Framework for Cognitive Self-Healing of Real Broadband Networks

Enock Cabral Almeida Vieira[1(✉)], Paulo Carvalho[2], and Flávio de Oliveira Silva[1,2]

[1] Federal University of Uberlândia, Uberlândia, MG, Brazil
{enock.vieira,flavio}@ufu.br
[2] Centro Algoritmi, University of Minho, Braga, Portugal
{pmc,flavio}@di.uminho.pt

Abstract. With the growing volume of data traffic demanded by corporate, business, and retail consumers, telecommunications operators are becoming an increasingly important player in the world economy. However, the operators must prepare themselves with solutions that allow dealing with incidents more quickly or even avoid them, always focusing on maintaining an acceptable customer service level. In this context, Self Healing (SH) solutions, supported by Machine Learning (ML) mechanisms, emerge as possibilities to address this challenge. This work presents a cognitive self-healing framework for telecommunications operators. This framework encompasses self-diagnosis, analysis, and automatic actuation for failure mitigation in fiber broadband telecommunications based on Gigabit Passive Optical Network (GPON). In addition, we did an experimental evaluation using a dataset from the operators' modems extracted from its Network Management System (NMS), bringing more reliability to our results. This work shows that using ML in telecommunication broadband networks is viable and can change how telecom operators manage and improve customer experience. We show that an intelligent model could do machine learning in telecom networks and make decisions without human intervention. Three automatic cognitive models were tested as experimental proof of concept with an average accuracy above 96%.

1 Introduction

The concept of network Self Healing (SH), or self-correcting networks, is becoming a major objective of telecommunications operators to achieve the expected customer's level of service. According to [1], a system with a self-healing property is expected to be able to monitor and recognize anomalies, locate the source of a failure, respond to changing conditions, and execute mechanisms to bring the system back to the *normal* operational state.

L. Barolli (Ed.): AINA 2024, LNDECT 201, pp. 163–174, 2024.
https://doi.org/10.1007/978-3-031-57870-0_15

With the increasing deployment of new technologies of Automation, Artificial Intelligence, Analytics and related Application Programming Interfaces (API), possibilities are created for SH applications in various types of networks, such as fiber in the loop access networks (Fiber to the x (FTTx), where x generalizes for several configurations of fiber deployment), metropolitan networks (metro rings), intercity backbone networks, mobile networks (Fourth Generation Cellular Mobile Network (4G) and Fifth Generation Cellular Mobile Network (5G)) and data center enterprise networks. These types of networks have one need in common: assuring high service levels. Despite the particularity of each one, all of them require specific actions to identify failures/anomalies, analyze these failures, and propose their correction. Most of these activities are carried out manually by telecommunications operators' analysts. This may lead to more time to resolve an issue, increasing the Mean Time To Repair (MTTR), impacting the Service Level Agreement (SLA), and increasing service user complaints.

In this work, we present a framework that includes self-diagnosis, analysis, and application of intelligence, and, based on automation, it can deal with network failures to reduce the downtime of telecommunications networks. The work relies on information generated by network elements. It dynamically treats these volumes of data, resorting to artificial intelligence to learn and understand the behavior of data flows progressively and take self-correcting actions if required.

The work contributes to leveraging customer experience in fixed broadband networks using Machine Learning (ML) to support SH using data available in systems that support telecom industry operations and are present in every operator's infrastructure. In addition, our work contributes to the literature by providing a study that uses a real dataset, publicly available [2], and provides three different ML models that provide high accuracy and contribute to the research in this area of study.

The remainder of this paper is organized as follows: the related works are discussed and compared in Sect. 2; the proposed framework for cognitive self-healing in telco networks and the experimental methodology is described in Sect. 3; and the obtained results and insights are included in Sect. 4. Finally, the study's concluding remarks and suggestions for future work are presented in Sect. 5.

2 Related Work

In this section, we will discuss related work and observe research opportunities for evolution.

