Diagnosing Failures in the Mobile Network Operation using Ensemble of Classifiers

Fabio Oliveira dos Santos, Carlos Salvador Gomide Richa, Dianne Scherly Varela de Medeiros,
Diogo Menezes Ferrazani Mattos, and Natalia Castro Fernandes
Midiacom - PPGEET
Universidade Federal Fluminense
Niterói/RJ, Brazil
{menezes,natalia}@midiacom.uff.br, {santosfabio,salvador_richa,diannescherly}@id.uff.br

***Abstract*—Mobile networks combine different communication technologies sharing the same infrastructure. Although each technology accomplishes different networking requirements, it also hinders the network operation. Standard network monitoring tools often fail to detect service faults due to the multitude of monitored parameters of hardware from different vendors. To overcome this issue, we propose the Advanced Infrastructure for fault Diagnosis in Mobile Networks (AID-MN). AID-MN relies on applying ensemble of classifiers, a machine-learning technique, on fault postmortem data to automatically recognize the causes of service faults. We evaluate the performance of AID-MN in a case study of an operational access mobile network aiming at diagnosing intermittent 2G service failures. The results show that AID-MN assists the detection of the fault cause and supports to offer a feasible solution to the fault in a fast and effective way. Any service that have information about your status can be used with our solution.**

***Index Terms*—Mobile networks, monitoring, machine learning, service fault diagnosis**

I. Introduction

The complexity of mobile networks continually increases due to the coexistence of several generations of mobile network technologies, known as multi-generation mobile networks. The consequence is the existence of a diversity of elements, protocols, and systems, required to provide current service applications and sharing the same infrastructure [1]. A complex routing structure in the access network connects a large number of elements in mobile networks. The access network, however, needs to accommodate interface types and network traffic from different mobile network generations [2].

The inherent complexity of mobile networks hinders their operation and management due to the multitude of coexisting hardware and software generations. Even though network management and operation tools provide essential information about the status of the network infrastructure and services, it is common to observe total or partial faults of the services. This behavior happens because the management and operation tools cannot provide accurate information regarding the faults [3]. A solution to overcome the lack of accurate information is to inspect the network manually. Nevertheless, manual inspection is costly and requires a highly skilled workforce, becoming impractical as the fault scales. Moreover, it is hard to find a fixed-model solution for all services that run across the network. Hence, it is necessary to develop automated tools to help on diagnosing faults in large scale mobile networks [4].

In this work, we propose the Advanced Infrastructure for fault Diagnostics in Mobile Networks (AID-MN) system. The system relies on the ensemble of classifiers technique and it aims at detecting the causes of network service faults. We evaluate the use of three ensembles of classifiers algorithms, Gradient Boosting Trees, Ada Boost, and Random Forest, to obtain better monitoring of critical services. We address the crucial issue that critical services require high network performance in scenarios where traditional management tools do not provide enough information for assuring correct operation of the network. AID-MN collects and stores performance data from mobile services and network information, such as network topology used by each service, network variables related to the topology, and network element information. Then, classification algorithms run over these data to diagnose which network variables have the highest correlation with a service fault. As a consequence, our proposal converts the task of detecting and diagnosing faults in an automated standard classification problem.

We evaluate AID-MN against a case study of fault diagnosis in Circuit EMulation (CEM) services for 2G networks. These networks emulate a Time Division Multiplexing (TDM) channel over the IP/MPLS infrastructure to enable 2G to coexist with the modern 3G/4G network infrastructure. It is noteworthy that 2G service is still widely used in the Machine-to-Machine (M2M) communications, such as the credit card operation service. We evaluate AID-MN in an operational network, in which we run several tests focused on the analysis of a usual outage of the 2G service. The outage is transparent to network monitoring tools instigating an intermittent behavior of the failure. The results show that AID-MN enables to discover the cause of the fault and supports the diagnosis of service fault in complex mobile networks.

The remainder of the paper is organized as follows. Section II contextualizes multi-generation mobile networks. Section III discusses related work. Section IV presents the AID-MN proposal and Section V describes the evaluation results using a case study. Finally, Section VI concludes the paper.

This work is supported by CNPq, CAPES and FAPERJ.

978-3-903176-23-2 © 2019 IFIP

II. Multi-generation mobile networks

The adoption of 4G technology brought significant advances to the mobile network infrastructure. However, there is still a need for operation and maintenance of previous generation mobile networks (2G/3G), since several customers continue to operate with mobile terminals that rely on these technologies. According to the GSMA report [2], 2G technology still accounts for 29% of global mobile network connections, compared to 28% 3G and 43% 4G. The same report points out that the technologies must still coexist over a long period. The report includes the usage of both Machine-to-Machine (M2M) and personal cell phones. In Brazil, telecommunications companies still preserve M2M clients and keep the 2G network available with the same coverage. Indeed, nowadays, 2G networks provide a more reliable service, since cell phones no longer use this technology as they migrated to 3G or 4G access technologies. Therefore, 2G networks are almost exclusively for M2M clients and, thus, there is currently no plan to shut down the 2G network. Current mobile network scenario still includes all generations from 2G up to now.

