GANSlicing: A GAN-Based Software Defined Mobile Network Slicing Scheme for IoT Applications

Ruichun Gu*†, *IEEE Student Member*, Junxing Zhang*, *IEEE Member*
*College of Computer Science, Inner Mongolia University, Hohhot, China
†Inner Mongolia University of Science and Technology, Baotou, China
reachcool@imust.cn, junxing@imu.edu.cn

***Abstract*—With the rapid development of the mobile network and growing complexity of new networking applications, it is challenging to meet the diverse resource demands under the current mobile network architecture, especially for IoT applications. In this paper, we propose GANSlicing, a dynamic service-oriented software-defined mobile network slicing scheme that leverages Generative Adversarial Networks (GANs) based prediction to timely and flexibly allocate resources for IoT applications and to improve Quality of Experience (QoE) of users. Compared with the current tenant-oriented mobile network slicing scheme, GANSlicing is able to accept 16% more requests with 12% fewer resources for the same service request batch according to our evaluation. The result demonstrates that the proposed scheme not only improves the utilization of resources but also enhances the QoE of IoT applications.**

***Index Terms*—GANs; Slice; IoT; 5G; SDN; NFV**

I. Introduction

With the rapid development of the cellular network and mobile computing, more and more networking applications with high complexity emerge. It is a big challenge to meet the diverse resource requirements of these applications under the current mobile network architecture. According to the estimation of EU FP7, the number of connected devices in the cellular network will increase 100 times and the mobile data volume will soar 1000 times by 2020. The exponential growth of mobile devices, applications and data prompt the evolution of the mobile network [1].

Fortunately, the 5th generation (5G) mobile network is designed with the faster data rate, higher bandwidth, lower latency and seamless handover for all the connections, which is supposed to overcome the shortcomings of the current mobile network and meet future demands. For IoT (Internet of Things) applications, smart things or devices should be connected in a reliable manner to offer different services. Furthermore, with the concept of network slicing, 5G has the ability to provide services that have more complex resource requirements.

Network slicing is a cutting-edge technology of 5G that is able to create multiple types of virtual networks tailored to accommodate different requirements. However, in the carefully observe, we can find out that the at the same time, different slices always have different resource demands, and the requests of one slice during its lifetime always changing, but the resources provision cannot be modified dynamically. Therefore, two problems in the mobile network slicing are exposed. The first is the lack of flexibility in resource allocation, which often leads to low resource utilization. The second is resource allocation lags behind the time users need resources resulting in the poor Quality of Experience (QoE).

For example, in the Internet of Medical Things (IoMT), a glucose monitoring device uploads the measured data in a periodical manner when the blood sugar is stable and within the threshold. A network slice with only a few resources is needed in this scenario. However, once the monitored value fluctuates or exceeds the threshold, more detailed data must be uploaded in time. A slice that offers higher bandwidth and lower latency is in need at that time. If the patient wants to consult the doctor via a video stream or receive the remote treatment, a slice that ensures the best network condition might be in order. After the sugar level returns to the normal range, the resources in the network slice should be restored to the low-performance level for better global utilization. In these situations, the network slicing scheme must be able to allocate resource flexibly and timely.

In order to improve the flexibility and timeliness of the mobile network slicing, we consider three design issues. The first issue of designing the slicing scheme is a global perspective of the whole network resources, including both physical and virtual resources of the cellular network. With the central control of all resources, networking, computing, as well as storage devices can be managed and utilized more efficiently and flexibly. The second issue we consider is to dynamically allocate resources for network slices in a service-oriented manner. An orchestrator allocates resources in a dynamic way for all the slices to provide different classes of services. The third issue is to predict user requirements for different resources with an advanced deep learning model. If resources in network slices can be tailored and allocated automatically before requests arrive, the QoS of services and QoE of users will be enhanced and the utilization of resources will be improved.

With the recent advances of deep learning, Generative

This work was partially supported by the National Natural Science Foundation of China (Grant No. 61261019), the Inner Mongolia Science & Technology Plan, and the Inner Mongolia Autonomous Region Natural Science Foundation (Grant No. 2018MS06023).

Corresponding author: junxing@imu.edu.cn

Adversarial Networks (GANs) [12] has shown superior performance on complex data modeling directly from raw data. It develops two effective models (i.e., Generator and Discriminator), after training, GANs can perform inference by simply once feed-forward computation, which is generally fast and also possible to be distributed for large-scale inference.