The work of [3], uses ML with supervised learning techniques, like Decision Tree (DT) and Support Vector Machines (SVM) in a wireless sensors network, using a 30 sensors data set. They generate Link quality estimation to improve fault prediction. However, this work did not apply the concepts of SH nor real data from telecommunication operators.

The paper [4] works with unsupervised learning in a broadband internet service provider to improve the fault detection process using historical logs from a network operator. Meanwhile, they do not propose a near real-time solution for this process.

In [5], actors used DT algorithms to identify faults in enterprise networks. They used a generic data set and classified situations as normal or anomalous, associating this with the correlated hardware and software components. Despite this, they did not apply SH concepts nor classify the situations in a near real-time way.

As regards network management, the authors of [6], adopting the concept of cognitive computing, propose an intelligent architecture of autonomous network based on Network Functions Virtualization (NFV), capable of achieving Quality of Service (QoS) objectives and operational efficiency. This architecture, called CogNet [6], uses ML algorithms to dynamically adapt resources to the immediate requirements of NFVs, minimizing application performance degradations. As in CogNet, our work will use ML for data analysis and subsequent suggestions of the best action to be worked on. However, CogNet only applies self-management to generic NFV structures, not real broadband network elements.

According to [7], an important point to be considered in carrying out a SH process is to work on the data collection part. The work identifies that telecommunication networks have four striking characteristics that can be difficult to handle with automation: volume (large quantities), speed (high frequency of collection), variety, and value (information hidden in the data). In [7], a group of databases is used in a single database for querying, which helps in intelligent agent learning. Despite studying the characteristics of telecommunications network data, [7] does not use real data from operators to validate its model.

In [8], the authors tested seven SH methods for 4G. They realized that unsupervised classification methods such as ADABOOST, Fuzzy logic-based with *big data* had behaved well, bringing good diagnoses to the root causes of failures in a mobile network. Conversely, in the present work, supervised learning methods are adopted, attending to the characteristics of data available for the training phase (see details in Sect. 3).

The work [9] uses real data from a telecom operator to do traffic prediction analysis of a broadband concentrator - Broadband Network Gateway (BNG). This work classifies and detects inconsistent data from a field failure report of a Gigabit Passive Optical Network (GPON) broadband access network. Our approach also uses real data from a telecom operator but applies SH associated with ML to the operator's network instead.

In [10], Convolutional Neural Networks (CNN)s are applied to fault diagnosis in telecommunications networks. The study compares solutions using the Naive Bayes algorithm, CNNs, and the Random Forest (RF) algorithm. Their results show that CNNs perform better than the other methods studied. However, despite working with network incidents in a dataset [2] adapted between real and interpolated data, this work did not apply the concept of SH.

Table 1 compares the related works discussed above, identifying each approach's characteristics and positioning the present work.

The Self Healing column classifies if the work addresses the concept of SH. The Machine Learning column classifies if the work applies the concept of ML. The next column shows if the work considers SH applied to broadband networks. The following column shows if the work uses real network telecommunication operator's data in their experiment. The last three columns identify if the papers use a real-time process to collect data or make an action in the network and also identify the network type and the SH domain studied.

Table 1. Related Work Comparison

Work	Self-Healing	Machine Learning	SH in Broadband Networks	Network Operator Real data	Real Time	Network Type	SH Domain
Wang [3]	No	Yes	No	No	No	Wireless	Fault Prediction
Hashmi [4]	Yes	Yes	Yes	Yes	No	Broadband	Fault Detection
Kiciman [5]	No	Yes	No	No	No	Enterprise	Fault Localization
Xu [6]	Yes	Yes	No	No	Yes	Cellular	Resource Mgmt
Omar [7]	Yes	Yes	No	No	No	Cellular	Fault Det., Aut. SH
Rahmani [8]	Yes	Yes	No	No	No	Cellular	Fault Diagnosis
Silva [9]	No	Yes	No	Yes	No	Broadband	Traf. Pred., Anom. Detec.
Bothe [10]	Yes	Yes	No	Yes	Yes	Cellular	Fault Diagnosis
Present Work	Yes	Yes	Yes	Yes	Yes	Broadband	Fault Detect., Aut. SH

To improve the analysis, Table 2 shows the ML techniques and some features used in each approach.