The core IP network provides connectivity between the 3G and 4G mobile network elements, but deterministic 2G traffic also shares the same IP network core. The legacy base transceiver stations (BTSs) of the 2G mobile network only deploys deterministic interfaces, usually using Time-Division Multiplexing (TDM) in the pulse-code modulation pattern (PCM-30), which groups 32 channels of 2,048 Mb/s. Two Of these 32 channels are usually control and signaling channels, with 30 channels available for transporting useful information from users. Thus, there is still a need for a network for 2G traffic, which has strong time constraints in order to keep the deterministic behavior.

The core of transport networks used for mobile telephony has also evolved over the last years, with newer technologies available for packet switching. Currently, transport networks use IP/MPLS technology as the transport network for 2G, 3G, and 4G services. As a consequence, it requires the implementation of emulation services of deterministic circuits over IP/MPLS packet network. The services to emulate deterministic circuits on an IP/MPLS network are the Circuit Emulation over Packet (CEoP) and Structure-agnostic TDM over Packet (SAToP), usually called CEM. Both CEoP and SATop define means to provide transport by Timex Division Multiplexing (TDM) through MPLS packet networks. SAToP is the standardized name for unstructured transport while CEoP is often used to transport circuits whose payload is structured. With the use of emulation technologies, it is possible to transport TDM frames, usually carried by a deterministic Plesiochronous Digital Hierarchy (PDH) or Synchronous Digital Hierarchy (SDH) network, in an IP/MPLS, thus avoiding the problematic maintenance of various physical circuits of a deterministic network. Traditional TDM transport using deterministic networks involves a large number of dedicated circuits physically provided through dedicated and often non-optimized transmission media.

III. Related Work

Previous work focused on using machine learning algorithms in the verification of anomalies in IP networks [3], [5]–[9]. Zubair et al [10] demonstrate the use of a framework for evaluating real-time Netflow data to detect anomalies in the network traffic pattern. While this approach is useful for checking traffic anomalies, using only NetFlow collections can lead to inaccuracies in results, due to large sampling intervals. Chen *et al.* [3] propose to train decision trees on the request traces from periods when users experience failures on the *ebay* website. The proposal identifies paths through the tree and ranks them according to their degree of correlation with failure, and it merges tree nodes according to the observed partial order of system components. Although the proposal signs a possible use of machine learning techniques on failure diagnosis, the proposal considers only the C4.5 tree training algorithm, and it creates a single tree per failure. On the other hand, our proposal evaluates three different classification algorithm, and we estimate the influence of each feature on the failure according to the information gain on the classification.

There are also efforts to propose machine-learning based systems capable of managing and operating the network without any human intervention [11]–[15]. Current network environments are complex, and translating this complexity into router and switch configuration files is a task which will probably lead to a reasonable amount of human errors. However, whenever a new service or client is added to the network, changes in these configuration files are required. Thus, machine learning systems can be useful in standardizing and verifying deployment of network services, thus avoiding much of these errors. Such systems would also be able to predict and directly support the fault diagnosis tasks in the networks [11], [16]–[19]. However, this methodology runs up against the difficulty of obtaining standardized and easily accessible information from the network elements.

Also, there are proposals using machine learning to estimate the probability of failure in specific components of the network [20]–[22]. These approaches base the prediction in the relationship between environmental factors and network usage. In these cases, environmental and usage factors can be taken into account during the training phase of the machine learning algorithm so that this algorithm can predict failure for specific parts of the network.

Several works apply algorithms, such as Feature Selection and Principal Component Analysis (PCA), on network flows to automatically classify applications and flow patterns [23]–[25]. Such algorithms can be used to relate network variables to fault patterns, as is the goal of our work.

Machine learning-based applications are also used for traffic classification to define QoS patterns [26]–[28]. The increasing use of unknown ports as well as the use of cryptography in different application flows tends to limit current traffic filtering/classification models. Hence tools able to classify packages and flows in real time are necessary not only for improvements in network design but also for daily operation and definition of quality of service policies. The use of machine learning algorithms in the creation of a system that can perform real-time classification of network traffic has been proposed and evaluated [29]–[31].

In this work, we use machine learning algorithms to correlate network variables with the degradation and unavailability of mobile services running over a TCP/IP network. AID-MN collects available network variables and uses machine learning algorithms to obtain the relationship between the variables and the service states at the time of collection.

IV. Advanced Infrastructure for fault Diagnosis in Mobile Networks (AID-MN)

We propose a system to diagnosis service faults in multi-generation mobile networks, namely Advanced Infrastructure for fault Diagnosis in Mobile Networks (AID-MN). The proposed system uses machine learning algorithms to obtain relevant information from network variables. These variables may be directly related to the unavailability or degradation of mobile services. AID-MN is suitable in scenarios where the user quality of experience is affected, even though the standard network monitoring tools show the network is correctly operating. Hence, our proposal can aid the diagnosis of service faults when standard monitoring tools are not able to automatically detect the problem.

