Machine Learning-Based Framework for Autonomous Network Management in 5G Systems

Wei Jiang[*][†], Mathias Strufe[*], and Hans D. Schotten[†][*]
[*] Intelligent Networking Group, German Research Center for Artificial Intelligence (DFKI)
Trippstadter street 122, Kaiserslautern, 67663 Germany
Emails: {wei.jiang, mathias.strufe, hans.schotten}@dfki.de
[†] Institute for Wireless Communication and Navigation, University of Kaiserslautern
Building 11, Paul-Ehrlich street, Kaiserslautern, 67663 Germany
Emails: {wei.jiang@dfki.uni-kl.de, schotten@eit.uni-kl.de}

***Abstract*—To meet the radical technical requirements specified by ITU-R IMT-2020, the fifth Generation (5G) mobile system will become more complicated and heterogeneous. It imposes a great challenge on today's network managing approaches, which are already costly, vulnerable, time-consuming and therefore inapplicable to the 5G system. By applying machine learning, a possibility on autonomically self-organizing 5G networks is opened. With minimal human interventions, autonomic management can lower operational expenditure, improve user's experience and shorten time-to-market of new services. In this paper, the concept of intelligence slicing, a flexible and scalable framework for applying machine learning to enable self-organizing 5G networks, is proposed. The life-cycle management of intelligence slices, as well as intelligence domain that is defined as the effective area of a slice, are discussed. Moreover, a proof-of-concept experiment upon a wireless network test-bed is illustrated.**

I. Introduction

To support the intended usage scenarios of IMT-2020, i.e., enhanced mobile broadband, ultra-reliable and low-latency, and massive machine-type communications, minimum technical requirements such as spectral efficiency, latency, and reliability, have been recommended by ITU-R [1]. The forthcoming fifth Generation (5G) system is envisioned to become more complicated so as to meet such radical key performance indicators, which inevitably impose a high challenge on network management. However, today's network managing approaches are already costly, vulnerable, time-consuming and therefore inapplicable to the 5G system. By the advent of Self-Organizing Networks (SON), mobile operators have to keep an operational group with a large number of network administrators with high expertise, resulting in a costly Operational Expenditure (OPEX) that is currently three times Capital Expenditure (CAPEX) and keeps rising [2]. Moreover, troubleshooting is hard to fully avoid an interruption of network operation, which deteriorates system's Quality-of-Service (QoS) and end user's Quality-of-Experience (QoE).

The SON concept [3] has been firstly introduced into cellular networks by 3GPP Release 8 as a driving 4G technology. However, current SON methods focused on traditional networks, which differ with software-defined and virtualized 5G infrastructure enabled by Software-Defined Network (SDN) [4] and Network Function Virtualization (NFV) [5]. Further, the automatic processing in SON usually relies on simple approaches like triggering and some operations are still carried out manually. Although the application of Machine Learning (ML) in SON is recently explored, the focus is mainly on utilizing a specific algorithm to tackle a single aspect of network, e.g., clustering to Mobility Load Balancing (MLB) and Q-learning to Inter-Cell Interference Coordination (ICIC). To the best knowledge of the authors, an intelligent framework that is capable of applying diverse ML techniques to manage a variety of network problems is still an open issue [6].

In this context, the SELFNET project [7] has been established to explore self-organizing network management in virtualized and software-defined 5G infrastructure. Relying on ML techniques, autonomic management with the capabilities of self-healing, self-protection and self-optimization can be realized [8]. As a part of SELFNET efforts, in this paper, we propose the concept of intelligence slicing, which enables a flexible and scalable framework for applying machine learning to implement autonomic management. Following the *divide-and-conquer* strategy, each intelligence slice only focuses on a dedicated network problem and can select the most suitable algorithm according to its characteristics. Besides, each slice can be independently on-boarded, updated and scaled, which provides the flexibility and scalability to handle a wider variety of existing and emerging network problems by means of accommodating current and potential ML techniques.