Table 2. Related Work Features and ML Techniques

Work	ML Technique	Features used
Wang [3]	Sup: DT, SVM	Signal strength, Channel load
Hashmi [4]	Unsup:: k-Means, FCM·SOM	Fault time and cause, Region, MTTR
Kiciman [5]	Sup: DT	Paths classified as Normal or Anomalous
Xu [6]	Unsup: ANN	QoS, Energy cons., VNF Load
Omar [7]	Sup: DT	Max. data rate, #users per BS and SNR
Rahmani [8]	Unsup: SVM,Adaboost, Fuzzy	Cell Power, Avg speed, Signal Quality,
Silva [9]	Unsup: SOM Sup: RNN	OLT RX Power, ONT RX Power, ONU dist.
Bothe [10]	Sup: RF, CNN, Neur.	Cell Outage, TxPower, Antenna Tilt
Present Work	Sup: ANN, DT, GradBoost	Jitter, Latency, PL, WiFi Quality

3 A Framework for Cognitive Self-Healing in Real Broadband Networks

The present work specifies a framework for implementing Cognitive Self-Healing in GPON networks (see Fig. 1). This framework is intended to be general enough for any telecommunications operator to implement it, bringing the benefits already mentioned in this paper's introduction, such as improved service and user satisfaction.

The first step was studying the problems and needs of a telecommunications operator to identify the parameters that must be monitored to ensure the best possible experience for the operator's users in using the broadband service. To this end, several interviews were conducted with the operator's Network Operations Center (NOC) and marketing area employees.

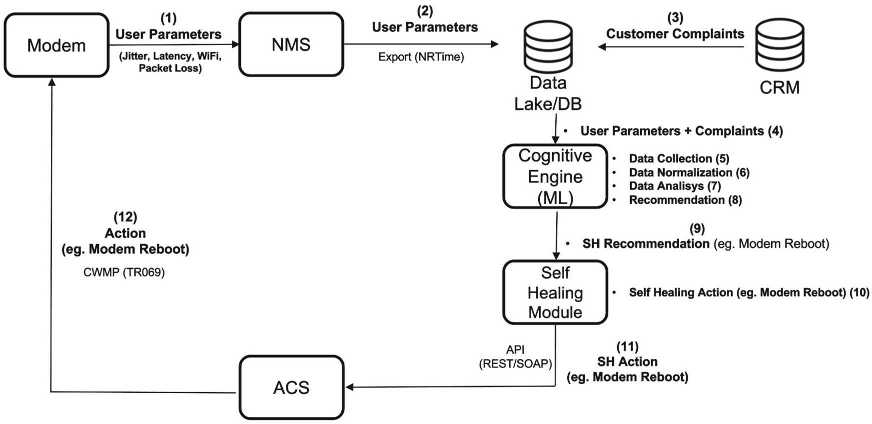

Fig. 1. Framework for Cognitive Self-Healing in Broadband Networks

One of the frequent fixed broadband internet users' pains is the low internet speed experience. The user usually does not know if problems are in the operator network, whether it is a Wireless Fidelity (Wi-Fi) router, or a specific modem issue. According to the visited operator, Modem and Wi-Fi represent together 42% of the total user complaints.

After analyzing information of the operator's Customer Relationship Management (CRM), it was noticed, by cross-referencing customer complaint information with the parameters of their modem (see Table 3), that there was a pattern indicating that a customer complaint occurred when a combination of parameters was outside the acceptable limits for a good user experience.

In this way, CRM data is collected and filtered to determine whether a customer has complained and combined with the real network parameters from the Network Management System (NMS). Now, we could see that when the user

Table 3. Parameters Definition

Parameter	Definition
Latency	Amount of time it takes for data to get from one network point to another
Jitter	Variation in latency
Packet Loss	The percentage of packets lost during transmission over a network
2.4 GHz Channel Quality	Quality of a 2.4 GHz channel, rated on a scale of 1 to 5
5GHz Channel Quality	Quality of a 5 GHz channel, rated on a scale of 1 to 5
N distant devices	Number of devices far from the modem (more than 10 m)

complained, its modem parameters were out of the acceptable limits for a good Quality of Experience (QoE).