AID-MN enables the collection of network and service information automatically. Network information comprises, for instance, topology, delay between elements, and usage of QoS queues. Service information, in turn, comprises packet loss counters and service availability. AID-MN correlates network variables to service status. The identification of these variables and of the relationships between them and the service is crucial to the diagnosis of the service fault, allowing to discover the root cause of the fault. In addition, it provides information to plan expansions and renewals, and to improve the performance of the services provided over the mobile networks.

Figure 1 shows the basic diagram of the proposed system. AID-MN is composed of three subsystems, which interact to collect and process network data. The *Collection Subsystem* creates data inventories about the network and its services. The information collected covers both network elements and services. The *Control and Storage Subsystem* has modules that perform the control and storage of the Collection Subsystem inventories. The main functions of this subsystem are to initialize all the modules of the other subsystems in the correct order and to check the availability of the information required by each module. The storage module provides a simple function to store all the data generated by the collection modules. The *Analysis and Evaluation Subsystem* provides modules that use the information stored in the inventories and implements machine learning algorithms. The algorithms are used to obtain the weight information of the importance of each variable used by the machine learning algorithms to find the relationship between variables and the service status.

A. Collection Subsystem

The Collection Subsystem is composed of a set of modules that access network devices using SNMP and CLI *Command-line interface* and store the collected data. The use of SNMP and CLI gives AID-MN the ability to work in multiplatform environments as it access equipment from different vendors, since these equipment support SNMP and CLI. This subsystem is also scalable, as mobile networks usually have large-scale access networks. The scalability of collection subsystem is archive with multithreading of components of SNMP ans CLI to collect a lot of data independent of type of equipment and vendor at same time It is composed of three modules to provide its functionalities.

1) Inventory Generation: We generate an inventory that contains all the network devices used in the collection. To this end, the Inventory Generation module performs two tasks:

- *Access Network Inventory Generation*: the system generates a list with all the routers IPs. The list is filled through queries to the database of the management systems, limiting the search to routers that are involved with the active services in the monitored region.
- *Core and Distribution Networks Inventory Generation*: the system creates a list of routers IPs that perform network aggregation functions, *i.e.*, routers that have more than one IGP routing protocol or more than one internal area. This list is also restricted to routers in the area under monitoring.

2) Service Data Collection: The Service Data Collection module uses the information generated by the Inventory Generation module of the access elements to verify all active emulation services on the network, the IPs of the routers used to establish this service, as well as the performance counters of each service.

3) Network Data Collection: The Network Data Collection module performs collections of different network variables correlated to devices, paths, topologies, services, and packet switching/routing performance. The main tasks of this module are the following:

- *Device Information*: the system creates a dictionary that has the device IP as key and a tuple containing the hardware version, software version, and router name as values. We obtain this information through the Ansible framework, which allows to get several information about the network elements via the Simple Network Management Protocol (SNMP).
- *Active IP information*: Ansible is used to obtain information from the network devices via the command-line interface (CLI). From this collection, a dictionary is created containing the identification IPs of the routers as a key and, as values, a list with all the interface IPs of these routers. This dictionary correlates all the IPs in the network and with the respective elements.
- *Topology Generation*: AID-MN discovers network topology at the time of collection using SNMP. SNMP obtains this information through queries to the Open Shortest

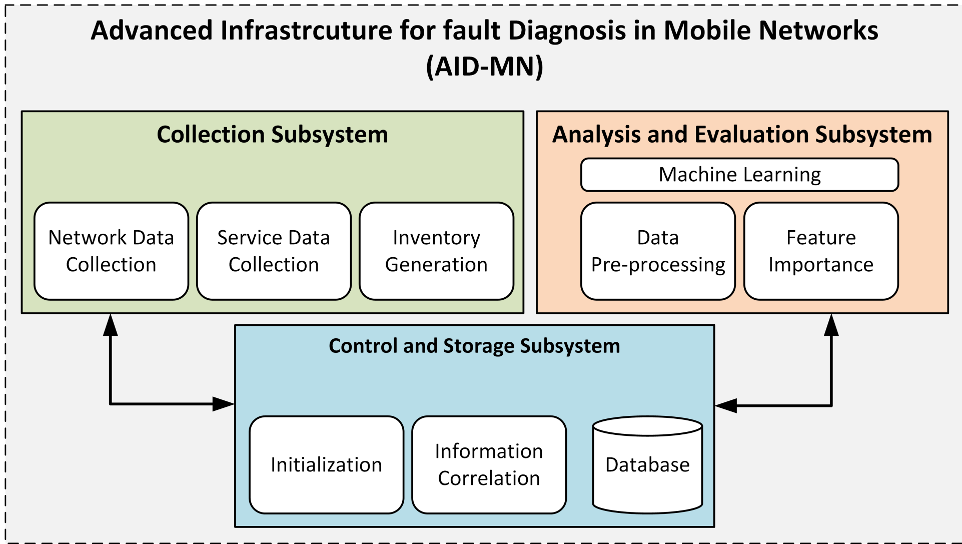


Fig. 1: Architecture of the proposed system for aiding the fault diagnosis in mobile networks. AID-MN presents three subsystems for collection, analysis and evaluation, and storage and control.