In this paper, we address the problems of the current network slicing scheme for the mobile network in a dynamic and forecasting way using GANs. Our proposed scheme adopts and extends the techniques of Deep Learning, Software Defined Network (SDN), and Network Function Virtualization (NFV). The main contributions of the paper are:

1) A service-oriented slicing scheme that serves different requests with various classes of slices that have suitable resources allocated dynamically;
2) A deep learning model that leverages GANs to predict the resources requirements of users in a dynamic and timely manner.

The rest of this paper is organized as follows. Section II discusses the related work. Section III outlines the proposed GANSlicing architecture. Section IV presents the implementation details. We describe the evaluation process and experimental results in Section V. The whole paper is concluded in the last section.

II. Related Work

There is some research on the challenges of mobile network slicing. J. Ordonez-Lucena [2] proposed a comprehensive survey on the network slicing for 5G with SDN and NFV, which figured out that the resources management mechanisms without violating the performance are required. Thus some efficient resource allocation algorithms are needed for network slicing. Paper [3] proposed that the stricter intra-slice separation would require more bandwidth, and more stringent E2E deploy requirements impact resource utilization and request acceptance rate. JOX [4] is an event-driven network slicing orchestrator, which is implemented with the jujucharms [5]. JOX supportes virtualized service chains, could be deployed on KVM or LXC.

Data-driven or machine learning is playing an increasingly important role in the mobile networking domain. D. Adami [6] introduced an OpenFlow-based Virtualization-aware Networking (OFVN) platform for the network resource control in the Cloud DC environment, which is based on the traffic load over DC links to place the VMs into physical services leveraging flow-based OpenFlow statistics. V. Sciancalepro [7] proposed a solution for mobile slicing traffic forecasting with a network slicing broker system based on the Holt-Winters theory, which could leverage the resource utilization for different slices. M. Jiang [8] proposed an auction-based model for the network resource allocation in network slicing, with a higher satisfaction of each slice and increase the revenue in the perspective of network operators. DeepRM [9] is presented for the resource allocation in the data center, which translates the problem of packing tasks with multiple resource demands into a deep reinforcement learning problem. B. Ko [10] presented a machine learning based method for dynamic resource management problem in edge computing. These two kinds of literature are similar to our work, but they are focused on the data center and edge computing, while we are interested in the mobile core network. C. Zhang [13] had summarized both basic concepts, and advanced principles of various deep learning models then correlated the deep learning and mobile networking disciplines by reviewing work across different application scenarios. They had also discussed how to tailor deep learning models to general mobile networking applications.

Other papers [14], [15] also similar to our work are that C. Zhang combines Zipper Network and GAN models in the mobile traffic inferring, and Deep Spatio-Temporal Neural Networks in the Long-Term Mobile Traffic Forecasting. Deep-Q [16] is a data-driven QoS inference system, aimed to model the QoS distribution in real networks without human analysis. in [21], a decentralized resource allocation mechanism is proposed for the V2V communications based on deep reinforcement learning, where each V2V link is regarded as an agent, making its own decisions to find optimal spectrum and power for transmission. The V2V communications always demand uRLLC network, which is one of the three main scenarios of 5G network, the requirements do not changed, and the motivation of our work is to meet the dynamic change of diversity network requirements. As mentioned in the working group 3 of FG-ML5G (Machine Learning for Future Networks including 5G) in ITU-T [22], whose ultimate objective is to enable efficient use of ML technologies in future networks, which is also similar to our work.

Reviewing the current related literature, we can observe that schemes of the deep learning based dynamic slicing in mobile networks are emerging. Additionally, only a few kinds of research proposed the concept of dynamic resource allocation and little for the cellular network. In this work, we propose GANSlicing, a framework focuses on the resource orchestrating of mobile network slicing on-demand of the services requests prediction based on GANs, and introduce a service-oriented slicing architecture adopting the concept of SDN and NFV for the next generation (e.g. 5G) mobile network.

III. GANSlicing Architecture

A. Components of GANSlicing

As depicted in the figure 1, there are four layers in the GANSlicing architecture; they are Data Layer, Control Layer, Management Layer, and the Prediction Layer from bottom to top.

There are two boxes in the Data Layer, the switches, and core components. The switches are in charge of the traffic forwarding according to the flow-tables. Of course, the SDN controller has a global control of these switches through the OpenFlow protocol. And the slices here are the logical network topologies. In the components box, the virtualized core components such as vMME, vSGW, vPGW, vHSS, are running in the LXC containers [11]. The same as the switches, the virtualized components here also sliced for kinds of services. All the virtualized EPC components running in the LXC are customed, each slice contains one or more groups of the EPC components with different resources quotas tailored for different services. The IT resources (e.g., CPUs, Memory) will be configured to the components dynamically according to the requests of the application.

