A Machine Learning Based Management System for Network Services

Yekta Turk[1], Engin Zeydan[2] and Zeki Bilgin[1]
[1] Ericsson Research, Istanbul, Turkey, 34396. E-mail: [yekta.turk, zeki.bilgin]@ericsson.com
[2] CTTC, Castelldefels, Barcelona, Spain, 08860. Email: engin.zeydan@cttc.cat

***Abstract*—Providing high quality and uninterrupted network service is becoming crucial for service providers. In this paper, we present an approach to quantify and indicate service quality based on the topology state transitions from the perspective of network service provider. Building our model as a Finite State Machine (FSM), we show novel application of machine learning (ML) classification algorithms to classify appropriate states for undefined input alphabets in FSM. In other words, we implement ML algorithms to extract both service states and possible root causes of service degradation only from measured certain Key Performance Indicator (KPI) values that are observed directly through network elements. We have implemented our network topology state classification approach using the dataset obtained in Graphical Network Simulator 3 (GNS3) simulation environment, and performed measurements to evaluate classification accuracy of different algorithms. Additionally, we have identified priority of relevant KPIs impacting the service quality. Our results indicate that network topology state changes can be classified up to 88% accuracy and F1 scores using ensemble learning methods such as Gradient Boosting Classifiers.**

***Index Terms*—network, log analysis, automation, machine learning.**

I. Introduction

Emerging network technologies, such as 5G and Internet of Things (IoT), adopt service-oriented architectures. This leads to new business opportunities in both vertical and horizontal markets. In this context, an important requirement will be accurate assessment of service quality. It is a well-known argument that *you can't manage what you can't measure*. In an environment where there are so many services, it can be expected that automated management of these services with zero touch would be very appropriate for Mobile Network Operators (MNOs) in the perspective of network operations. Therefore, it is very important for network operators to provide healthier services to their end-users over their networks. However, it is not always possible to monitor the quality of provided services due to several reasons such as privacy concerns of end-users, abundance of service consumers, inability of the service provider to observe service quality on the consumer side and excessive workload. For these reasons, network operators are mostly not aware of real-time quality and operational aspects of the services they provide. Besides service monitoring, finding the root cause of the problems that affect service quality has very importance for MNOs.

As there is a visible tendency towards self-management in network technologies, by exploiting the power of rising Artificial Intelligence (AI)/Machine Learning (ML) techniques, it is time to (re)consider how to measure and predict service status based on only the observable parameters on the side of network providers. In this paper, we open a relatively little studied issue of network service management up for discussion, and propose a methodology to classify network service quality state as well as find the root cause of the network problem using ML technique of classification. Although the Zero-Touch Service Management (ZSM) [1] study managed by European Telecommunications Standard Institute (ETSI) proposes a concept for adapting ML to service management, low level details such as which Key Performance Indicators (KPIs) will be used and using which architecture the ML algorithms will be implemented, are not described yet. On the other hand, ZSM focuses on services provided by the network and management of these services which is a different concept from the management of the network. In this study, ML based service management is discussed at implementation level. We also show feasibility of the proposed methodology and the accuracy of the root cause classifications by means of emulations using the data generated from Graphical Network Simulator 3 (GNS3) simulation environment.

A. Related Work

There are many studies that are investigating the application of ML techniques to the area of network management and this area has been emerging in recent times. However, these studies are mostly concerned with the network management as a whole and are not focused on specifically service management. In this sense, the article [2] reviews the applications of fundamental ML concepts on communication networks, and presents a case study which aims at detecting abnormal elements in a multi-layer real network using unsupervised ML. Another paper [3] reviews the well-known ML concepts along with their applications in the context of optical networks, and discusses different aspects of ML implementations such as algorithm choice, data and model management strategies, and integration into existing network control and management tools. The survey paper [4] provides an overview of deep learning architectures and algorithms for controlling network traffic, and surveys the state-of-the-art ML and new deep learning researches in networking related areas.

There are also studies presenting implementations of ML techniques to leverage autonomy and self-management capabilities while exploiting the available of networking data [5]–[9]. These studies focus on improving ML techniques for network management but do not provide a system that is concentrating specifically on network services. For example, the article in [5] considers the usage of ML for cognitive network management, and provides a discussion on how to defeat the bottlenecks and limiting factors for the deployment of autonomic systems with ML. Similarly, the study in [6] proposes a cognitive management framework with unsupervised deep learning and probabilistic generative models for network optimization. The paper in [7] proposes a prediction model with deep learning for internet traffic flow forecast in real-time. To build Self Organizing Networks (SONs), the paper in [8] leverages cellular mobile data to cluster the Base Stations (BSs) using unsupervised learning approaches so that inter-cluster handover rates can be reduced. Similarly, the authors in [9] predicts the number of potential user equipments (UEs) in a given cell using Bayesian neural networks.