Path First (OSPF) Management Information Base (MIB) of all OSPF processes on the network. Topology generation is performed using Python's NetworkX [32] library. From the network point of view, the same framework used by the OSPF protocol for broadcast networks (Designated Router (DR) election and DR IP association in zero cost topology) is used to interconnect all IP addresses of an element to its management IP, since it is common to have more than one process of IGP routing in the access networks. As a consequence of this type of configuration, a router has more than one routing ID (RID). Our system creates a connection with zero cost between the IPs of the same element. To generate the topology we use the dictionary that contains Active IP Information to perform this task. Thus it is possible to create a complete and coherent view of the network. Topology is stored as a Pickle file, which saves the topology in the form of a database for graphs.

- *Path Generation*: the system relates the source and destination IPs of the emulation services to the paths used in the topology for each service, using the Shortest Path First (SPF) algorithm in the topology generated in the Topology Generation task. The Path Generation checks the IPs used by each service and associates them with the networks used to interconnect the routers used along the path of the service.
- *Probe Management*: AID-MN generates a dictionary containing as key the destination IP of the monitored service for each router that has an active service and associates it to a probe. The probes provide the proposed monitoring system with packet loss and delay information in the paths used by the monitored service.
- *Path State*: interface and path information available from the Path Generation task is used to perform usage collection of the QoS queues of all interfaces used by all routers in the services paths. Thus, the collections are restricted only to the interfaces that are used for packet switching for the monitored services. Hence, for each path, the following information is generated: minimum length of the queue, maximum queue usage, and minimum queue usage.
- *Delay and Packet Loss*: information from the Probe Management and Topology Generation is used to obtain the closest probe to the service destination. This probe is used to get delay and packet loss information between the routers executing the monitored service.
- *PTP Collection*: the Precision Time Protocol (PTP) Collection uses the information generated by the Inventory Generation and Device Information to perform SNMP or Ansible queries to obtain status synchronization of routers. This information is essential when monitoring the link emulation service. The SNMP queries are preferred over the use of Ansible CLI since they usually require a smaller amount of resources to be executed in routers.

B. Control and Storage Subsystem

The Control and Storage Subsystem modules are responsible for performing two tasks:

- Initiate the information collections in the correct order, given that some modules depend on the information that is obtained by other modules. Figure 2 shows the data flow in AID-MN Collection subsystem.
- Correlate the status information generated by the various modules to obtain an integrated and relational information base.

C. Analysis and Evaluation Subsystem

The information generated by the Collection Subsystem must be evaluated and processed to become useful to network

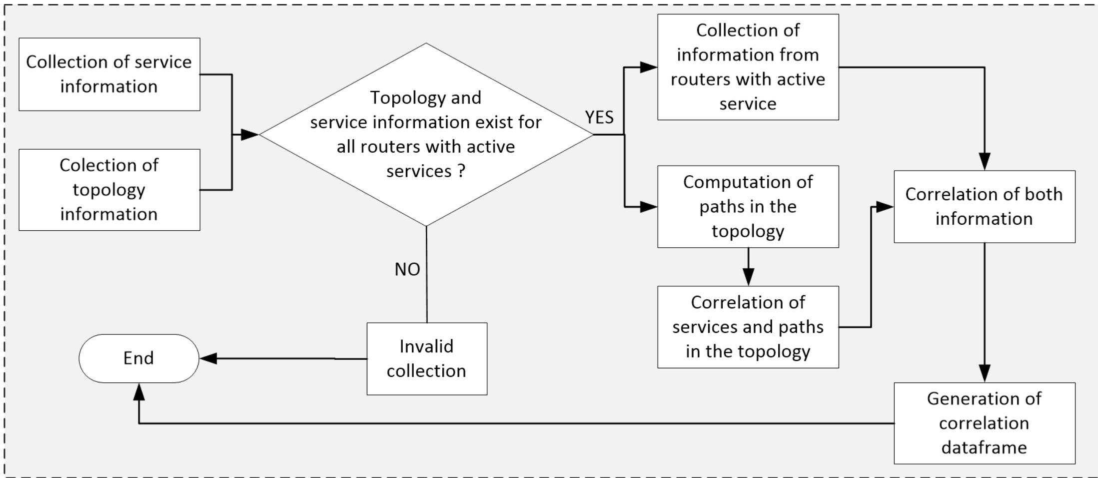


Fig. 2: AID-MN data flow.

administrators. The tasks of the Analysis and Evaluation Subsystem are to verify the availability of the information and apply the collected data to the machine learning algorithms, to correlate network variables and service status. This subsystem is composed of two modules to provide its functionalities.