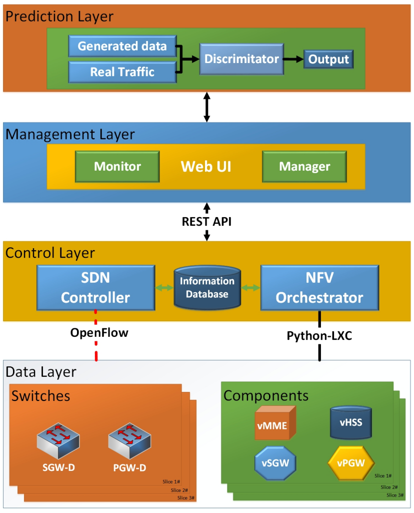

Fig. 1: The Architecture of The GANSlicing.

In the control layer, there are three functional components, the controller, the orchestrator, and an information database. The SDN controller is in charge of controlling the switches and monitoring the traffic status through the OpenFlow protocol, collecting the network statistics into the information database. The database is storing two kinds of data, that is the network resources usage, and total quota, as well as the status of IT resource utilization, and will be updated by the controller and the orchestrator when a slice is created or destroyed, or sometimes the dynamically modify of the resources of the slices.

The NFV orchestrator is in charge of the LXC containers. The same as the SDN controller, the NFV orchestrator has two functions, namely the monitoring and orchestrating of the virtualized components. All the status of resources utilization will be collected and inserted into the database. Furthermore, all the components are managed by the orchestrator, and slices with different requirements will be allocated with suitable resource quota on setup or during runtime.

The monitor of the management layer will collect the network features of the slices and feed to the prediction layer, then the output will send to the manager of the management layer. Administrators could observe and supervise the whole mobile network slices through the Web UI in the management layer. A RESTFul API is used to communicating with the control layer.

In the prediction layer, the GANs are implemented for traffic forecasting. The generator and the discriminator are two principal models of GANs. After the networks are trained with the observed resource and traffic data both from the controller and the orchestrator, the generator could generate a prediction for the further service demand through the random noise input. The discriminator determines whether the traffic data is real or generated. After some epochs, the resources will be allocated automatically according to the inferred traffic output with little wasted.

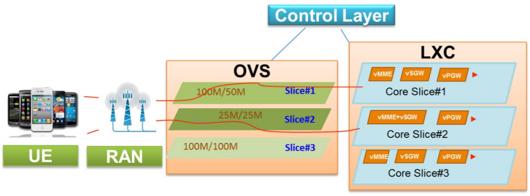

Fig. 2: The Real Process of GANSlicing Architecture

B. Slice Process

Once there is a new network service request arrived, if no matching item of the flow table in the switches, the traffic head will be forwarded to the SDN controller, the controller will initiate a new slice with meter configurations according to the requirements. At the same time, the NFV orchestrator will also setup new EPC components for this new slice. As the figure 2 plots, the slices in switches will be created with different UL/DL bandwidth, and the slices of components will be created with suitable IT resources. The figure 2 shows two different requests are served by two different slices with tailored resources. The $slice\#1$ with 100M/50M network bandwidth, and the dedicated components (e.g. vMME, vSGW and vPGW), which for the huge data traffic with low latency, e.g., the video stream in the remote surgery operation; And the $slice\#2$ with 25M/25M network bandwidth, is initialed with combined virtual components (e.g., vMME+vSGW and vPGW), could be used for the healthcare sensors on wearable devices.

Before creating a new slice, or if a running slice of service requests more resources, the controller and orchestrator of the control layer will make decisions based on the output of prediction layer, and assign the resources to the slice under the constraints of the available resources. The system could collect the free resources when a slice is destroyed or degraded, and the allocation will be updated with a new value. We could regard these as two resource pools, used for traffic forwards and the core network components, respectively.

C. Service-Oriented Slicing

In the GANSlicing framework, we propose a concept of Service-Oriented Slicing(SOS). All the slices are created for different services requests.

When a new service request is arriving, a new slice will be initialized with suitable resources tailored, of course, the functions are isolated between slices. Nevertheless, one E2E slice will not occupy the resources for its whole lifetime, the resource allocation will be varied along with the requests. As shown in the figure 3, the $slice\#1$ is served for the medical wearable devices IoT network, in this scenarios, some sensors of the devices will upload data to the server sporadically, which do not need high-bandwidth, low-latency

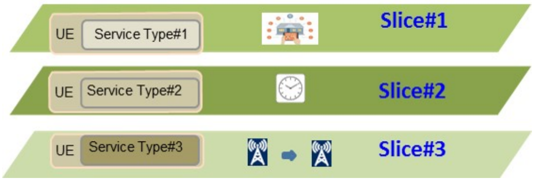

Fig. 3: The Service-Oriented Slicing

network. However, the demand will be changed when some data value is over the threshold. As a result, the resources allocation of this slice should be flexibility. The $slice\#2$ is serving the mobile live video streaming application, it needs high-bandwidth and low-latency configuration, so as the slice should be initialized with more resources. The $slice\#3$ is a Device-to-Device (D2D) network over the mobile network, and this service could be satisfied with a little resource allocation for control signalling.