The rest of this paper is organized as follows: the next section presents the intelligence-slicing framework. The life-cycle management of slices and the definition of intelligence domain are discussed in Section III. Section IV illustrates a

[*]This work was supported by the European Union's Horizon 2020 Programme under the 5G-PPP project: *Framework for Self-Organized Network Management in Virtualized and Software Defined Networks* (SELFNET) with Grant no. *H2020-ICT-2014-2/671672*.

9781538614785/18/$31.00 © 2018 IEEE

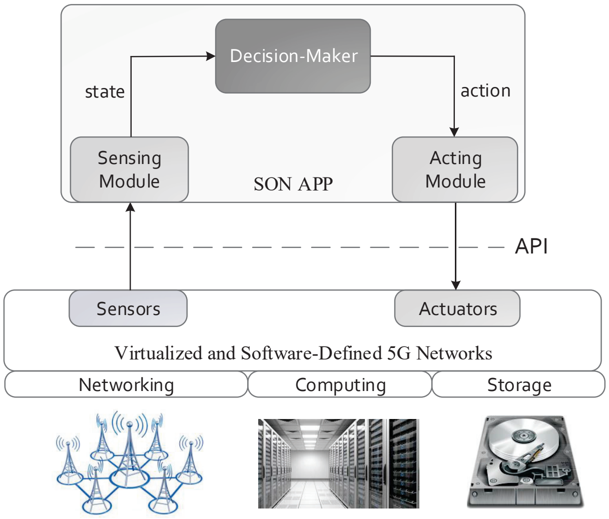

Fig. 1. Machine learning-based autonomic network management architecture.

proof-of-concept experiment upon a wireless test-bed. Finally, Section V concludes this paper.

II. Intelligence-Slicing Framework

This section first introduces the architecture of autonomic management for self-organizing 5G networks. Then, the concept of intelligence slicing that enables a flexible framework applying ML techniques to tackle a variety of network problems is presented.

A. Autonomic Management

Taking advantage of SDN and NFV techniques, network programmability is available through Application Programming Interface (API). The function of autonomic management can be regarded as an external SON Application (APP) on the top of software-defined and virtualized infrastructure. Apart from underlying physical and virtual resources, as illustrated in Fig.1, the autonomic architecture consists of: 1) SDN/NFV sensors that can extract metrics from networks; 2) Sensing module, in charge of collecting, aggregating, and analyzing metrics from sensors to derive network states; 3) the decision-maker that is responsible for diagnosing network problems and deciding an effective action; 4) Acting module that manages and coordinates physical and virtual resources to perform decided actions. 4) SDN/NFV actuators that are deployed in the network to perform a dedicated network function.

Once a network problem, such as Distributed Denial-of-Service (DDoS) attack, is detected, a SON control loop starting from sensors and terminating at actuators is triggered. The SON APP analyzes reported network states, diagnoses the root cause and decides a tactical action. As soon as the Acting module received an action request, it coordinates physical and virtual resources to enforce this action for mitigating this network problem. As shown in Fig.1, the input and output of the decision-maker are called *state* and *action*, respectively, which are defined as follows:

- **State**: A set of network-related information, such as alarms, events or metrics, that can be evaluated to indicate the characteristics of a network problem.
- **Action**: It is an implementable countermeasure to tackle the reported network problem taking into account available physical and virtual resources.

B. Intelligence Slicing

The 5G network management faces a more challenging situation than ever before. Most of the traditional problems in current networks still remain while some probably become more severe. For instance, the DDoS cyber-attack will be more impactive due to the introduction of Internet-of-Thing, where an attacker is able to compromise a large number of machine-type terminals as zombies. In addition, new management requirements, e.g., guaranteeing ultra-reliability and low-latency for upcoming industrial applications, will emerge.

In order to provide an affordable OPEX for 5G mobile operators, management tasks should be carried out in an autonomic way with the aid of learning techniques. However, different network problems have different characteristics so that a specific ML algorithm can only suit a very small portion of all possible problems. For example, smart antenna selection in Multi-Input Multi-Output (MIMO) system [9] requires a very prompt decision since radio channel's states vary within milliseconds. The detection of DDoS needs a global view of the network, leading to a huge data volume to be processed. On the contrary, only a few local network metrics are enough in the MIMO case. It is hard, if not infeasible, to find a universal ML technique to tackle all network problems. Moreover, due to the dynamicity of 5G infrastructure, the best algorithm for the current situation might be outdated with time goes.