1) Data pre-processing: The data collected, generated, and correlated by the AID-MN subsystems generates an n-dimensional data matrix, stored using the Pandas [33] DataFrame in the implementation. Each collection creates an array, where the lines are the various monitored services and the columns, the network variables associated with the state of the service. As a result of each monitoring action, AID-MN evaluates the network variables, correlating them with the state of the monitored services. The data pre-processing module is responsible for using preparing the collected data to use them as input to the machine learning algorithms.

In machine learning algorithms, each variable presented is called *feature*, and the result or output is called *target*. Thus, in AID-MN, the network variables are called features, and the service states are the targets. When creating a machine-learning-based system to diagnose network faults, not only is forecast accuracy important, but also which variables led the model to achieve that result. Therefore, it is crucial to link the main network variables related to failures. During the process of applying machine learning algorithms, the number of features should be minimized to simplify the process of identifying the most important features. The more features used, the more complex the model become. Selecting and restricting the features is called feature engineering. Without this process, models become more complex and error-prone, not only because of the complexity but also because of the variability of the various variables.

Due to the need to reduce the number of input variables and the great importance of the information that relates each variable to the result, several models offer information that describes the impact of each feature with its output. The availability of information regarding the importance of each feature is widespread in decision tree-based algorithms, such as Random Forest, Gradient Boosting, and Ada Boost.

The application of feature importance in the telecommunication networks helps service providers to achieve a better quality of experience for users. Mobile networks have a large number of elements and are very complex. Today, the various network variables are collected and analyzed by analysts in search of solutions for current faults in the networks. However, this work is usually reactive in terms of identifying an already active fault in the service. Although there are predictive and automatic tools in several networks, correlation in search of the root cause is usually performed in a time-consuming and manual way.

The proposed system aims to reduce the need for manual fault diagnosis. Using AID-MN, analysts and engineers can quickly identify the root cause of the problem and work on the network variable that is directly related to the failure under analysis.

2) Feature Importance: Each decision tree model has specific peculiarities in how to calculate the importance of each variable. However, generically, the seriousness with which each variable relates to the others and mainly to the result is linked to the fraction of times that the variable is used in the tree for decision making. In this way, the variables that are in the nodes closest to the top of the tree are more used for decision making than the variables that are in the nodes located in the bottom part of the tree. Besides the position of the variables in the tree, the degree of impurity of the decision are also taken into account when calculating the importance of the variable. Hence, this module is responsible to determine the importance of each network variable to the classification.

AID-MN works using the premise that the service status is known, but network failure root causes may be difficult to find. Hence, using the service status and corresponding values of network variables, it is possible to build a tree that points out which are the network variables that most relate to the service fault.

V. Case Study

The case study using AID-MN occurred in an operational multi-generation mobile network. In this network, frequent degradation was noticed in the 2G service, although the transport network did not present flaws. Thus, AID-MN was used to verify the root cause of this problem. The evaluation was carried out in a specific region with 1150 active routers with approximately 600 TDM emulated circuits, which is a requirement for 2G networks that run over IP/MPLS transport networks. Since it is a network in operation, there are restrictions on access and availability of the server and network resources, which has restricted the collection interval to every 30 minutes on a limited number of devices. This interval is a function of the available machine resources to run the proposed system as well as the consumption of network resources. As the monitoring consists of querying simple commands and using SNMP, we did not observe significant increases in resource consumption in network elements.

In our analysis, we applied the network variables in different machine learning algorithms to evaluate the accuracy and precision of the results.

A. Machine Learning Algorithm Choice

In this work, we use different machine learning algorithms with the collected data, to observe the algorithm accuracy related to the service state. The three algorithms that presented better precision were used to obtain the weights of each variable and thus to identify the relations of the variables with the service state. The classifiers are trained with sample of network variables and the status of services with these collect network variables. As the proposal of the study is to find the relationship between the variables and the service status, we choose the algorithms that obtained greater precision. It is important to emphasize that the purpose of the proposal is not to get generic modeling that can observe a new sample and define whether the 2G service using circuit emulation presents degradation. We don't need to predict the state since the service status is already available during collection. The purpose of AID-MN is to enable operators to quickly and accurately identify network problems that affect the offered services but are not clear using standard monitoring tools, and this goal with on good relation between network variable states (X on our Dataset) and service status (Y on our Dataset).

Firstly AID-MN was put into operation in the network to collect traffic in the presence or not of service failures to train the algorithms. Our network uses five QoS classes, namely Real Time, Signaling, User Data, Management, and Default. These collection and usage is based on real network and the classes and number of QOS queue are already in use on real network.