ML techniques and management of resources in physical or virtual environments are also well studied in the literature [10]–[13]. The study in [10] considers the problem of resource management in virtual networks, and presents the implementation of a distributed reinforcement learning algorithm to allocate resources dynamically in virtual network environment. Similarly, another study in [11] examines dynamic resource assignments for Virtual Network Functions (VNFs) using a method based on neural networks to estimate service degradation from observable performance data. The work in [12] explores the application of ensemble learning models to multiple network measurement problems, and introduces a generic ML model for the analysis of network measurements. The study in [13] presents two examples of ML implementation for control and management in optical networks. However, these studies do not identify the root causes of problems especially in service management domain. In our study, the root cause of service problems is determined by ML.

B. Our Contributions

In this paper, we consider the problem of network service management for MNOs, which is an important issue in compliance with the Service Level Agreement (SLA) between a service provider and end-users. Since SLAs usually define the required level of service expected from the service provider, real-time monitoring of the provided service quality is a top priority for network service providers. For this purpose, we define several network service states, each of which represents different levels of service quality provided by the network infrastructure. To classify these network states together with their corresponding root causes without relying on any information from the end-user side, we develop a methodology based on ML techniques. We use KPI values of the active underlying network topology for our classification purposes. Moreover, we prioritize these KPIs according to their impact on service quality by performing experimental evaluations using the data generated from GNS3 with real network configurations on routers and switches. On the theoretical side, we present formal definition of our problem of interest by formulating our model as a Deterministic Finite Automata (DFA), and show how we use ML to find likely states in Finite State Machine (FSM) for undefined input alphabets. This paper is also introducing the experimented dataset which is provided to IEEE DataPort platform as well as related ML algorithms that we have studied for root cause analysis purposes. We experimentally show that the proposed methodology can classify network service states together with their root causes up to 88% accuracy and F1 scores using ensemble learning methods such as Gradient Boosting Classifiers.

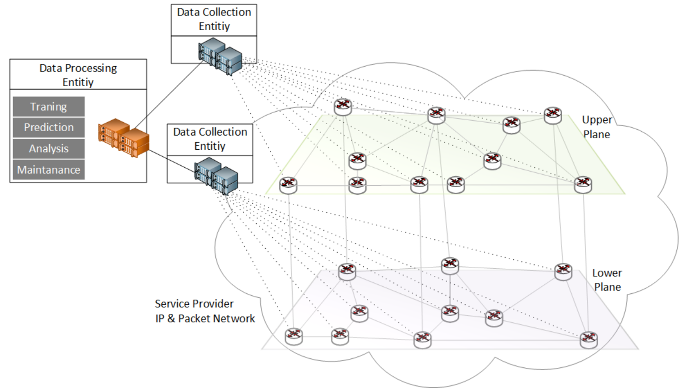


Fig. 1: The architecture of the predictive service management system with data collection and processing entities.

II. Designing Network Service for Predictive Management

Service providers design their networks in a redundant way in terms of physical connections to provide uninterrupted services to their end-users. This is also the case for the protocols used for routing and switching where all network configurations are made to run the network in a redundant form. With the help of redundant design, uninterrupted service can be provided in case some interruptions exist in a given route by accessing into an alternative route. Fig. 1 depicts a service provider network with two backup planes that contain primary and redundant routes planned for a service as well as data collection and processing entities. When designing Wide Area Networks (WANs), it is not expected to deploy as many connections and equipment as in typical local area networks due to operating expenditure (OPEX) and capital expenditure (CAPEX) constraints. In fact, the underlying backbone network of MNOs consists of two planes as the upper plane and the lower recovery plane. There are certain points where transitions are achieved between the two planes. The number of these transition points where the network traffic is flowing may vary depending on network design and the devices that can be deployed redundantly at these important transition points.

One of the important steps for the correct operation of the service providers' networks is to design the service routing paths to be installed for new internal or external end-users on the network. These network services can be any Layer 2 (L2) or Layer 3 (L3) services. End-users may want to use these services in either redundant or non-redundant mode based on their service requirements and tariffs. However, in practical systems most of the services are used in redundant mode. In this case, the planning units that are designing the network should plan the plane of the primary and the secondary paths of the service. Considering that there are hundreds of services within a network, this design phase is important to put less load on the same points with regard to capacity, capabilities and network robustness requirements. Depending on whether the service is L2 or L3, some of the redundancy services and techniques such as Virtual Router Redundancy Protocol (VRRP), Virtual Private LAN Services (VPLS), Routed VPLS (R-VPLS), Link Aggregation Group (LAG), Multi-Chassis Link Aggregation Group (MC-LAG), Multi-Protocol Label Switching Traffic Engineering (MPLS-TE) and Multiprotocol Extensions for Border Gateway Protocol (MP-BGP) attributes, etc. can be utilized.