All the slices are created for different services could be modified flexibility, agility and automatically in the framework of GANSlicing. The SOS mechanism not only satisfies the various network requests but also could assign resources efficiently.

D. Demand Prediction

As shown in the figure 1, the role of the prediction layer is forecasting the slice demand for decision making of the resource orchestration. The GANs is adopted in the system for automatic slicing. The statistical historical data is used to train the networks, and the forecasted demand will be generated by the generator network fed with the random noise data, the lower layers of the system will get the output of the converged adversarial networks after iterations. The slice features will be configured according to the output of the prediction layer.

With the help of prediction, all the configurations will pre-worked before the demand occurred, not only the user QoE will be guaranteed, but also the resource utilization will be improved.

IV. Implementation Of GANSlicing

We implement the GANSlicing in two parts, the first part is the SOS and the second part is the GAN-Based prediction.

A. Service-Oriented Slicing

In the implementation of the SOS, we adopted SDN and NFV, the traffic forward in the data plane we used SDN, and the components of EPC, such as vMME, vHSS, vSGW, and vPGW are implemented by NFV. The information database is implemented using MySQL.

1) Slicing on Data Plane: Programmability is an essential feature in the next generation mobile networks, and SDN is the technology to realize it. In GANSlicing, the SDN controller is developed based on the RYU controller, and the data plane switches are implemented by the OpenVSwitch (OVS). All the traffic flows are forwarded through the OVSes according to the flow tables managed by the controller.

VxLAN (Virtual Extensible LAN) is a current leading technology to realize network slicing within the mobile network, which is employed to slice the network in GANSlicing. VxLAN is a protocol that encapsulates the OSI layer 2 Ethernet frames within the layer 4 UDP datagrams. Each encapsulation can be considered as a VxLAN segment or a tunnel constructed using MAC in UDP. VxLAN endpoints that terminate VxLAN tunnels can be either virtual or physical network interfaces.

2) Slicing on EPC Components: Virtualization is another character of GANSlicing, virtual network functions can be leveraged in a quite straightforward manner to deploy and manage network slices dynamical and flexible. An orchestrator of the management layer to orchestrating the virtual core components elastically.

According to the services demand, the core components will be combined in chains to meet the requests of one or more service slices, the resources will be allocated for different services by the orchestrator on demand, and there will be several function chains for multiple slices.

The EPC components of GANSlicing are hosted in Linux Containers (LXC), through a robust API and simple tools, users could easily create and manage the virtual components. The resources allocated to different function chains could be configured by the Control Groups (Cgroups) elastically. Cgroups are kernel features that allow fine-grained control over resource allocation for a single process, which makes it possible to assign limits to memory, CPU time, or I/O [19]. The libvirt is a great API to create and manage the applications in LXC programmatically. The orchestrator will orchestrate the resources of the virtualized slice function chains through a python-lxc interface.

B. GAN-Based Slicing Forecasting

GANs have shown remarkable success for training models to produce realistic-looking data. GANs is a model made up of two entities: a generator and a discriminator. Here we will consider them both to always be parameterized neural networks: G and D, respectively.

1) Problem Statement: In high-speed mobile network environments, an accuracy one-step forecasting for the resources demand is usually of great concern to network managers to provide significant information for QoE guarantee. We aim to forecast the network resources fluctuation of slices, based on their historical information of several features. As the resources requirements are closely related to the network traffic fluctuation, and the slice resources demand always changing against the traffic load, we could infer the traffic load of different slices for the resources assignment.