The results are stored in a new database Data Lake (DTL) that will guide the intelligent software in the learning and automatic decision-making processes without human intervention. For this framework, we assume the NMS can export data to the DTL in a Near Real Time (NRT) model.

The framework includes implementing an ML model trained with normalized data from the DTL using supervised learning for data analysis. Three ML methods were used to evaluate which would be more accurate to identify whether the customer would complain. Once the model is trained, the model can be used to analyze the user parameters and predict whether the user will complain. Once it is identified that the client would complain, SH actions should take place. In practice, an SH module is proposed, capable of calling the API of an Auto-Configuration Server (ACS) platform that supports the CPE WAN Management Protocol (CWMP) in its Technical Report 069 (TR069) specification, defined in [11]. The ACS platform is very common in operators. It is used today by operators' customer care teams to reactively handle customer complaints with commands such as modem reboot and Wi-Fi channel change. Also, according [11], the TR069 specification is supported by the majority of the broadband modems in the world.

In the framework diagram (Fig. 1), the Modem module represents the user broadband equipment. The modem sends the user parameters (*Message #1*) to the NMS module, which collects the data from the modem. It is important to know that the network vendor implements the Modem and NMS interface in a proprietary IP protocol. However, this is not a problem for our framework because managing the modems with an NMS is mandatory to operate the network. To this framework, we assume that the NMS can export the user parameters (*Message #2*) to the Data Lake/DB module at a near real-time frequency.

The Data Lake/DB module receives data from the NMS and CRM modules. The CRM module exports the user complaints (*Message #3*) to the Data Lake/BD module being this data is important to train a core module, the Cognitive Engine (ML). The Cognitive Engine (ML) module is an intelligent component that analyses the data received from the Data Lake/BD module (*Message #4*), trains and tests a Machine Learning algorithm to understand where a specific user parameters combination could transform into a user complaint. The Cognitive Engine (ML) module, which will be further detailed in the next subsection, is expected to output SH recommendations (*Message #9*) to the Self-Healing module.

The Self Healing module is responsible for receiving the recommended action from the Cognitive Engine, analyzing the messages, identifying which type of action needs to be taken (*Message #10*), and calling the specific platform that will apply the SH action. Upon receiving an API call (normally in the HyperText Transfer Protocol (HTTP)/Representational State Transfer (REST)/Simple Object Access Protocol (SOAP)) from the SH module (*Message #11*), the ACS module triggers an action to the Modem module (*Message #12*), closing the loop of SH in the broadband network.

3.1 Cognitive Engine

This work focused on implementing the Cognitive Engine Module of the SH framework. Figure 2 illustrates the implemented components of the framework, highlighting the Cognitive Engine and the three types of algorithms we implemented.

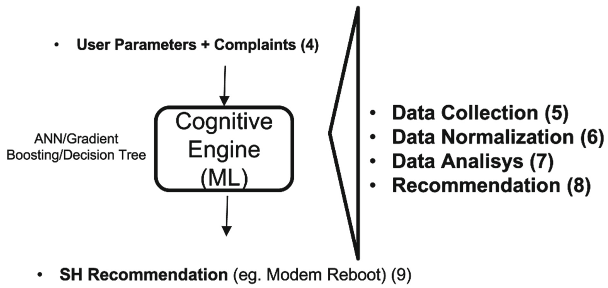

Fig. 2. Implemented Cognitive Engine Module

We implemented a ML model that analyzes the parameters of the user's broadband modem and predicts that the user would complain. From there, it would be enough to make an API call to complete the SH cycle by doing, for instance, a preventive reboot of the user's modem.

The Cognitive Engine module must take some important actions. First, the module must receive data from the DTL (*Message #5*) and once all the data are received, we pass to the data normalization step (*Message #6*), where it is important to adapt the data set replacing null values by zero, adapt some columns with fewer lines with zero and concatenates the values.