In large-scale and heterogeneous networks, a centralized framework with a universal intelligence processing is too complex, inefficient and hard to meet the real-time requirement [10]. Therefore, in this paper, the concept of intelligence slicing is proposed to enable a flexible and scalable framework that can accommodate diverse ML techniques to deal with all kinds of possible problems. Following the *divide-and-conquer*

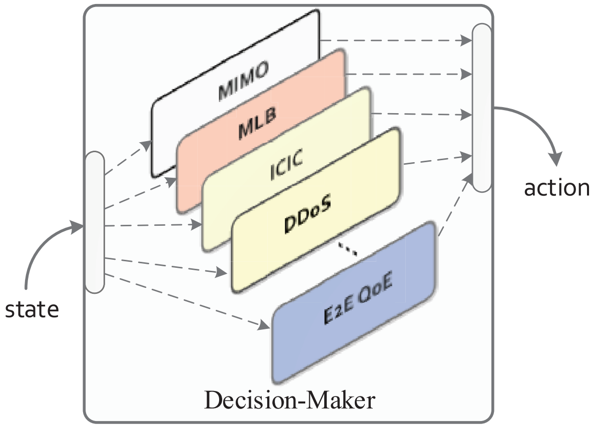

Fig. 2. The intelligence-slicing-based Decision-Maker.

strategy, each intelligence slice only focuses on a dedicated network problem. For instance, an MIMO slice is deployed in a Base Station (BS) to intelligently manipulate its antenna array, while another slice for DDoS can be instantiated in a data center to tackle anomalous traffic. The most suitable algorithm for its target problem is utilized to establish a slice. That is, the MIMO slice selects an algorithm with a predictive time as small as possible while the DDoS slice is capable of processing a big data. Different slices operate independently so that each slice has a flexibility to on-board, update, scale in terms of its respective situation. As shown in Fig.2, a number of exemplary slices for dealing with MIMO, MLB, ICIC, DDoS and End-to-End (E2E) QoE, respectively, are established in the SON decision-maker. Once the monitor reports a network state, it is forwarded to the respective slice to make a decision. It is noted that these slices are only *logically* centralized in the decision-maker and different slices can be deployed distributively in practice to meet diverse requirements in the heterogenous infrastructure.

III. Life cycle and domain of slices

Once an intelligence slice is on-boarded to handle a problem, its life starts. During its operation, the behavioral pattern of the target problem might vary due to the change of underlying infrastructure or environment. Thus, an updating or scaling is necessary. Later, after the pattern of this problem is well recognized by the management system, the slice can be further upgraded from an ML algorithm to rule-based processing so as to shorten processing time and simplify the system's implementation. In this section, the life-cycle management of slices and intelligence domain, which is defined as the effective area of a slice, are explained.

A. Life-Cycle Management

1) On-boarding: Once an unknown pattern indicating an unidentified network problem is found, the monitor reports its related network metrics to the decision-maker. These metrics are initially analyzed to qualitatively decide a suitable ML technique among supervised, unsupervised, reinforcement learning, etc. For each learning technique, there is a number of different algorithms. For example, supervised learning includes the following algorithms: Decision Tree (DT), Linear Discriminant Analysis, Support Vector Machine, Nearest Neighbor, etc. The decision-maker continues to make a quantitative comparison by calculating achievable performance for all available algorithms. Then, that algorithm achieved the optimal performance is applied in this slice. As shown in Fig.3, a slice called E2E QoE is established to guarantee the perceived quality of video delivery, which is based on a low-complexity DT algorithm.

2) Updating: There are two possible reasons for updating. First, the pattern of a network problem is possible to change due to the dynamicity of network infrastructure or external environment. For example, a cellular cell close to a shopping center is prone to be congested. From the perspective of ML, this situation can be reflected in a learning algorithm such as a specific leaf in a decision tree. Later, a small-cell BS is deployed there and the corresponding congestion vanishes from then on. As shown in Fig.3, the E2E QoE slice is updated by removing this outdated leaf from the previous classification tree. In addition to the change of a pattern, another motivation for updating is the emergence of a novel algorithm that can better process a fixed pattern.