We decide to use the network metrics such as downlink rate, uplink rate and delay for the traffic load prediction. Let DL_t represent the downlink bitrate, UL_t represent the uplink bitrate and D_t represent the network delay indicate the network requests of a slice at time t. Let $X_t = \{DL_t, UL_t, D_t\}^T$ represent the set of basic indicators of a slice at time t. Let Y_t denote the slice metrics of one slice for a 1-minute interval at time $t(t = 1, 2, ..., T)$, where T is the maximum lag of time. Given the random basic indicators information $X(X = \{X_1, X_2, ..., X_T\})$ and the historical slice metrics

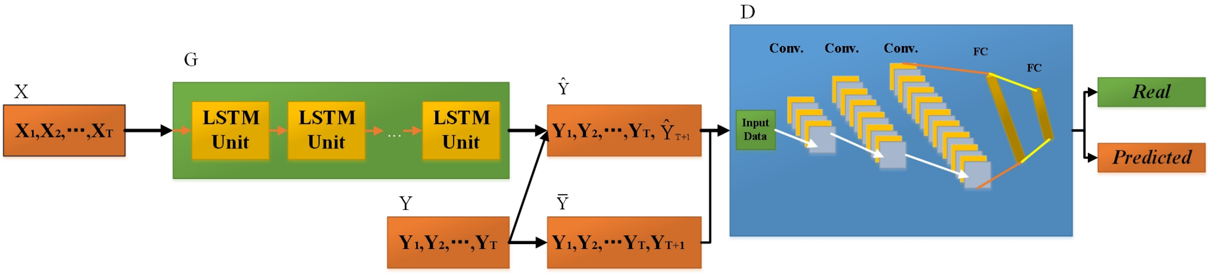

Fig. 4: The GANSlicing structure. The Generator(G) is founded on LSTM, which applies to predict $\hat{\mathrm{Y}}_{T+1}$. The discriminator (D) is based on CNN for the purpose of estimating the probability of whether an input is real ($\bar{\mathrm{Y}}$) or being predicted ($\hat{\mathrm{Y}}$). $Conv.$ means convolutional layer, FC is the fully connected layer.

$Y(Y = \{Y_1, Y_2, ..., Y_T\})$, our goal is to predict the slice metrics Y_{T+1} for the next 1-minute time interval. As there are 1440 minutes in one day, we set T to 1440 in this paper.

2) Prediction Model: The architecture of GANSlicing is illustrated as in figure 4. Since the network traffic data is a typical time series, we choose LSTM model as the generative model G to predict output $\hat{Y}_{T+1}$ based on the input data X.

$$\hat{Y}_{T+1} = G(X) \tag{1}$$

LSTM is an underlying deep learning model and capable of learning long-term dependencies. LSTM work tremendously well on various problems, such as natural language text compression, handwriting recognition, and electric load forecasting.

The discriminative model D is based on the CNN architecture and performs convolution operations on the 3-dimensional(e.g. DL_t, UL_t and D_t) input data comes from the real dataset $\bar{Y} = Y_1, Y_2, ..., Y_T, Y_{T+1}$ or being produced by a generative model $G(\hat{Y} = Y_1, Y_2, ..., Y_T, \hat{Y}_{T+1})$.

CNN is a class of feed-forward, a deep artificial neural network that has successfully been applied to analyzing visual imagery. CNN is also excellent in areas such as image and video recognition, recommender systems, and natural language processing [17].

Our primary motivation for using an adversarial network is that it can mimic the network operation of the administrators. An experienced manager usually predicts traffic flow through the available historical information, which is the work of the generative model G, and then judges the near probability of the forecast with the previous network traffic, as the discriminative model D does.

3) Models Training: The training of the pair(G, D) consists of two alternated steps. For clarity, we use the pure Stochastic Gradient Descent(SGD), it is easy to generalize the algorithm to mini-batches of size K by adding the losses over all the samples.

Training G(let(X, Y) be a sample from the dataset). To make the discriminative model D as "confused" as possible, the generative model G should reduce the adversarial loss as more as possible in the sense that D will not discriminate the prediction correctly. Classifying $\bar{Y}$ into class 1 and $\hat{Y}$ into class 0, the adversarial loss for G, namely L^G_{adv} is:

$$L^G_{adv}(\hat{Y}) = L_{sce}(D(\hat{Y}), 1) \tag{2}$$

where L_{sce} is the sigmoid cross-entropy loss, defined as

$$\begin{aligned} L_{sce}(A, B) = -\sum_i B_i(sigmoid(A_i)) \\ + (1 - B_i)\log(1 - sigmoid(A_i)) \end{aligned} \tag{3}$$

Algorithm 1 Training of the G and D

```
Set the learning rates $\gamma_D$ and $\gamma_G$,and parameters $\alpha_{adv}, \alpha_p, \alpha_{dpl}$;
Initialize weights $W_D$ and $W_G$
while not converged do
    Update the Generator $G$:
    Get $K$ new data samples$(X^1, Y^1), (X^2, Y^2), ..., (X^K, Y^K)$
    $W_G = W_G - \gamma_G \sum_i^K \frac{\partial L_G(X^{(i)}, Y^{(i)})}{\partial W_G}$
    Update the discriminator $D$:
    Get $K$ new data samples$(X^1, Y^1), (X^2, Y^2), ..., (X^K, Y^K)$
    $W_D = W_D - \gamma_D \sum_i^K \frac{\partial L_D(X^{(i)}, Y^{(i)})}{\partial W_D}$
end while
```