Three types of models were implemented (in Python), and their accuracy was evaluated, namely: DT, Artificial Neural Network (ANN), type (Multi-Layer Perceptron (MLP)), and Gradient Boosting. These models were selected because they represent supervised learning models, as we previously have the expected output (customer complaints or not). We choose one algorithm of higher complexity (ANN), one of low complexity (DT) and one of intermediate complexity (Gradient Boosting), to check if all of them could be applied to our framework.

For these models, the following modem user parameters were used as input for training each model: Latency, Jitter, Packet Loss, 2.4 GHz Channel Quality, 5 GHz Channel Quality, and number of distant devices. These parameters were described in Table 3.

The user parameters defined in Table 3 were crossed with the CRM database to find possible patterns. If some specific parameter configuration happens, the user normally complains to the contact center. With this rule in mind, training the ML models is the next step. If the models exhibit good accuracy in predicting a user complaint (*Message #7*), our model works, and Self-Healing recommendation (*Message #8*) can take place.

Now, we can describe each implemented algorithm.

3.1.1 Decision Tree

A DT is a non-parametric supervised learning method for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the features of the data. A tree can be seen as a constant approximation by parts. The deeper the tree, the more complex the decision rules and the finer the model. Our decision tree model analyzes the parameters of the DTL and identifies a potential problem.

Figure 3 illustrates the DT algorithm implemented inside our framework. The algorithm was created based on the class *DecisionTreeClassifier* from *sklearn* library [12], using a random state parameter to start this method. In this case, we use 70% of the data for training and 30% to test.

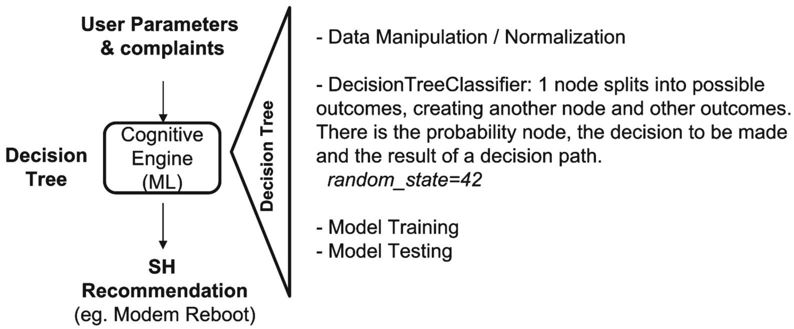

Fig. 3. Implemented Decision Tree

As output of the DT method, a data frame is generated as an Excel file. This data frame contains the user parameters, the expected output (user complains or not), and the real output. After this, the expected and real output can be compared, and the method's accuracy can be measured.

3.1.2 Fully Connected Neural Network (Multi-Layer Perceptron)

A MLP is an ANN like the perceptron type (classifier), but with more than one layer of neurons in direct feed. This type of network comprises layers of neurons connected by weighted synapses. Learning in this type of network is usually done through the backpropagation algorithm.

In our work, an ANN algorithm was created using the *Sequential* and *Dense* classes from Keras API [13], with the following parameters: 1 input layer (100 neurons, activation function = relu); 2 dense layers (100 neurons, activation function = relu); 1 output layer (1 neuron, activation function = sigmoid); Loss function='binary cross entropy'; optimizer='adam'; metrics=['accuracy']. Figure 4 illustrates the ANN algorithm implemented inside the framework. In this case we use 70% of the data to training and 30% to test.