A. Monitoring the Status of the Service

The performance of any network service can be monitored by means of probes which are put into user endpoints. In this case, via the IP-SLA tests which will run continuously on the probes, the performance status can be observed with tests such as iPerf. However, this observation is limited only to the performance of the service and does not contain adequate information regarding the root cause of the problems on the service. In this way, it is impossible to determine the effect of a problem caused by the provider network on the service. Once the problem has been detected by means of probes, operators can perform root cause analysis by manually operating fault management processes. In addition, these probe tests focus more on the correct operation of the services rather than service planning of the MNO. However from the perspective of the service provider, running each separate services as planned is critical for the correctness and robustness of the provided services over the network infrastructure.

Regarding the impacts of certain network problems on service status, the link or node failures are quickly detected by routing and transport protocols and the traffic direction is updated instantly to avoid service interruptions. On the other hand, when there is a faulty link causing packet losses a degradation occurs in packet delivery ratio at the network layer. Such an issue arising in the physical channel is mostly not be taken into consideration by the underlying routing protocol. Hence, service interruptions may be experienced while network traffic flows through such a faulty link. The same issue may also be experienced in the presence of faulty nodes caused by hardware or software defects. For this reason, the fault types causing to utilize poor channels in service delivery are identified within the problem types leading to Fair state, whereas the fault types causing sharp disconnections on a path are identified within those leading to Good state. Faulty link is the link that creates delay, packet error, burst error and packet drops. Faulty link can include one or more of these four conditions.

B. Formal Definition of the Proposed Model

In this subsection, we construct formal description of our model using finite automata theory. To do so, we specify 4 discrete service quality states that are determined based on certain network level KPI values. These states are shown in set Q and described below:

$$Q = \{Best, Good, Fair, Bad\}$$

where

- **Best:** The underlying network topology supporting the service and all related KPI values are as they should be.
- **Good:** The underlying network topology supporting the service is changed with respect to the original setting due to node failure, link failure or incorrect configuration of routing, and minority of the related KPI values are not within the expected range.
- **Fair:** The underlying network topology supporting the service is changed due to faulty node, faulty link or bandwidth saturation, and majority of the related KPI values are not within the expected range.
- **Bad:** There is faulty network element particularly at the service end point in the underlying network topology, and most of the related KPI values are out of the required range.

We use these states to quantify and represent service quality levels based on observed cumulative KPI values. Transition from one state to another one takes place

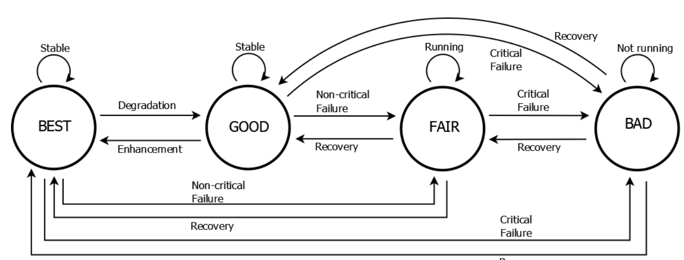


Fig. 2: Finite state machine and state change transitions between Best, Good, Fair and Bad topologies.

depending on variations of KPI values. From this, we build a FSM corresponding to our model along with state diagrams and transition relations as shown in Figure 2. This is a DFA (Deterministic Finite Automata) as the machine goes to one state only for any particular input. We define our state machine with a five-element tuple: $(Q, \Sigma, \delta_1, q_0, F)$ where Q is a finite set of states, Σ is a set of input alphabet, δ_1 is transition functions, q_0 is initial state and F is set of final states.

The input alphabet, Σ, is comprised of a set of KPIs. Let $\mathbf{S}$ be the set of KPIs that we are interested in.

$$S = \left[x_{i,j}\right]_{n \times m}$$

where $x_{i,j}$ is a KPI value. In this matrix representation, each column contains values for a particular KPI at different times, whereas each row includes values for all KPIs at a specific time. This implies that there are m KPIs and n instances. Each row comprises an input alphabet.

$$\Sigma = \{x_1, x_2, ..., x_n\}$$

where x_i is a vector and corresponds to an input alphabet.

The transition function, δ_1, determines the next state based on current state and input alphabet. Thus, we can formulate it as follows:

$$\delta_1 : Q \times \Sigma \longmapsto Q$$

Notice that there might be some input alphabets undefined in the DFA, which makes the DFA incomplete. This is actually inevitable in our use case especially when considering that there are a lot of KPI types which takes continuous values. Therefore, it is almost impossible to cover all possible input alphabets. This is where our ML model comes into play. We employ ML techniques as a generalization tool in the sense that it predicts likely states for undefined inputs based on given transition function. To the best of our knowledge, this is first study that use ML for predicting states in a DFA for undefined inputs. Our approach, thus, enables to avoid deadlock states in FSM as well.