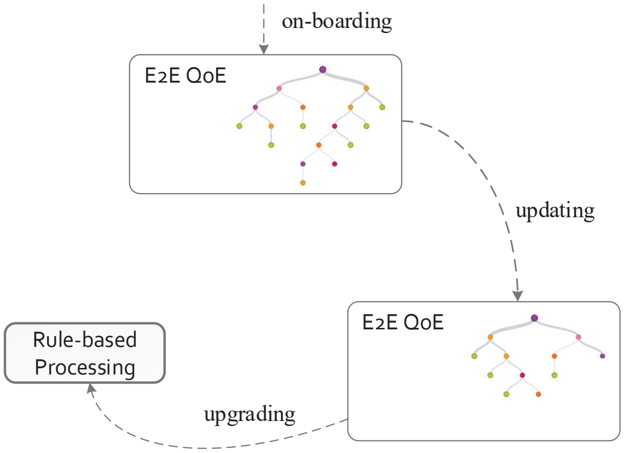

Fig. 3. The life cycle of an intelligence slice.

3) Upgrading: Experienced a number of times updating, an on-boarded slice steps into a stable phase. Here, we do not discuss the detailed criterion of indicating a stability. Simply speaking, when a slice can always properly handle a network problem in a very high accuracy, it can be regarded as a matured slice. To speed up the processing and simplify the system's implementation, a matured slice is upgraded to rule-based processing, as illustrated in Fig.3. The rules can be specified simply by script languages, e.g., eXtensible Markup Language (XML). When the monitor reports this problem again, the decision-maker invokes the rule-based processing directly to make a quick decision.

B. Intelligence Domain

We presented the life cycle of a slice, i.e., on-boarding, updating and upgrading, from a *temporal* point of view. This section further discusses intelligence domain, which is defined as the effective area of a slice, from a perspective of *spatial* coverage. As mentioned before, different network problems have different characteristics, among which its effective area is an important factor. Using the MIMO slice as an example, the decision on antenna selection is derived from channel states between antenna array and the users within a cell and the impact of this decision is also constrained in this cell. Due to fast fading of radio channels, such physical layer decisions should be quickly made and executed. To minimize transmission delay, the MIMO slice has to deploy as close to the antenna area as possible. As shown in Fig.4, the cell coverage of LTE BS 1 is also the effective area of the MIMO slice, which is referred to as intelligence domain. Mobile load balancing is an effective SON approach to improve the