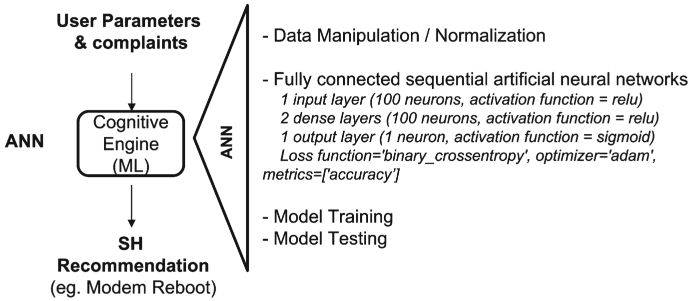

Fig. 4. Implemented ANN

A data frame is generated as an Excel file as an output of the ANN method. This data frame contains the user parameters, the expected output (user complains or not), and the real output. After this, the expected and real outputs are compared, and the method's accuracy is measured.

3.1.3 Gradient Boosting

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of a set of weak prediction models, usually decision trees. It constructs the model in steps and generalizes them, allowing the optimization of an arbitrary differentiable loss function. In short, the previous learning data analysis errors are gradually reduced.

In our work, a Gradient Boosting algorithm was created using the class *GradientBoostingClassifier* from the *sklearn* library [12], using a random state parameter to start this method. Figure 5 illustrates the Gradient Boosting algorithm implemented inside the framework. In this case, we use 70% of the data for training and 30% to test.

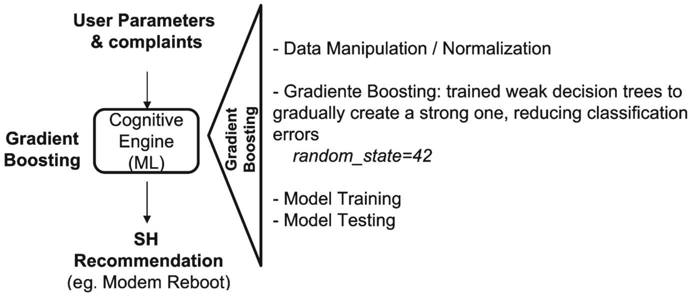

Fig. 5. Implemented Gradient Boosting

As before, the Gradient Boosting method generates a data frame as output (Excel file). This data frame contains the user parameters, the expected output (user complains or not), and the real output. After this, the expected and real outputs are compared, and the method's accuracy is measured.

4 Results and Discussion

In the tests carried out, real data of users from two cities served by the operator was collected: one dataset focusing on the retail market segment (residential broadband) and the other on the business market segment (micro and small business). These datasets were unified, anonymized and provided information about the modem parameters (Jitter, Latency, Packet Loss, and Quality of the Wi-Fi Channel used). The expected output regarding customer complaints in the period (customer complain = 1 or don't complain = 0) was added to this data, completing the final dataset [2]. In this way, it was expected that the models would predict, by analyzing the modem data, whether the user would complain.

The results of the algorithms are presented in Table 4, where "Precision" is the most important parameter for comparing the models.

Table 4. Results

Algorithm	#Layers	#Epochs	#Neurons	#Coded Lines	#Training Time (s)	Precision (%)
Decision Tree	NA	NA	NA	162	120	97,87
Artificial Neural Network	4 layers	1000	100	183	600	97,55
Gradient Boosting	NA	NA	NA	161	300	96,90

As shown in the table, all the algorithms led to a precision higher than 96%, which is acceptable in this context. This precision validates our assumption that a cognitive model could be important in an operator's customer care process.

The experiments also showed that the ANN is a flexible and high-potential tool to solve classification problems. In this model, the training time (in this experiment, the slower) is a point to observe and will depend on the problem to solve. As we used 2043 lines in our experiment, it was not a concern. However, it can be challenging when the problem needs larger data volumes to train the model.

As with any ML model, some post-training precautions are important to maintain good performance when the algorithm is in production once data sources on the operators can change (new data, dynamic environment, etc.). The model revaluation is very important to ensure process performance. The periodic model retraining with new data is important in this phase to ensure that the accuracy will continue.

The proposed framework can effectively change how operators work, allowing them to take preemptive actions based on ML. In this way, the system can automatically decide based on user parameters without human intervention. Once this framework is implemented in telecommunications operators, user satisfaction can be improved because SH actions can be carried out timely (e.g., modem reboot), avoiding user complaints.