On the other hand, there are specific root causes of KPI variations such as node or link failures, faulty network elements, route change, protocol down, and more. Let R denote this set of root causes as follows:

$$R = \{r_1, r_2, ..., r_t\},$$

where r_i for $i = 1, 2, ..t$ represents a specific root cause. There is a causal relationship between root causes and KPI values. We show this relationship with another function δ_2 as follows:

$$\delta_2 : Q \times \Sigma \longmapsto R$$

Now we can define our ML problem based on given functions and sets above. Let Σ^* be a set containing all undefined input alphabets in the DFA.

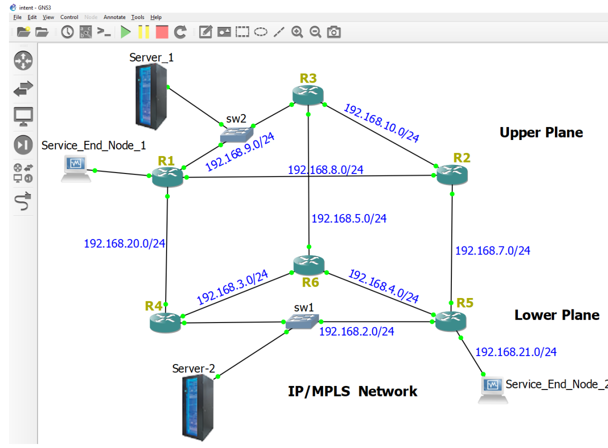


Fig. 3: Simulation setup.

$$\Sigma^* = \{x_{n+1}, x_{n+2}, ..., x_{n+t}\}$$

Problem Definition (Multi-label Classification): Given δ_1, δ_2 and Σ^*, calculate δ_1^* and δ_2^* such that $\delta_1^* : Q \times \Sigma^* \longmapsto Q$ and $\delta_2^* : Q \times \Sigma^* \longmapsto R$.

Notice that in this model δ_1 and δ_2 constitute training set of data, details of which are described earlier, whereas δ_1^* and δ_2^* correspond to set of test data. ML algorithm, basically, constructs an advanced classification model based on the training dataset to find correct classification of unknown service states and related root causes in the test dataset.

III. Simulation Results

A. Emulation Environment

We have prepared our emulation setup using the GNS3 emulator environment, which is close to real systems. The configurations on routers are based on a real Internet Protocol (IP)/Multiprotocol Label Switching (MPLS) network. MPLS operates in Label Distribution Protocol (LDP) mode and Open Shortest Path First (OSPF) is used as a routing protocol. All routers are in the area 0. MP-BGP has been activated for Virtual Private Network (VPN) services. An L2 service has been opened between the service-end-node-1 connected to the R1 router and the service-end-node-2 connected to the R5 router as shown in Fig. 3. The primary path of the service was determined to be R1-R4-R5 and is more likely to go through the lower plane. In other words, this is the appropriate path of the service at the design stage and it is expected to go through this route when there are no problems. The redundant path is the R1-R2-R5 path that uses the upper path. In order to not to be affected by failures on link and router and perform continuous data transfer, the servers are connected to the switches. In case the servers are connected to a router and a problem exists with this router, all network connections will be disconnected during data collection process. On the other hand, data can be

TABLE I: All features and their corresponding descriptions of the dataset in [14]

Feature	Description	Feature	Description	Feature	Description
sysUpTimeInstance	System open time	IFInUnknownProtos$^{(*)}$	number of unreadable packets from port	IFOutErrors$^{(*)}$	Number of incorrect packets outgoing from port
IFAdminStatus$^{(*)}$	administrative open or closed state of the port	IFLastChange$^{(*)}$	time elapsed when the port was last opened or closed	IFOutOctets$^{(*)}$	total number of packets outgoing from port
IFInDiscards$^{(*)}$	Number of incoming dropped packets	IFMtu$^{(*)}$	Maximum size of the Ethernet packet through the port	IFOutQLen$^{(*)}$	Number of pending queues in port
IFInErrors$^{(*)}$	number of incorrect packets received at incoming port	IFOperStatus$^{(*)}$	whether the port works or not	IPCidrRouteNextHop & IPCidrRouteType $^{(*)}$	Routing Table: Next Hop & Type
IfInOctets$^{(*)}$	Total number of packets received at incoming port	IFOutDiscards$^{(*)}$	number of outgoing packets that are dropped on port	IPFragFails	Number of non-fragmantable IP packets
IPInAddrErrors	The number of input datagrams discarded due to header's destination field was not valid	memory_usage	Memory Usage	tunnel_uptime	Service duration
IPInHdrErrors	The number of input IP datagrams discarded due to errors in their IP headers	tunnel_received_byte	Number of packets received from service	tunnel_unsent	Number of packets that cannot be sent in service
IPOutDiscards	The number of output datagrams discarded due to IPv4 header's destination field was not a valid	tunnel_resend	Number of packets retransmitted in service	CPU_usage	CPU Usage
IPReasmReqd	Number of IP packets that can not be reassembled	tunnel_sent_bytes	Number of packets sent from service	class	represents 23 distinct root causes

TABLE II

States of the network services and their corresponding root causes defined from quality and service design perspectives.