However, in practice, only minimizing adversarial loss cannot guarantee satisfying inferring. That G could generate wrong samples to "confuse" D, without being equal to $\hat{Y}_{T+1}$, and then D will learn to discriminate these wrong data, leading G to generate wrong "confusing" samples, and so on. To address this issue, the generative model G should decrease the error loss; which is the L_p loss:

$$L_p(\bar{Y}, \hat{Y}) = \|\bar{Y} - \hat{Y}\|_p \tag{4}$$

where p=1 or p=2.

As mentioned before, mobile traffic prediction is significant to network resource allocation and guarantee of the QoE, thus we also define the direction prediction loss function L_{dpl} below:

$$L_{dpl}(\bar{Y}, \hat{Y}) = |sgn(\hat{Y}_{T+1} - Y_T) - sgn(Y_{T+1} - Y_T)| \tag{5}$$

the sgn is the sign function.

Then, combining all the losses defined above with suitable parameters α_{adv}, α_p, and α_{dpl}, the final loss on G is defined as:

$$\begin{aligned} L_G(X, Y) = \alpha_{adv} L^G_{adv}(\hat{Y}) + \alpha_p L_p(\bar{Y}, \hat{Y}) \\ + \alpha_{dpl} L_{dpl}(\bar{Y}, \hat{Y}). \end{aligned} \tag{6}$$

Next, we perform an SGD iteration on G to minimize the $L_G(X, Y)$ while keeping the weights of D is fixed.

Training D (let (X, Y) be a different data sample). Since the role of D is to determine whether the input 3-dim data is Y or $\hat{Y}$, the target loss is equal to the adversarial loss on D. While keeping the weights of G is fixed, we perform one SGD step to achieve a minimal target loss, $L_D(X, Y)$:

$$\begin{aligned} L_D(X, Y) &= L^D_{adv}(\bar{Y}, \hat{Y}) \\ &= L_{sce}(D(\hat{Y}), 0) + L_{sce}(D(\bar{Y}), 1). \end{aligned} \tag{7}$$

We train the generator and discriminator iteratively, which is summarized in Algorithm 1, with mini-batches of size K.

4) Training Dataset: In order to emulate a real IoT network, the training dataset we used is downloaded from The Mobile and Internet Systems Laboratory in the Department of Computer Science at University College Cork, which provides a 4G trace dataset composed of client-side cellular key performance indicators (KPIs) collected from two major Irish mobile operators, across different mobility patterns (e.g., static, pedestrian, car, tram and train) [20].

V. Performance Evaluation

A. Emulation Environment

We use the virtualized environment based on the VirtualBox to implement the emulation environment. The implementation of the mobile traffic forecast is using Tensorflow with 2 hidden layers, running backend.

1) SOS with different SLA: We implemented a simple CoAP client using the CoAPthon library. 20000 clients are generating CoAP requests mimicking body status sensor devices sending different data with variance requests, thus all separated E2E slices are followed with dedicated Service-Level Agreements(SLAs). There are three use cases we designed for the experiments:

a) Video Medical Consultation: Double direction live stream (UL/DL). We call it the Service Type 1 ($ST1$), which needs high throughput (25M) and low latency;

b) Virtual Treatment: Single direction live stream (DL). We call it the Service Type 2 ($ST2$), which needs medium throughput (15M) and low latency;

c) Data Submit: Single direction data submit (UL). We call it the Service Type 3 ($ST3$), which needs low throughput (10M) do not sensitive with latency.

2) Resource allocation: There are two different situations about the slice resources management, the first one is when a new service demand arrived, a new slice with resources tailored to meet the request for the guarantee of QoE will be created; the second one is after the slice is established, the resource usage will be monitored, and the following requests will be forecasted, the resources will be scaled up or down to satisfy the requests.

B. Experiments and Results Analysis

We have conducted experiments to evaluate the performance of the proposed dynamic Service-Oriented Slice (SOS) and the GAN-Based traffic inference architecture, named GANSlicing. In the first, we conducted an experiment to evaluate the accuracy of the network traffic prediction, and then we analyzed the performance of the dynamically slicing based on services requirements, and the network metrics, such as service acceptance ratio and the network bandwidth utilization, which are compared with the prototype of Tenant-Oriented Slice(TOS), the most prevalent slicing scheme in mobile networks [23]. All the Experiments are repeated 100 times, the requests traffic are created randomly, and we calculate the average value.