We performed precision tests on the machine learning algorithms, using the QoS information presented in Table I. AID-MN collects data for all QoS queues configured in network devices. Given the nature of the QoS rules, some of this

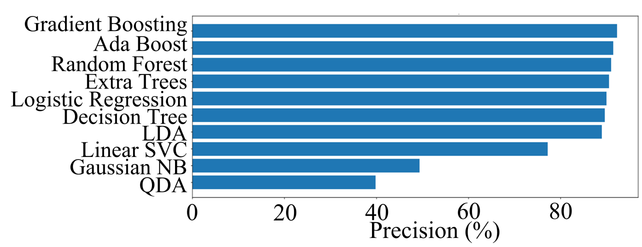
TABLE I: Variables collected for each QoS queue, considering time intervals of 30 s.

Variable	Definition
queue_30_drop_rate_bps	Drops in bps
queue_30_off_rate_bps	Received traffic in bps
queue_NoBuffer	Number of buffer overflows
queue_bytes	Average queue size in bytes
queue_deep	Automatic queue expansions to reduce discards
queue_drop	Number of discards
queue_limit	Queue size
queue_packets'	Number of packets transmitted through the queue

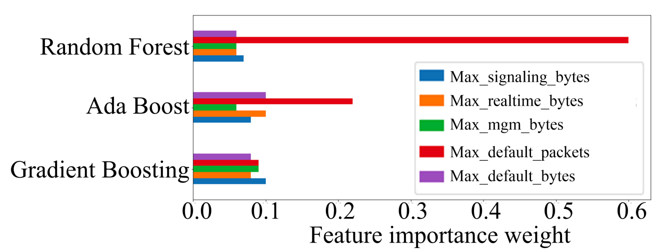
information may be non-existent or unused for a particular type of queue.

The monitored data is standardized to be processed by machine learning algorithms. Figure 3a shows the obtained algorithm precision. We observe that the algorithms Gradient Boosting Classifier, Ada Boost Classifier, and Random Forest Classifier obtained better results. When observing the values of feature importance for the three algorithms that presented better performance, it is noticed that the features of QoS queues received greater importance, as can be observed in Figure 3b.

As the results pointed to the QoS queues as the cause of the service degradation, it was still necessary to find out which queue was most related to the problem. Thus, we aggregated the feature importance of each feature for each queue so that each queue could reflect the total importance value of its features. Indeed, this was important as the three algorithms that obtained the highest indexes of precision did not point the same features with the greatest importance in the same class



(a) Precision of the studied machine learning algorithms.



(b) Feature importance of the machine learning algorithms that presented better precision.

Fig. 3: Machine learning algorithm evaluation using AID-MN, considering both precision and feature importance analysis.

TABLE II: Queue weight defined by the machine learning algorithms.

Algorithm	Queue	Weight	# of features
Gradient Boosting Classifier	Real Time	0.20	3
	Signaling	0.21	3
	Management	0.15	4
	User Data	0.12	4
	Default	0.21	6
Ada Boost Classifier	Real Time	0.14	3
	Signaling	0.16	3
	Management	0.12	1
	User Data	0.20	4
	Default	0.34	3
Random Forest	Real Time	0.18	3
	Signaling	0.16	7
	Management	0.16	4
	User Data	0.23	7
	Default	0.17	7

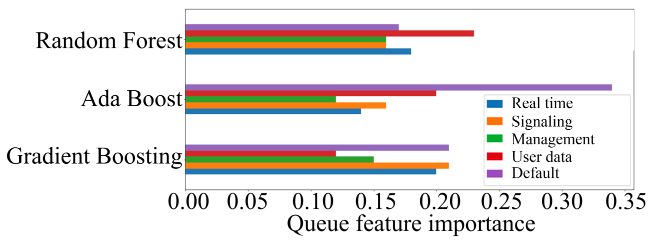


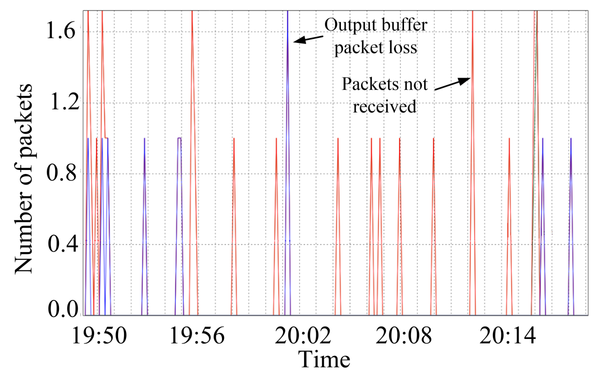
Fig. 4: Most important features considering QoS queue parameter aggregations.

of QoS. Figure 4 shows the result of this analysis. We observe that the two algorithms that obtained the best precision indexes pointed the QoS queue "Default" as the most important queue. Table II summarizes the observed results.