Service State	**Quality Status**	**Service Design Status**	**Problem Type**	**Root Cause**
Best	Stable	As planned	- No root cause	- Routing Path (R1,R4,R5)
Good	Stable	Not as planned	- Node Failure on the Path - Link Failure on the Path - Incorrect Configured Routing/ Transport Protocol in the Network	- Failure R2, - Failure R3 - Failure R4, - Failure R6 - Failure R2 and R4 - Link Fail R1 to R2, - Link Fail R1 to R3 - Link Fail R1 to R4, - Link Fail R4 to R5 - Link Fail R1 to R2 and R1 to R4 - Routing Changed over R2
Fair	Not Stable	Ignored	- Faulty Node on the Path - Faulty Link on the Path - Service Bandwidth Saturation	- Faulty R2, - Faulty R3 - Faulty R4, - Faulty R5 - Faulty Link R1 to R2, - Faulty Link R1 to R3 - Faulty Link R1 to R4, - Faulty Link R4 to R5 - Load Traffic
Bad	No Service	Ignored	- Service End Node Failure - Protocol Failure in Network/Service	- Failure R5 - Protocol Down in Service

collected continuously by using switches. Data collection and processing entity is run in the same server due to the constraints of the simulation environment. All KPI data were collected from both R1 and R5 routers expressing the start and end points of the service. The L2 service is a point-to-point virtual leased line service. We also assigned a quality-of-service (QoS) value to this service, and put the service's bandwidth limit to 5 Mbps. We created traffic on both service end-nodes using iPerf. To generate service traffic, we have mutually generated two-sided User Datagram Protocol (UDP) traffic.

B. Dataset and training process

We have introduced our dataset used in our analysis via IEEE Dataport web portal [14] in csv format. Table I summarizes all the features and their corresponding de-

TABLE III: IP address vs type mapping used for feature naming in [14]

IP Address	Type	IP Address	Type	IP Address	Type
0.0.0.0	type0	192.168.3.0	type30	192.168.8.2	type82
1.1.1.1	type1	192.168.4.0	type40	192.168.9.0	type90
3.3.3.3	type3	192.168.5.0	type50	192.168.9.2	type92
4.4.4.4	type4	192.168.5.2	type52	192.168.10.0	type100
5.5.5.5	type5	192.168.6.0	type60	192.168.30.0	type300
6.6.6.6	type6	192.168.7.0	type70	192.168.40.0	type400
192.168.2.0	type20	192.168.8.0	type80	192.168.60.0	type600

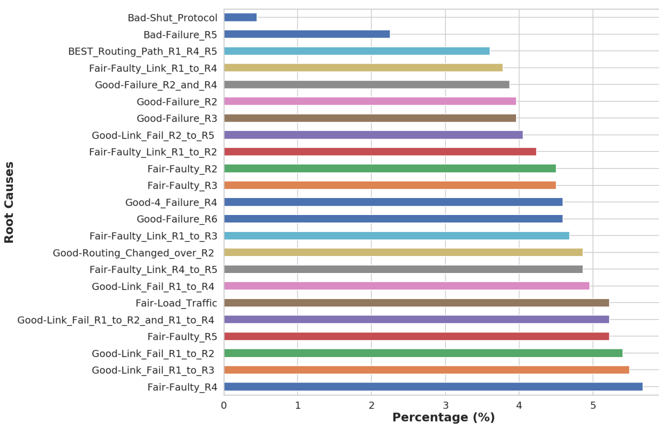


Fig. 4: Root cause measurement distribution of the collected dataset.

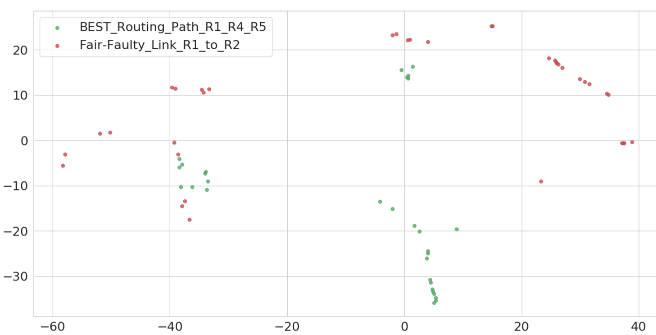


Fig. 5: T-SNE plot of two root causes.

scriptions in our dataset. Some of the features marked with (*) have multiple consecutive values, hence there exists 130 features where 27 of them are shown in Table. I. In our dataset, we have done feature name summarization of next hop and type of the routers where Table III shows the mapping between the IP address versus type used during our feature naming stage for the dataset. Based on our classification metrics, a total of 1110 rows of measurements

TABLE IV
DIFFERENT CLASSIFIERS' TEST ACCURACY, PRECISION, RECALL AND F1 SCORES.

