP and MAC addresses of the old server.

Experimental implementation: The experimental topology and scenario are illustrated in Fig.13. Worker1 and worker3 work as operating and available servers, respectively, running the same application. Worker8 works as a client (or car) served by worker1. In this test, the worker1 was turned down, SDN application selected server3 as migration server to provide service for the client. We performed a Ping test and TCP test services to evaluate the results.

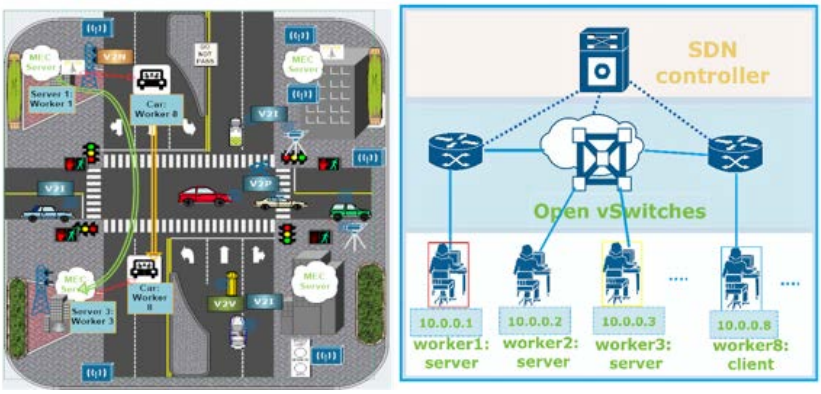

Figure 13. Experimental scenarios

Ping test result: Fig.14 shows the result of ping from worker8 to worker1. As can be seen, the client smoothly received reply messages with IP of worker1 even though worker1 was out of service. The time to receive the first message was 28.6ms much longer than the time of the following messages. It involves the time when a switch receives the first message of the connection; it must send packet-in to ask the controller, which will check the packet headers and then install the rules in table flow entries to instruct the switch how to handle this message and later messages. For more information, Wireshark was used to capture Ethernet packets at worker8 and worker3; the results are shown in Fig.15 and Fig.16, respectively. It is clear that all the messages sent from worker8 to worker1 were redirected to worker3 as shown in Fig.15; all the messages from worker3 to worker8 were modified the source IP address from those of worker3 to those of worker1 as shown in Fig.15.

TCP service test result: Fig.16 describes our experiment, worker1 and worker3 run as TCP servers listening at port 4444, worker8 run as a TCP client connecting to port 4444 of server1. The results in Fig.17 shows that the TCP service is served by worker3, not worker1

In summary, the Open5GMEC platform supports transparent migration, which is essential in V2X communication to enable more robust applications regarding resource management, service recovery, computing scaling, handover for high mobility services

```
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=28.6 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.594 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.112 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.124 ms
```

Figure 14. Ping test of MEC server migration

	Time	Source	Destination	Protocol	Length	Info
31	196.107641530	10.0.0.3	10.0.0.8	ICMP	98	Echo (ping) reply
32	197.091378489	10.0.0.8	10.0.0.3	ICMP	98	Echo (ping) request
33	197.091418934	10.0.0.3	10.0.0.8	ICMP	98	Echo (ping) reply
34	198.100084875	10.0.0.8	10.0.0.3	ICMP	98	Echo (ping) request
35	198.100109270	10.0.0.3	10.0.0.8	ICMP	98	Echo (ping) reply

Figure 15. Captured packets of the Ping test at worker3

	Time	Source	Destination	Protocol	Length	Info
37	223.073514247	10.0.0.8	10.0.0.1	ICMP	98	Echo (ping) request
38	223.102089522	10.0.0.1	10.0.0.8	ICMP	98	Echo (ping) reply
39	224.074689701	10.0.0.8	10.0.0.1	ICMP	98	Echo (ping) request
40	224.075233847	10.0.0.1	10.0.0.8	ICMP	98	Echo (ping) reply
41	225.083643535	10.0.0.8	10.0.0.1	ICMP	98	Echo (ping) request