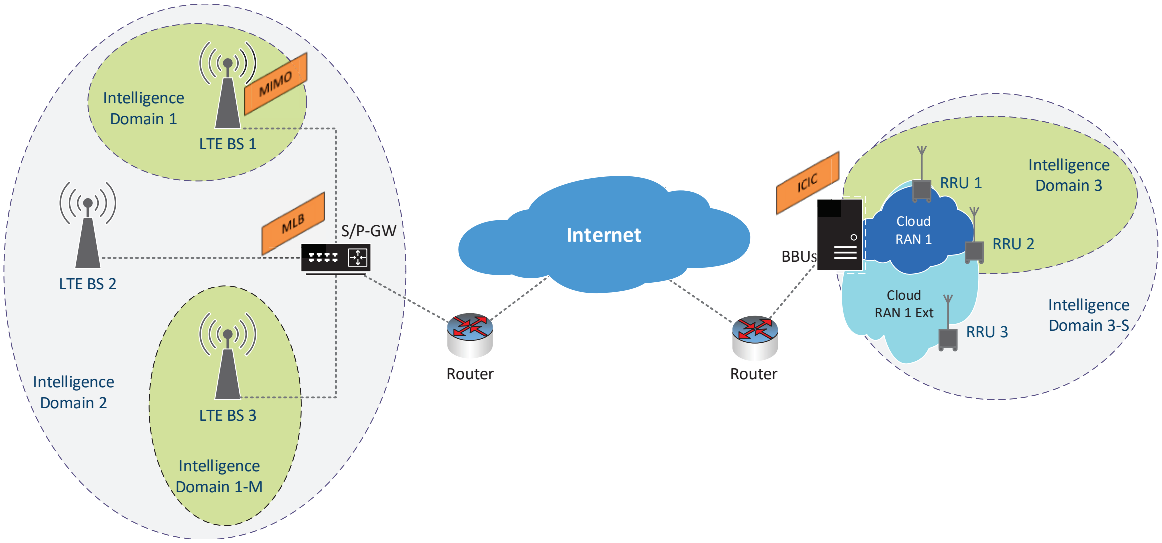

Fig. 4. Illustration of intelligence domains.

end-user's QoE and increase system capacity by dynamically distributing user traffic across several adjacent cells. Those adjacent BSs participating in load balancing should reports its metrics to a centralized coordinator. As shown in Fig.4, intelligence domain 2 covering three LTE BSs stands for the effective area of the MLB slice. In comparison to fast fading, traffic's fluctuation within a cell is relatively slow. Hence, the MLB slice is able to tolerating the transmission delay and can be deployed in the Serving/Packet data network Gateway (S/P-GW) to facilitate a centralized control. Cloud Radio Access Network (RAN) is a novel mobile architecture, which pools Base-Band Units (BBUs) of a number of BSs and connects Remote Radio Units (RRUs) via fiber-optic front-haul. As shown by intelligence domain 3 in Fig.4, the ICIC slice can be deployed in the BBU pool, where cooperatively processing signals originating from several cells is straightforward.

Similar to the life-cycle management, there are two operations for an intelligence domain, which are given as follows:

1) Mirroring: The establishment of an intelligence slice is not a trivial work since a learning system needs to be trained into a learned system. For supervised and unsupervised learning, a training dataset is necessary. Data acquisition is sometimes difficult, especially for supervised learning, where data need to be labeled. Reinforcement learning does not need training dataset, but the learning system has to iteratively try all possible actions for each state and observe their outcomes. The learning process is time-consuming and computationally complex. In a large-scale network, if each intelligence slice is independently trained, it will be a tremendous work. In some cases, fortunately, the learned system can be directly duplicated to similar situations. For example, as shown in Fig.4, the MIMO slice trained based on manipulating experiences in the antenna array of LTE BS 1 can be mirrored to BS 3. That is because the antenna selection is in terms of channel state information, which is independent to base stations. Hence, BS 3 can copy an on-boarded slice directly without a training phase. Even if the situation in BS 3 has some differences with the original BS, the slice can adapt to the new situation through a re-training during its operation, which is anyway better than training from scratch. In this figure, intelligence domain 1-M denotes a mirrored domain from intelligence domain 1.

2) Scaling: Scaling means an intelligence domain being enlarged or shrunk since the effective area of a slice needs to change due to the dynamicity of underlying infrastructure or external environment. At the early stage, Cloud-RAN 1 covers two RRUs, i.e., RRU 1 and 2. When a new RRU is deployed, this cloud is extended to a larger coverage with three RRUs, as indicated by Cloud-RAN 1 Ext illustrated in Fig.4. Accordingly, the intelligence domain of the ICIC slice should be scaled to a larger effective area to coordinate the inter-cell interferences among three RRUs. As illustrated in Fig.4, intelligence domain 3-S is scaled from intelligence domain 3.

IV. Experimental Demonstration

To justify the ML-based network management framework, a proof-of-concept experiment in terms of an end-to-end video transmission is carried out in a wireless network test-bed. As shown in Fig.5, the test-bed is differentiated into several layers: physical infrastructure, virtual network, SON, and application layer. The physical infrastructure mainly consists of several high-volume servers, Mini-PCs, two switches and Radio Frequency (RF) units. A switch is utilized to interconnect the servers and acts also as a gateway to Internet. OpenStack [11], an open-source software to create a cloud environment, is installed in a server to support NFV-based networking. Another server equipped with a special graphical card is allocated to run ML algorithms in a parallel-computing way with its powerful Graphical Processing Units. Other servers and Mini-PCs act as computing nodes under the management of an OpenStack controller, which is responsible for instantiating virtual machines so as to carry Virtual Network