Classifier	Accuracy	Precision	Recall	F1 Score
LR	0.81	0.81	0.81	0.81
DT	0.83	0.83	0.83	0.83
GB	0.88	0.88	0.88	0.88
RF	0.87	0.87	0.87	0.87

of different KPIs exists. Together with our classification algorithms, we try to classify each network measurement obtained on R1 into one of 23 root cases corresponding to each network topology as outlined in Table II.

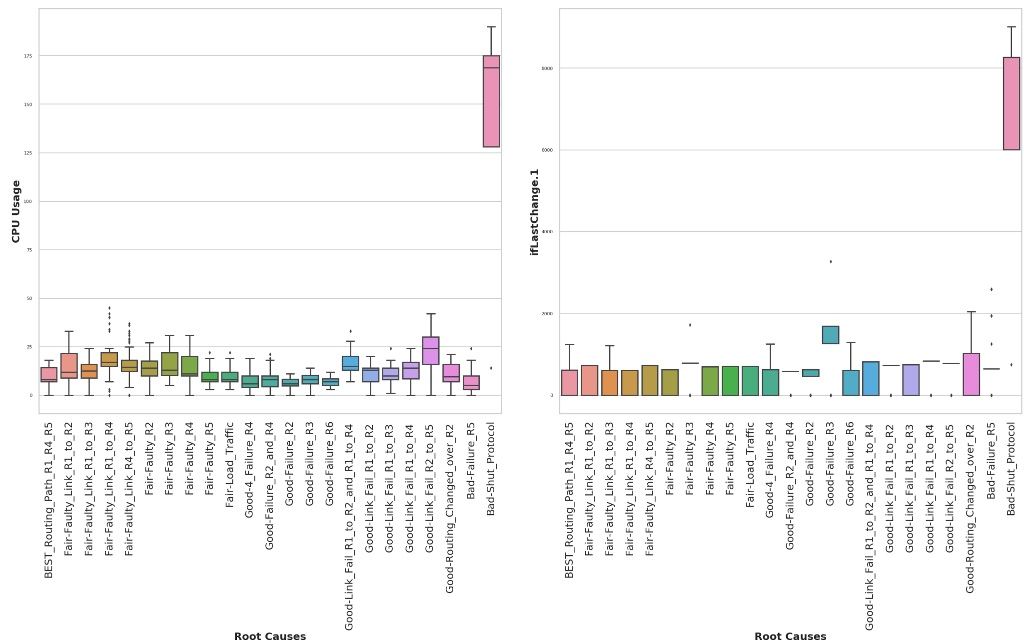
The configuration errors and undesired network topology state changes can be created by MNO's operational units. We've collected the above 130 KPIs for situations where the network works the best and the service works as planned. We then created problems with all nodes and links starting from the primary path. In this way, the problems at the network level carrying the service were obtained as trained data. Node failure, node reboot, link failure, link flap, link error, link packet drop, link delay, routing problems and their variations are some of the network level problems. Service level training data is generated after occurrence of network level problems. These are problems that can occur within the service itself and are affecting the end-users' service. These problems have no effect on network level because the service is already limited to a bandwidth. Service interrupt, service failure and service bandwidth saturation data are generated as service training data.

C. Exploratory Data Analysis

Fig. 4 shows the class distribution percentages inside the dataset. In our dataset, we have 23 distinct root causes where top three root causes $Fair - Faulty\_R4$, $Good - Link\_Fail\_R1\_to\_R3$ and $Good - Link\_Fail\_R1\_to\_R2$ are with 5.68%, 5.5% and 5.4% respectively.