Figure 16. Captured packets of the Ping test at worker8

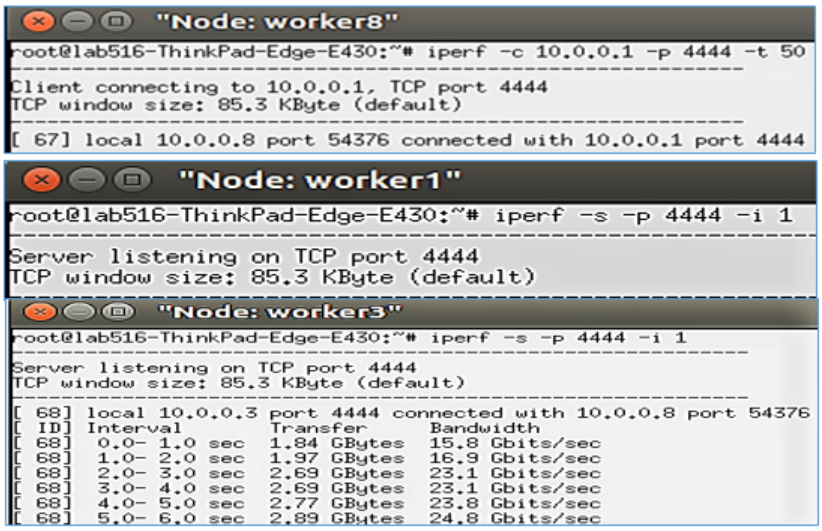

Figure 17. TCP test result at client and servers

To evaluate the computing capacity of the platform, we run the classification application (described in the first stage slicing section) with a specific configuration. Fig.18 shows the Spark computing cluster, which contains four computing workers with 32 cores and 31.3 GB memory. A stream of 210.000.000 data samples was generated with a different data rate as in Fig. 19. After receiving the data, Apache Kafka created a topic and then produced the data as the classification testing dataset to Apache Spark. After that, the incoming samples were classified by SVM classifier built from Spark MLlib 2.0. Generally, when a new input data is received, Spark streaming model continually records, processes, and updates model parameters. However, to attain better performance, this study used the mini-batch method in which the stream of data is discretized into a sequence mini-batches called sequence of RDD (Resilient Distributed

Copyright © Society for Science and Education, United Kingdom

Datasets). The mini-batch length can be defined by either a number of records or time interval. Here, we used the time duration of 3 seconds for each mini-batch. Fig. 23 shows the input and the output data rate. Fig. 20 shows the time, mini-batch sizes (number of records), scheduling time, processing and the total delay of several mini-batches that have the highest input data rate. It is

Figure 18. Spark computing cluster

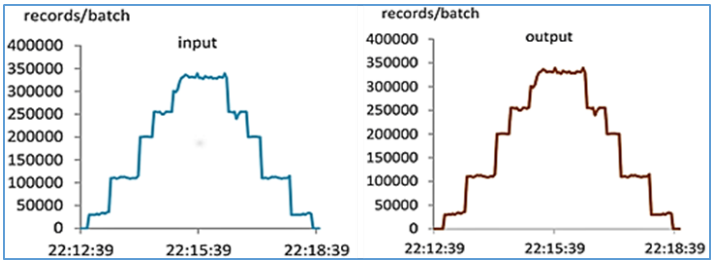

Figure 19. Input and output timelines

Batch Time	Input Size	Scheduling Delay (?)	Processing Time (?)	Total Delay (?)
2017/12/08 22:15:24	330084 records	7 ms	0.9 s	0.9 s
2017/12/08 22:15:21	334218 records	1 s	1 s	2 s
2017/12/08 22:15:18	336126 records	0 ms	4 s	4 s
2017/12/08 22:15:15	331674 records	8 ms	1 s	1 s
2017/12/08 22:15:12	329448 records	1 ms	0.9 s	0.9 s
2017/12/08 22:15:09	320862 records	0.1 s	2 s	2 s

Figure 20. Mini-batches computing timeline

clear that the computing platform is powerful in providing high computing capacity with a small time for scheduling and processing so that all the data samples were classified smoothly with low-total-latency. In summary, Open5GMEC is powerful enough to be deployed for the industrial networks

7 Conclusion

The proposed Open5GMEC integrating state-of-the-art technologies, such as SDN/NFV, big data, and ML, is expected as a comprehensive solution for future 5G MEC and vehicular communications. It supports the full power of programmability and virtualization, robust and fast computation, automatic and intelligent optimization to make network components more transparent and coordinated. Furthermore, based on the platform, this study analyzed and implemented several significant applications for V2X communication that their results are pioneers for building more complex applications in V2X and 5G areas.

Our future work will focus on applying the framework to develop other MEC and CORD components and V2X applications such as SDN-based R-CORD and M-CORD.