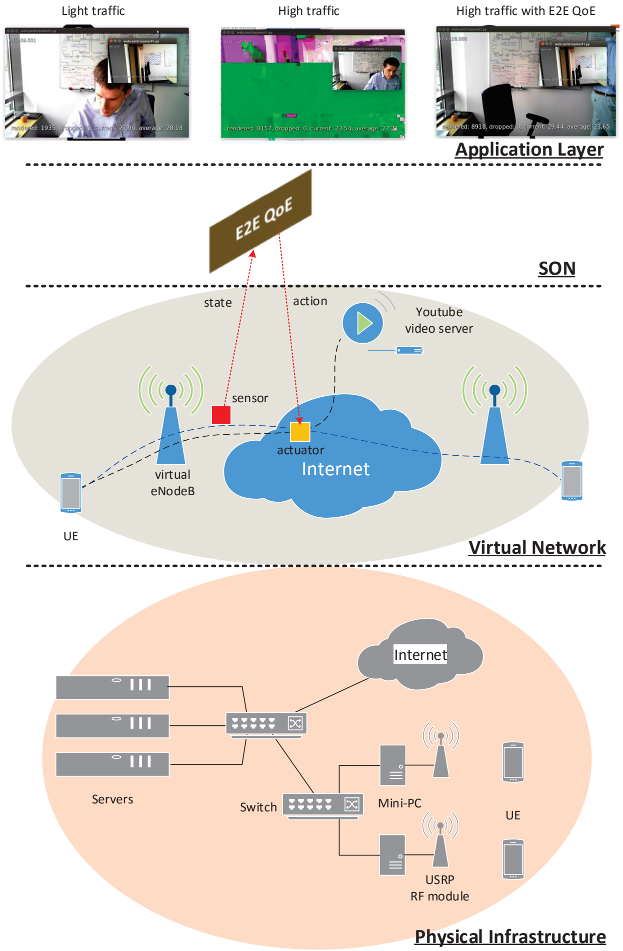

Fig. 5. An E2E video experiment upon the wireless test-bed with differentiated layers.

Functions (VNFs). To emulate a realistic wireless scenario, OpenAirInterface [12], an open-source LTE implementation, is utilized. It provides a full set of VNFs for LTE radio access and core networks, such as eNodeB, Mobility Management Entity (MME) and S/P-GW. Besides, software-defined radio modules (i.e., USRP B210) with antennas are applied to support a radio link with User Equipment (UE).

The scenario of video transmission is used as an example to show self-optimization capability of an intelligence slice [13]. As shown in Fig.5, a virtual network consisting of two virtual base stations and related core network functions is instantiated on the top of the physical infrastructure. A mobile user accesses the YouTube server to watch high-definition video programs. Meanwhile, this user sets up a peer-to-peer (P2P) video talk with another user. As shown in Fig.5, two dashed lines in the virtual network denote two video flows for P2P communication and YouTube down-streaming, respectively. To emulate traffic congestion in a real network, the transmission bandwidth of the virtual network is deliberately selected. Thus, once either video flow generates a high-volume traffic, the congestion that leads to a worse QoE occurs.

To guarantee the perceived QoE of end users, an intelligence slice called E2E QoE is on-boarded in the SON decision-maker. This slice's function is identifying congestion's occurrence and deciding a countermeasure action. Here, an instance of supervised learning, i.e., classification, is selected as the ML technique. A number of typical classifiers, i.e., Decision Tree (DT), Linear Discriminant Analysis (LDA), Support Vector Machine (SVM) and Nearest Neighbor (NN), are implemented. These algorithms that are applied in the test-bed are briefly reviewed as follows:

TABLE I
LIST OF EXEMPLARY METRICS

Index	Feature	Definition
1	EPC_Traffic_In	Incoming trafffic of EPC
2	EPC_Traffic_Out	Outgoing trafffic of EPC
3	Server_Traffic_In	Incoming trafffic of iPerf3
4	PLR	Average Packet Loss Rate
5	Delay	Round trip delay
6	eNB_CPU_Util	eNodeB's CPU utilization
7	eNB_Mem_Util	eNodeB's memory utilization
8	eNB_CPU_Temp	eNodeB's CPU temperature
9	UE_CPU_Util	UE's CPU utilization
10	UE_Mem_Util	UE's memory utilization

a) DT: Decision Tree [14] is a classical supervised learning method used for classifying. Decision rules are inferred from a training dataset and a tree-shaped diagram is built. Each node of the decision tree relies on a feature to separate the data, and each branch represents a possible decision. DT is simple, interpretable and fast, whereas it is hard to apply in a complex and non-linear case.

b) LDA: Discriminant analysis is a classification method, which assumes that different classes generate data based on different Gaussian distributions. LDA [15] is to find a linear combination of features that maximize the ratio of inter-class variance to the intra-class variance in any particular dataset so as to guarantee maximal separability.

c) SVM: SVM [16] utilizes a so-called hyperplane to separate all data points of one class from another. The number of features does not affect the computational complexity of SVM, so that it can perform well in the case of high-dimensional and continuous features. However, it is a binary classifier and a multi-class problem can be solved only by transferring into multiply binary problems.

d) NN: Another algorithm called Nearest Neighbor is applied for data classification following the hypothesis that close proximity in terms of inter-data distance have an similarity. The class of an unclassified observation can be decided by observing the classes of its nearest neighbors. It is among the simplest algorithms with a good predictive accuracy. But it needs high memory usage, is vulnerable to noisy data and is not easy to interpret.

Since classification algorithms are data-driven, a training dataset is necessary for both training and prediction phases. In the experiment, by means of different SDN/NFV sensors such as Zabbix [17], a wide variety of network metrics such

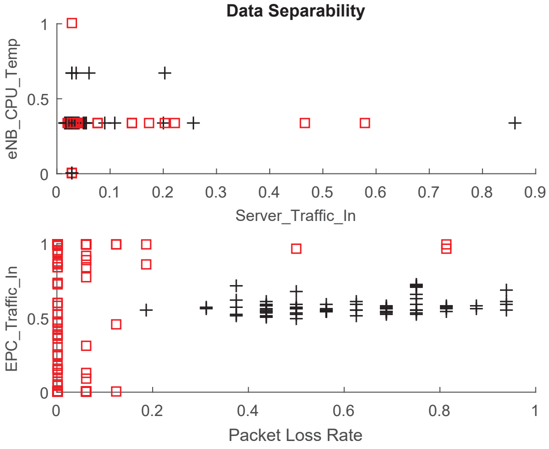

Fig. 6. An example of data separability.

as those listed in Table I, can be collected. Accordingly, a training dataset consisting of the listed network metrics with 600 observations has been acquired. By exhaustively evaluating predicative accuracies of each classifier in terms of different combinations of network metrics, we can conclude that Decision Tree and Nearest Neighbor can achieve the optimal accuracy of 100% merely by using two metrics: *eNB_Mem_Util* and *EPC_Mem_Util*, which are the memory usages of virtual machines running network functions for eNodeB and EPC, respectively.

To observe the data separability of collected network metrics, some metrics in the dataset are visualized. Randomly selecting a data observation, whose features have the following normalized values: *EPC_Traffic_In*=0.996, *PLR*=0, *Server_Traffic_In*=0.034, *eNB_CPU_Temp*=0.667, and its label *Congestion*=0. In the lower part of Fig.6, two most relevant features *PLR* and *EPC_Traffic_In* are applied. The mark ’+’ denotes the presence of congestion, while ’□’ stands for the absence of congestion. Similarly, in the upper figure, two irrelevant features are used. As shown in Fig.6, in the case of applying two relevant features, the congested and non-congested data have a clear border with highly separability. In comparison, the data in the case of applying two irrelevant features are hard to be separated. It implies that the correct selection of metrics with the high relevance plays an important role in the data classification.