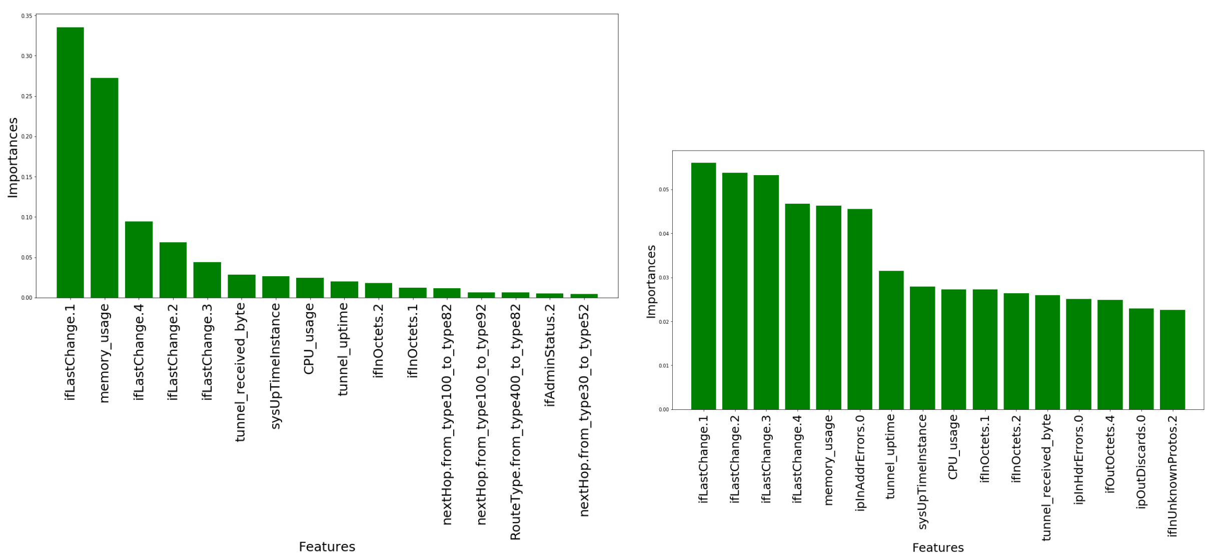
The dimension of the features in our dataset is large to be visualized. Fig. 5 shows the t-Distributed Stochastic Neighbor Embedding (T-SNE) plot of two classes to visualize their nature. T-SNE is a technique that is used for data visualization purposes. For this, we have used top two components with maximum information. Each dot in the Fig. 5 represents a measurement. $BestRoutingPathR1 - R4 - R5$ is represented by green color and $FairFaultyLinkR1toR2$ is represented with red color. From observing this figure, we can observe that some of the measurements are close to each other, hence it is not an easy task to classify all root causes accurately with simplistic models (e.g. using linear classifiers).

In Fig. 6, we plot the boxplot distribution of the two selected KPIs namely $CPUusage$ and $ifLastChange.1$ which have been in top 5 of the feature importance list as given in Fig. 7 in next section. These plots indicate that some distinctive values of the selected KPIs exists that can yield insights into obtaining better classification accuracy.



(a) (b)

Fig. 6: Boxplot over classes (a) CPU utilization (b) ifLastChange.1 feature



(a) (b)

Fig. 7: Top 16 features and their corresponding importance values for the (a) DT classifier. (b) RF classifier.

D. Analysis Results

To analyze the behaviour of our approach, we have performed the classification of each row data generated from different KPI values at each instant t for a given router R1 as shown in Fig 3. Note that without loss of generality, similar classification analysis and observation of data can also be done with observing the data over the other remaining routers. To analyze the accuracy of our model, we conducted a series of experiments. We compare the performances of Logistic Regression (LR) [15], Decision Tree (DT) [16], Radio Frequency (RF) [17] and Gradient Boosting (GB) [18] classifiers over the observed dataset. The 130 KPIs specified for all training data were continuously collected during the entire data collection process. For our analysis purposes, we have standardized the features by removing the mean and scaling to unit variance. We have also selected 80% for training and the remaining 20% of the dataset for testing purposes using stratifiedKfold to create class balanced training and test dataset [19]. To explore the most suitable classifier attributes and hyper-parameters, we utilized grid search.

Fig. 7 depicts the feature importance values of the DT and RF classification algorithms. For DT classifier, the top five important features are $IFLastChange\_1$, $memory\_usage$, , $IFLastChange\_4$, $IFLastChange\_2$ and $IFLastChange\_4$ with importance values of $0.33, 0.27, 0.09, 0.07, 0.04$ respectively and for RF classifier, the top five important features are$ifLastChange.4$, $ifLastChange.3$, $ifLastChange.1$, $memory_usage$ and $ipInAddrErrors$ with importance values of $0.056, 0.0548, 0.0523, 0.0498, 0.0463$ respectively. In this notation, as an example $IFLastChange\_1$ corresponds to first value of the $IFLastChange$ feature.

Finally, Table IV summarizes the accuracy, precision, recall and F1 scores of LR, DT, RF and GB classifiers. The parameters for RF are $n\_estimators = 150$, $max\_depth = 20$, for GB $n\_estimators = 100$, $max\_depth = 3$ and $learning\_rate = 0.1$, for LR $max\_iter = 1000$ and for DT $max\_depth = 14$, $min\_samples\_leaf = 1$ are selected to be after grid search hyper-parameter optimization process. From Table IV, we can observe that GB classifier outperforms the others with 0.88 accuracy followed by RF, DT and LR with 0.87, 0.83 and 0.81 accuracy respectively. When F1 scores are compared, similar alignments of the observed classifiers are observed to occur.