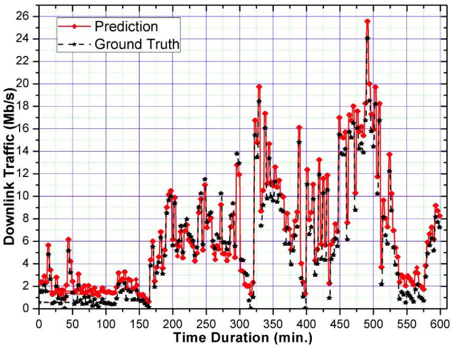

Fig. 5: The Prediction of The Mobile Traffic

1) GAN-Based Traffic Inferring: In this experiment, the accuracy of traffic inferring based dynamic slicing is conducted after a training of 24 hours, and the results are analyzed for better resource utilization. First of all, we evaluate the precision of predicting the mobile network traffic for the resource pre-allocation to meet the requirements dynamically based on the GANs.

The figure 5 shows the results of the downlink mobile traffic prediction during 600 minutes. We can observe the accuracy of the forecast is very close to the ground truth traffic, despite few oscillating behaviors of the traffic. We conducted a one-step-ahead prediction here, which means that we can use the history values to predict the traffic for the next time slot precisely. Furthermore, we change the time scale from minutes to seconds, the prediction results and the mean prediction error is very similar to figure 5, which not described due to space limits.

Therefore, with the accuracy traffic inferring, we could allocation the network resources preveniently to different application requests for a better QoE. In our architecture, all the resources are under control of the control layer, and when slices are destroyed or degraded, the controller cloud allocates the free resources of them to meet other new requirements in-the-fly exactly.

2) Service-Oriented Dynamic Slicing: We conducted experiments on fine-grained flexibly Service-Oriented Slicing (SOS) of the GANSlicing, compared the service acceptance ratio and the resource utilization (e.g., network bandwidth) with the concept of the Tenant-Oriented Slice (TOS).

We assume that there is a network with 150Mb bandwidth, and three tenants based slices allocated with each quota of 50Mb broadband. In ordinary, slices not always be saturated, but the assigned resources could not be changed or shared with others. As described in table I, in the tenant A (TA), which is fulfilled by three type of services; In the tenant B (TB), the service type 1 ($ST1$) and service type 3 ($ST3$) are running, wherein tenant C (TC), the $ST1$ and $ST2$ are running.

If we slice this network based on tenants, allocate the all 150Mb to three horizon slices averagely, and could not orchestrate the free network resources between slices. However, there is a revolutionary slice scheme in the GANSlicing, which is slicing based on services vertically, all the network resources are allocated agility to meet the requirements of different services. As described in table I, in the vertical slices of service types, the total network bandwidth is 125Mb; hence the multi-service slice could provide more 25Mb for other applications. As a result, the service acceptance could be of more than 16% compare to the multi-tenant slice.

TABLE I: The Slice on SOS vs. TOS

Type	***ST 1***	***ST 2***	***ST 3***	***Quota***
TA	**25**	**15**	**10**	**50**
TB	**25**	**-**	**10**	**50**
TC	**25**	**15**	**-**	**50**
Total	**75**	**30**	**20**	**-**

Furthermore, in order to evaluate the user QoE, we have analyzed the acceptance ratio of the services requests and compared the results between these two kinds of slice prototypes. As the figure 6 depicted, in the GANSlicing, the acceptance rate is always higher than the TOS type after the demand bandwidth is reached to 120 Mbps, and the GANSlicing could keep 100% accept before 200 Mbps. It can be concluded that GANSlicing is outperforming that of TOS, which indicates a better QoE.

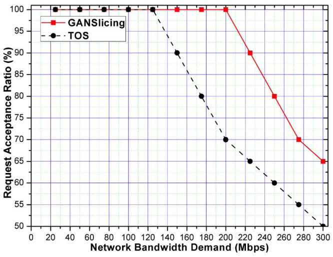

Fig. 6: The Comparison of Acceptance Ratio

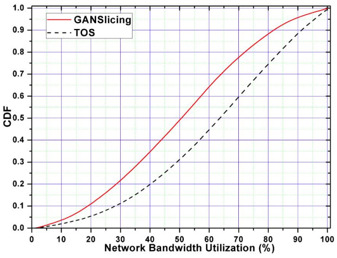

Fig. 7: The Comparison of CDF

At last, we compared the Cumulative Distribution Function (CDF) of the network resources utilization between GANSlicing and TOS. It could be observed from the figure 7 that the GANSlicing always utilizes fewer network resources than TOS. Between the 30th and 80th percentile of the CDF, the network bandwidth utilization of GANSlicing is relatively lower about 12% than the TOS scheme. The difference is mainly since the GANSlicing could allocate the resources dynamically under an accuracy demand prediction for vertical services requests, but the resources of TOS could not be managed elastically.