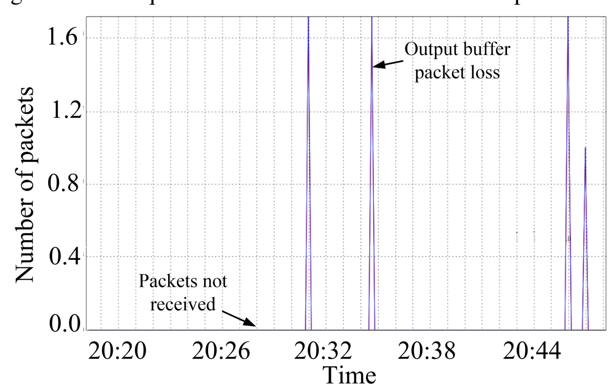
The results indicate that the Default queue is strongly related to the circuit emulation service failure. Hence, the Default traffic is influencing the emulation services, even though the emulation service uses the Real-Time queue, which is the QoS highest priority class. Based on this result obtained with AID-MN, we analyze the network observing the indicated characteristic. Thus, to verify the effects of the traffic that travel in the non-real-time queues on the emulation service, we reduced the size of the Default queue in all elements of the service path. In the network used in the case study, the Default queue usually has a limit of 8k packets, and as we see in Figure 5a, with this packet limit, we observe the circuit emulation service failures. With this result, and more investigation on default queue some UDP burst traffic was found on default classes. These traffic are related with one wrong service deployment.

We also observed in the results of AID-MN that the second variable with the highest correlation with the state of the service is LINK_CAPAC, which is associated to paths that do not have only optical links. This result suggests that these different transmission mediums insert delays of small magnitude in the real-time queue due to the traffic pattern in the default queue. Such delays are quite minor and do not generate packet losses as perceived by the results of the probes collections, but can be correlated with the emulation service failures.

Hence we empirically reduced the limit of the Default queue by 50%, that is, it passed to 4000 packets only on the non-100% optical paths towards the network core used by the service of circuit emulation. After this reduction, we noticed better performance in the counters of the emulation services. Although there is no loss in any of the queues, we realized that, after limiting the Default queue, the desired performance was achieved for the emulation services, as shown in Figure 5b.



(a) Circuit emulation service for 2G networks with degradation when using the default queue with a maximum size of 8000 packets.



(b) Circuit emulation service for 2G networks without degradation when using the default queue with a maximum size of 4000 packets.

Fig. 5: Service status before and after the use of AID-MN to discover the root key of failures on the circuit emulation service on the 2G networks.

VI. Conclusions

We proposed the Advanced Infrastructure for fault Diagnosis in Mobile Networks (AID-MN), which is a machine learning based system to automatically diagnose faults in complex multi-generation mobile networks. AID-MN collects information about routers and services, surveying the status of network variables. Then, it correlates this data with the service faults to find the cause of such faults. This allows network operators to apply the needed countermeasures to mitigate the service faults. We designed and implemented AID-MN, further applying it in a case study in which we diagnose faults in the Circuit EMulation (CEM) service (2G network) in an operational access network. We observed that AID-MN can find a number of direct relationships between the several network variables and the status of the CEM service. The proposed system can also identify nontrivial relationships with a reasonable margin of precision. Such relationships can be used to diagnose network faults that are not observed by traditional monitoring tools, aiding to operate the access networks used in mobile telephony.

As future work, it is necessary to consider and analyze which variables need to be effectively monitored to quickly obtain network information without overloading network elements and systems. The larger the number of variables available for the machine learning algorithms, the more complex the models generated by them. On the other hand, the use of a large number of variables increases the probability of having access to the variable that most relates to the problem in the network variable set. We also intend to apply AID-MN to diagnose the root problems of other services in multi-generation mobile networks. Additionally, we will evaluate other machine learning algorithms and techniques to select network variables. The application of the proposed system for other services of the network core is also a target for future analyses.

References

[1] M. Hadzialic, B. Dosenovic, M. Dzaferagic, and J. Musovic, "Cloud-ran: Innovative radio access network architecture," in *Proceedings ELMAR-2013*, Sep. 2013, pp. 115–120.

[2] G. Association, "The mobile economy," *GSM Intelligence Journal*, 2019.

[3] M. Chen, A. X. Zheng, J. Lloyd, M. I. Jordan, and E. Brewer, "Failure diagnosis using decision trees," in *International Conference on Autonomic Computing, 2004. Proceedings.*, May 2004, pp. 36–43.

[4] G. F. Ciocarlie, U. Lindqvist, S. Nováczki, and H. Sanneck, "Detecting anomalies in cellular networks using an ensemble method," in *Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013)*, Oct. 2013, pp. 171–174.

[5] M. Zhu, K. Ye, and C.-Z. Xu, "Network anomaly detection and identification based on deep learning methods," *Lecture Notes in Computer Science*, vol. 10967, 2018.

[6] J. Zhang, R. Gardner, and I. Vukotic, "Anomaly detection in wide area network meshes using two machine learning algorithms," *Future Generation Computer Systems*, 2019.