IV. Conclusions

In this paper, we have focused on network service management problem of MNOs. As an attempt to solve this problem, we proposed a ML based framework that helps to measure service quality in predefined states (i.e good, fair, medium and bad), and also determine the different root causes based on these states and KPIs measurements. We established our formal model as a FSM, and presented a novel usage of ML techniques to determine most appropriate states in FSM for undefined input alphabets. The implementation of our approach is done using the data generated from the GNS3 network emulator platform with realistic configurations of switches and routers which is also published into IEEE's DataPort platform. Furthermore, we identified the most significant KPIs that have impact on service quality using various ML approaches on the generated synthetic network data. Our results have indicated that among all investigated classification algorithms, GB has provided higher accuracy and F1 scores with 88%. Our presented approach shows a promising use case of application of ML for automation in network service management.

V. Acknowledgements

This work was partially funded by Spanish MINECO grant TEC2017-88373-R (5G-REFINE) and by Generalitat de Catalunya grant 2017 SGR 1195 and partially supported by The Scientific and Technological Research Council of Turkey in part under 1515 Frontier R&D Laboratories Support Program with project no: 5169902.

References

[1] ETSI Specification, "Zero touch network and Service Management (ZSM); Proof of Concept Framework." https://bit.ly/2k6vusj, 2018. [Online; accessed 21-June-2019].

[2] D. Cote, "Using machine learning in communication networks [invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, pp. D100–D109, Oct 2018.

[3] D. Rafique and L. Velasco, "Machine learning for network automation: overview, architecture, and applications [Invited Tutorial]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 10, pp. 126–143, 2018.

[4] Z. M. Fadlullah et al., "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems," *IEEE Communications Surveys Tutorials*, vol. 19, pp. 2432–2455, Fourthquarter 2017.

[5] S. Ayoubi, N. Limam, M. A. Salahuddin, N. Shahriar, R. Boutaba, F. Estrada-Solano, and O. M. Caicedo, "Machine learning for cognitive network management," *IEEE Communications Magazine*, vol. 56, pp. 158–165, Jan 2018.

[6] M. Zorzi, A. Zanella, A. Testolin, M. D. F. De Grazia, and M. Zorzi, "Cognition-based networks: A new perspective on network optimization using learning and distributed intelligence," *IEEE Access*, vol. 3, pp. 1512–1530, 2015.

[7] W. Wang et al., "A network traffic flow prediction with deep learning approach for large-scale metropolitan area network," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–9, April 2018.

[8] O. Narmanlioglu and E. Zeydan, "Mobility-aware cell clustering mechanism for self-organizing networks," *IEEE Access*, vol. 6, pp. 65405–65417, 2018.

[9] O. Narmanlioglu, E. Zeydan, M. Kandemir, and T. Kranda, "Prediction of active ue number with bayesian neural networks for self-organizing lte networks," in *2017 8th International Conference on the Network of the Future (NOF)*, pp. 73–78, IEEE, 2017.

[10] R. Mijumbi et al., "Design and evaluation of learning algorithms for dynamic resource management in virtual networks," in *2014 IEEE Network Operations and Management Symposium (NOMS)*, pp. 1–9, May 2014.

[11] T. Kawabata, T. Kurimoto, and K. Mizutani, "Toward preventive network service management by neural networks," in *2018 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, pp. 125–126, June 2018.

[12] P. Casas, J. Vanerio, and K. Fukuda, "Gml learning, a generic machine learning model for network measurements analysis," in *2017 13th International Conference on Network and Service Management (CNSM)*, pp. 1–9, Nov 2017.

[13] A. P. Vela, M. Ruiz, and L. Velasco, "Examples of machine learning algorithms for optical network control and management," in *2018 20th International Conference on Transparent Optical Networks (ICTON)*, pp. 1–4, July 2018.

[14] Yekta Turk and Engin Zeydan and Zeki Bilgin , "The Good, The Bad and The Fair: KPIs from Network Elements, IEEE Dataport." dx.doi.org/10.21227/1bm6-wa12.

[15] Scikit-learn, "Linear classifiers (SVM, logistic regression, a.o.) with SGD training." https://bit.ly/2Iu28NQ, 2019. [Online; accessed 08-April-2019].

[16] Scikit-learn, "Decision Tree Classifier." https://bit.ly/2WEHy2E, 2019. [Online; accessed 08-April-2019].

[17] Scikit-learn, "Random Forest for Classification." https://bit.ly/2C3ICoi, 2019. [Online; accessed 08-April-2019].

[18] Scikit-learn, "Gradient Boosting for Classification." https://bit.ly/2IVUrz8, 2019. [Online; accessed 08-April-2019].

[19] Scikit-learn, "Stratified K-Folds cross-validator." https://bit.ly/2KtkH7i, 2019. [Online; accessed 08-April-2019].