Thus, the E2E QoE slice is on-boarded by means of applying the trained classifier. The sensor continuously collect these two metrics from the wireless network. Once a video traffic congestion is detected, the SON framework responds to provide an effective action, e.g., allocating more resources to a virtual switch or degrade the definition of video. In the experiment, as illustrated in the application layer of Fig.5, there is no congestion and the quality of video is good in the case of *light* traffic. Before the on-boarding of the slice, once the traffic increases beyond a threshold, the video becomes blurry due to congestion, as shown in the case of *high* traffic. On the contrary, the E2E QoE slice can avoid traffic congestion even if the traffic is *high* and guarantee the perceived QoE. We can make a conclusion here that the applied intelligence slice is effective to the target problem.

V. Conclusions

In this paper, we proposed an intelligence-slicing framework for applying machine learning to autonomically manage the upcoming 5G networks. Instead of focusing on a specific network problem or exploring a universal algorithm, this framework was designed to provide flexibility and scalability for accommodating diverse learning techniques to tackle a wider variety of network problems. A proof-of-concept experiment in terms of video QoE provisioning was carried out in a wireless test-bed to demonstrate this framework. In the next step, more ML algorithms and other representative network problems will be experimented in the test-bed to further verify the effectiveness of the intelligence-slicing framework.

References

[1] ITU-R, “Minimum requirements related to technical performance for IMT-2020 radio interface(s),” Working Party 5D, Tech. Rep., Feb. 2017.

[2] “Top ten pain points of operating networks,” Aviat Networks, 2011.

[3] S. Dixit *et al.*, “On the design of self-organized cellular wireless networks,” *IEEE Commun. Mag.*, vol. 43, no. 7, pp. 86–93, Jul. 2005.

[4] B. A. A. Nunes *et al.*, “A survey of software-defined networking: Past, present, and future of programmable networks,” *IEEE Commun. Surveys*, vol. 16, no. 3, pp. 1617–1634, 2014.

[5] R. Mijumbi *et al.*, “Network function virtualization: State-of-the-art and research challenges,” *IEEE Commun. Surveys*, vol. 18, no. 1, pp. 236–262, 2016.

[6] W. Jiang, M. Strufe, and H. D. Schotten, “Autonomic network management for software-define and virtualized 5G systems,” in *Proc. European Wireless*, Dresden, Germany, May 2017.

[7] EU H2020 5G-PPP SELFNET project. [Online]. Available: https://selfnet-5g.eu/

[8] W. Jiang, M. Strufe, and H. D. Schotten, “Intelligent network management for 5G systems: The SELFNET approach,” in *Proc. IEEE European Conf. on Net. and Commun. (EUCNC)*, Oulu, Finland, Jun. 2017, pp. 109–113.

[9] J. Joung, “Machine learning-based antenna selection in wireless communications,” *IEEE Commun. Lett.*, vol. 20, no. 11, pp. 2241–2244, Nov. 2016.

[10] W. Jiang, M. Strufe, and H. D. Schotten, “A SON decision-making framework for intelligent management in 5G mobile networks,” in *Proc. 3rd IEEE Intl. Conf. on Compu. and Commun. (ICCC)*, Chengdu, China, Dec. 2017.

[11] OpenStack. [Online]. Available: https://www.openstack.org/

[12] N. Nikaein *et al.*, “OpenAirInterface: A flexible platform for 5G research,” *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 33–38, 2014.

[13] W. Jiang, M. Strufe, and H. D. Schotten, “Experimental results for artificial intelligence-based self-organized 5G networks,” in *Proc. IEEE Intl. Symp. on Personal, Indoor and Mobile Radio Commun. (PIMRC)*, Montreal, QC, Canada, Oct. 2017, pp. 1–6.

[14] S. K. Murthy, “Automatic construction of decision trees from data: A multi-disciplinary survey,” *Journal on Data Mining and Knowledge Discovery*, vol. 2, no. 4, pp. 345–389, Dec. 1998.

[15] Y. Guo, T. Hastie, and R. Tibshirani, “Regularized discriminant analysis and its application in microarrays,” *Biostatistics*, vol. 1, no. 1, pp. 1–18, 2005.

[16] C. J. Burges, “A tutorial on support vector machines for pattern recognition,” *Journal on data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, Dec. 1998.

[17] ZABBIX: The enterprise-class monitoring solution for everyone. Zabbix LLC. [Online]. Available: http://www.zabbix.com/