[7] M. Nawir, A. Amir, N. Yaakob, and O. B. Lynn, "Effective and efficient network anomaly detection system using machine learning algorithm," *Bulletin of Eletrical Engineering and Informatics (BEEI)*, 2019.

[8] X. Dai, N. Wang, and W. Wang, "Application of machine learning in BGP anomaly detection," *Journal of Physics: Conference Series Robotics and Artificial Intelligence*, 2019.

[9] G. F. Jr, J. J. P. C. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. P. Jr., "A comprehensive survey on network anomaly detection," *Telecommunication Systems*, 2019.

[10] S. Zhao, M. Chandrashekar, Y. Lee, and D. Medhi, "Real-time network anomaly detection system using machine learning," in *2015 11th International Conference on the Design of Reliable Communication Networks (DRCN)*, Mar. 2015, pp. 267–270.

[11] 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*, pp. 158–165, 2018.

[12] Y. Xiao, S. Nazarian, and P. Bogdan, "Self-optimizing and self-programming computing systems: A combined compiler, complex networks, and machine learning approach," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2019.

[13] G. Saadon, Y. Haddad, and N. Simoni, "A survey of application orchestration and oss in next-generation network management," *Computer Standards & Interfaces*, 2018.

[14] H. Gacanin, "Knowledge-driven wireless networks with artificial intelligence: Design, challenges and opportunities," *Computer Science*, 2018.

[15] D. Lynch, M. Fenton, D. Fagan, S. Kucera, H. Claussen, and M. O'Neill, "Automated self-optimization in heterogeneous wireless communications networks," *IEEE/ACM Transactions on Networking*, 2019.

[16] J. N. . N. Adams, "Real-time dynamic network anomaly detection," *IEEE Intelligent Systems*, 2018.

[17] A. Putina, S. Barth, A. Bifet, D. Pletcher, C. Precup, P. Nivaggioli, and D. Rossi, "Unsupervised real-time detection of bgp anomalies leveraging high-rate and fine-grained telemetry data," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Apr. 2018, pp. 1–2.

[18] J. Park, D. H. Choi, Y.-B. Jeon, Y. NamMin, and H.-S. Park, "Network anomaly detection based on probabilistic analysis," *Soft Computing*, 2018.

[19] X. Miao, Y. Liu, H. Zhao, and C. Li, "Distributed online one-class support vector machine for anomaly detection over networks," *IEEE Transactions on Cybernetics*, 2019.

[20] Z. WANG, M. ZHANG, D. WANG, C. SONG, M. LIU, J. LI, L. LOU, , and Z. LIU, "Failure prediction using machine learning and time series in optical network," *Optics Express*, vol. 25, no. 16, pp. 18 553–18 565, 2017.

[21] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin, and J. Aguilar, "Towards the deployment of machine learning solutions in network traffic classification: A systematic survey," *IEEE Communications Surveys & Tutorials*, 2018.

[22] G. Sun, L. Liang, T. Chen, F. Xiao, and F. Lang, "Network traffic classification based on transfer learning," *Computers & Electrical Engineering*, 2018.

[23] N. Williams, S. Zander, and G. Armitage, "Evaluating machine learning algorithms for automated network application identification," *CAIA Technical Reports*, pp. 1–14, 2006.

[24] Z. Wu, Y. Dong, L. Yang, and P. Tang, "A new structure for internet video traffic classification using machine learning," in *Proceedings of 2018 Sixth International Conference on Advanced Cloud and Big Data (CBD)*, 08 2018, pp. 322–327.

[25] H. Lim, J. Kim, J. Heo, K. Kim, Y. Hong, and Y. Han, "Packet-based network traffic classification using deep learning," in *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Feb 2019, pp. 046–051.

[26] A. Saeed and M. Kolberg, "Towards optimizing WLANs power saving: Novel context-aware network traffic classification based on a machine learning approach," *IEEE Access*, 2018.

[27] I. Possebon, da Silva, Anderson, L. Granville, A. Schaeffer-Filho, and A. Marnerides, "Improved network traffic classification using ensemble learning," in *IEEE Symposium on Computers and Communications (ISCC) 2019*. IEEE, 4 2019.

[28] D. A. Hughes, "Multi-level learning for classifying traffic flows," *United States Patent Vasseur et al.*, no. US10257082B2, 2017.

[29] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, vol. 10, no. 4, 2008.

[30] K. L. Dias, M. A. Pongelupe, and W. M. C. L. de Errico, "An innovative approach for real-time network traffic classification," *Computer Networks*, 2019.

[31] P. Wang, X. Chen, F. Ye, and Z. Sun, "A survey of techniques for mobile service encrypted traffic classification using deep learning," *IEEE Access*, 2019.

[32] Aric Hagberg, Dan Schult, and Pieter Swart, "Networkx."

[33] Python / Pandas community, "Pandas." [Online]. Available: https://pandas.pydata.org