ACKNOWLEDGMENT

This paper is particularly supported by “the Center for 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, and the Ministry of Science and Technology of Taiwan under Grants: MOST 107-2221-E-009-056.

REFERENCES

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A Survey on Mobile Edge Computing: The Communication Perspective,” vol. 19, no. 4, pp. 2322–2358, 2017.

[2] M. C. Computing, X. Chen, L. Jiao, and W. Li, “Efficient Multi-User Computation Offloading for Mobile Edge Cloud Computing,” vol. 24, no. 5, pp. 2795–2808, 2016.

[3] S. A. A. Shah, E. Ahmed, M. Imran, and S. Zeadally, “5G for Vehicular Communications,” IEEE Commun. Mag., vol. 56, no. 1, pp. 111–117, 2018.

[4] S. Chen et al., “Vehicle-to-Everything (v2x) Services Supported by LTE-Based Systems and 5G,” IEEE Commun. Stand. Mag., vol. 1, no. 2, pp. 70–76, 2017.

[5] R. Molina-Masegosa and J. Gozalvez, “LTE-V for Sidelink 5G V2X Vehicular Communications: A New 5G Technology for Short-Range Vehicle-to-Everything Communications,” IEEE Veh. Technol. Mag., vol. 12, no. 4, pp. 30–39, 2017.

[6] X. Duan, Y. Liu, and X. Wang, “SDN enabled 5G-VANET: Adaptive vehicle clustering and beamformed transmission for aggregated traffic,” IEEE Commun. Mag., vol. 55, no. 7, pp. 120–127, 2017.

[7] H. C. Chang et al., “Empirical Experience and Experimental Evaluation of Open5GCore over Hypervisor and Container,” Wirel. Commun. Mob. Comput., vol. 2018, no. i, 2018.

[8] 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.

[9] 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.

[10] N. Cheng et al., “Big Data Driven Vehicular Networks,” IEEE Netw., vol. PP, pp. 1–8, 2018.

[11] 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), Incheon, Korea, Sep 2017.

Copyright © Society for Science and Education, United Kingdom

[12] M. Boban, A. Kousaridas, K. Manolakis, J. Eichinger, and W. Xu, "Use Cases, Requirements, and Design Considerations for 5G V2X," Netw. Internet Archit., pp. 1–10, 2017.

[13] L. Le, D. Sinh, B. P. Lin, and L. Tung, "Applying Big Data, Machine Learning, and SDN/NFV to 5G Traffic Clustering, Forecasting, and Management," IEEE NetSoft, vol. 870, no. NetSoft, pp. 168–176, 2018.

[14] 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," in Conference: IEEE VTS APWCS 2015, Singapore.

[15] 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.

[16] 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.

[17] Luong-Vy Le; Bao-Shuh Lin; Do Sinh, "Applying Big Data, Machine Learning, and SDN/NFV for 5G Early-Stage Traffic Classification and Network QoS Control," Trans. Networks Commun., vol. 6, no. 2, pp. 36–50, 2018.

[18] G. X. and H. G. and L. Q. and B. H. and K. L. and J. Wang, "Vehicle Trajectory Prediction by Integrating Physics- and Maneuver-Based Approaches Using Interactive Multiple Models," IEEE Trans. Ind. Electron., vol. 65, no. 7, pp. 5999–6008, 2018.

[19] L. Liang, H. Ye, and G. Y. Li, "Towards Intelligent Vehicular Networks: A Machine Learning Framework," pp. 1–11, 2018.

[20] 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.

[21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, 2017.

[22] D. Oñoro-Rubio and R. J. López-Sastre, "Towards perspective-free object counting with deep learning," Comput. Vis. -- ECCV 2016, vol. 9911, pp. 615–629, 2016.

[23] T. Lin, C. L. Zitnick, and P. Doll, "Microsoft COCO : Common Objects in Context," Comput. Vis. -- ECCV 2014, pp. 740--755, 2014.

[24] H. T. Chen, C. L. Chou, T. S. Fu, S. Y. Lee, and B. S. P. Lin, "Recognizing tactic patterns in broadcast basketball video using player trajectory," J. Vis. Commun. Image Represent., vol. 23, no. 6, pp. 932–947, 2012.

[25] H. T. Chen, Y. Z. He, C. L. Chou, S. Y. Lee, B. S. P. Lin, and J. Y. Yu, "Computer-assisted self-training system for sports exercise using kinects," 2013 IEEE Int. Conf. Multimed. Expo Work., pp. 3–6, 2013.