It is important to note that this framework can be implemented in telecommunications operators with proprietary, open solutions or in hybrid scenarios. This flexibility strengthens the present framework proposal.

5 Conclusions

This paper presented a new framework for cognitive self-healing in real broadband networks to reduce customer complaints by anticipating corrective measures based on ML. The study relies on real broadband user parameters collected from a network operator in Brazil, discussed with its engineering and operating teams to learn how to extract value from CRM and Modem data. Once data was available in a Data Lake/DB, a Cognitive Engine (ML) was implemented to predict whether a user would complain with optimal accuracy. The results proved that the three cognitive data analysis methods yielded an optimal accuracy above 96%. With this proven, a Self-Healing script was applied, when applicable, using the CWMP platform to interact with the customer device to avoid a likely complaint.

In future work, new ML algorithms, such as CNN and Generative Artificial Intelligence (GenAI), may be considered an alternative to the implemented models, focusing on increasing the accuracy achieved. Another suggestion is to enhance the final SH script to evaluate the performance of the complete loop (total time from diagnosis to corrective action). An additional point to explore is the application of this framework to other types of networks, such as metro network switches, backbone routers (edge and core), access aggregators (Optical Line Terminal (OLT)s), and 5G voice and data switching centers.

Acknowledgements. The authors thank Algar Telecom S/A (Algar Telecom), a Brazilian telecommunications operator, who provided the real anonymous broadband network database. The authors also thanks the National Council for Scientific and Technological Development (CNPq) under grant number 421944/2021-8 (call CNPq/MCTI/FNDCT 18/2021) and Centro Algoritmi, funded by Fundação para a Ciência e Tecnologia (FCT) within the RD Units Project Scope 2020-2023 (UIDB/00319/2020) for partially support this work.

References

1. Souza Neto, N., Oliveira, D., Gonçalves, M., Silva, F., Frosi, P.: Self-healing in the scope of software-based computer and mobile networks, pp. 325–344 (2021)
2. Vieira, E., De Oliveira Silva, F.: Real dataset from broadband customers of a Brazilian telecom operator (2024). https://zenodo.org/records/10482897
3. Wang, Y., Martonosi, M., Peh, L.-S.: Predicting link quality using supervised learning in wireless sensor networks. Mobile Comput. Commun. Rev. **11**, 71–83 (2007)
4. Hashmi, U.S., Darbandi, A., Imran, A.: Enabling proactive self-healing by data mining network failure logs. In: 2017 International Conference on Computing, Networking and Communications (ICNC), pp. 511–517 (2017)
5. Kiciman, E., Fox, A.: Detecting application-level failures in component-based internet services. IEEE Trans. Neural Netw. **16**(5), 1027–1041 (2005)
6. Xu, L., et al.: Cognet: a network management architecture featuring cognitive capabilities. In: 2016 European Conference on Networks and Communications (EuCNC), pp. 325–329 (2016)
7. Omar, T., Ketseoglou, T., Naffaa, I.: A novel self-healing model using precoding & big-data based approach for 5g networks. Perv. Mob. Comput. **73**, 101365 (2021)
8. Rahmani, J., Sadeqi, A., Ayeh Mensah, D.N.: MPRA - self-healing in LTE networks with unsupervised learning techniques (2020)
9. Silva, W.S., Silva de Morais, A., Silva, W.D.O.: Proposal for the use of neural networks for data clustering in the context of qualitative analysis of complaints information in telecommunications services. New Trends Qual. Res. **4**, 499–506 (2020). https://publi.ludomedia.org/index.php/ntqr/article/view/66
10. Bothe, S., Masood, U., Farooq, H., Imran, A.: Neuromorphic AI empowered root cause analysis of faults in emerging networks. CoRR arxiv:2005.01472 (2020)
11. BroadBandForum. Technical report - tr-069 cpe wan management protocol (2020). https://www.broadband-forum.org/pdfs/tr-069-1-6-1.pdf
12. Pedregosa, F., et al.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
13. Chollet, F., et al.: Keras (2015). https://github.com/fchollet/keras