VI. Conclusion

In this paper, we proposed GANSlicing, a dynamic service-oriented software-defined mobile network slicing framework based on GANs to predict the resource demands of IoT applications, aiming to optimize the resource utilization and improve user QoE. GANSlicing is targeted to evolve the next generation mobile network for higher performance of the vertical industries with better fine-grained slicing with the dynamic resource orchestrate. In accordance with evaluation, GANSlicing not only leverages the utilization of the constrained network resources, improves the service acceptance ratio, but also enhances the service quality of new services in IoT applications.

In the future, we plan to compare the prediction accuracy of GANSlicing with other time-series prediction approaches, such as LSTM and ARIMA. We are also interested in adopting Multi-access Edge Computing (MEC) in IoMT applications. Because the Service-Oriented Slicing categorizes similar services into one slice, the performance could be improved through edge computing. Moreover, the computation cost in the training or prediction stage may also be optimized in future work.

References

[1] Strm, E.G. and T. Svensson, EU FP7 INFSO-ICT-317669 METIS, D1.1 Scenarios, requirements and KPIs for 5G mobile and wireless system. Scenario, 2013.

[2] Ordonez-Lucena, J., et al., Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges. IEEE Communications Magazine, 2017. 55(5): pp. 80–87.

[3] Sattar, D. and A. Matrawy, Optimal Slice Allocation in 5G Core Networks. 2018.

[4] Katsalis, et al., JOX: An event-driven orchestrator for 5G network slicing. 2018.

[5] Juju VNFM Framework. https://jujucharms.com/

[6] Adami, D., et al., Effective Resource Control Strategies using OpenFlow in Cloud Data Center. 2013: pp. 568–574.

[7] Sciancalepore, V., et al., Mobile traffic forecasting for maximizing 5G network slicing resource utilization. IEEE INFOCOM 2017

[8] Jiang, M., et al., Network slicing in 5G: An auction-based model. 2017 IEEE International Conference on Communications (ICC), pp. 1–6.

[9] Mao, H., et al., Resource Management with Deep Reinforcement Learning. ACM Workshop on Hot Topics in Networks, pp.50–56. 2016

[10] Ko, B.J., et al, Machine learning for dynamic resource allocation at network edge. in Proc. of SPIE. 2018.

[11] Ivanov, K., Containerization with LXC: get acquainted with the world of LXC. 2017.

[12] I. Goodfellow, et al., Generative adversarial networks, in Advances in neural information processing systems, 2014, pp. 26722680.

[13] Zhang, C., X. Ouyang and P. Patras, ZipNet-GAN: Inferring Fine-grained Mobile Traffic Patterns via a Generative Adversarial Neural Network, ACM CoNEXT. 2017.

[14] Zhang, C., P. Patras and H. Haddadi, Deep Learning in Mobile and Wireless Networking: A Survey. ArXiv, 2018.

[15] Zhang, C. and P. Patras, Long-Term Mobile Traffic Forecasting Using Deep Spatio-Temporal Neural Networks. arXiv, 2018.

[16] Xiao, S. He, D. and Gong Z. Deep-Q: Traffic-driven QoS Inference using Deep Generative Network. ACM SIGCOMM 2018

[17] Zhou X, Pan Z, Hu G, et al. Stock Market Prediction on High-Frequency Data Using Generative Adversarial Nets. Mathematical Problems in Engineering, 2018(1):1-11.

[18] Bao W, Yue J, Rao Y. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. Plos One, 2017, 12(7).

[19] Ivanov, K. Containerization with LXC. Birmingham: Packt Publishing. 2017.

[20] Beyond Throughput: a 4G LTE Dataset with Channel and Context Metrics, "http://www.cs.ucc.ie/ dr11/4G_Dataset/",2018

[21] Ye, H., Li, G.Y. Deep Reinforcement Learning for Resource Allocation in V2V Communications. 2018 IEEE International Conference on Communications (ICC),1-6.

[22] ITU-T, Focus Group on Machine Learning for Future Networks including 5G, "https://www.itu.int/en/ITU-T/focusgroups/ml5g/Pages/default.aspx", 2019

[23] Garces, P.C., et al, Network slicing games: Enabling customization in multi-tenant networks. INFOCOM 2017, 1-9.