# On Using Explainable Artificial Intelligence for Failure Identification in Microwave Networks

Omran Ayoub, Francesco Musumeci, Fatima Ezzeddine, Claudio Passera, and Massimo Tornatore

***Abstract*—Artificial Intelligence (AI) has demonstrated super-human capabilities in solving a significant number of tasks, leading to widespread industrial adoption. For in-field network-management application, AI-based solutions, however, have often risen skepticism among practitioners as their internal reasoning is not exposed and their decisions cannot be easily explained, preventing humans from trusting and even understanding them. To address this shortcoming, a new area in AI, called Explainable AI (XAI), is attracting the attention of both academic and industrial researchers. XAI is concerned with explaining and interpreting the internal reasoning and the outcome of AI-based models to achieve more trustable and practical deployment. In this work, we investigate the application of XAI for automated failure-cause identification in microwave networks. We first show how existing supervised ML algorithms can be used to solve the problem of failure-cause identification, achieving an accuracy around 94%. Then, we explore the application of well-known XAI frameworks (such as SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME)) to address important practical questions rising during the actual deployment of automated failure-cause identification in microwave networks. These questions, if answered, allow for a deeper understanding of the behavior of the ML algorithm adopted. Precisely, we exploit XAI to understand the main reasons leading to ML algorithm's decisions and to explain why the model makes identification errors over specific instances.**

## I. Introduction

Following the increasing availability of monitoring data and the recent advances in computing platforms, Artificial Intelligence (AI) and Machine Learning (ML) are becoming key tools for network operators to automate network management and address, among others, challenging failure management problems as failure detection, failure-cause identification, failure prediction and localization. In this first wave of deployment of ML-based failure management solution, operators have often relied on complex ML models, used as "black boxes", i.e., as models that do not expose their internal reasoning. This represents a main obstacle for successful field deployment of ML models, as their output are not easily interpretable and explainable and network operators may not gain full trust of such "black box" model decision.

To address the trust issue commented above, efforts are being made in the field of interpretable ML and eXplainable Artificial Intelligence (XAI) to explain decisions or predictions of a ML model with the aim of transforming the black box into a "transparent" (or "glass") box. By applying XAI frameworks, humans can have an improved understanding of a model's behavior and know when to trust its decisions, as XAI frameworks allow to shed light on how model features are used as driving factors towards decisions. In this paper, we describe our first step in the application of XAI frameworks, for the specific problem of automated failure-cause identification in microwave networks.

Automated failure-cause identification allows operators to reduce service unavailability by repairing failures much more rapidly than when relying on time-consuming manual analysis of failure logs. In our previous work [1], we modeled the problem of failure-cause identification in microwave networks as a classification problem and proposed supervised ML algorithms that were able to discriminate with high accuracy among different failure-causes. Based on the ML model decision, network operators can timely take the most appropriate countermeasure to repair a failure, which may, e.g., consist of an on-site intervention vs. a remote equipment configuration. Considering that, often, microwave equipment is situated in areas not easily-reachable (e.g., on top of a hill, a location that might even require a helicopter to move the repair crew), we emphasize here how initiating a repair action based on a wrong failure-cause identification can lead to significant and unnecessary costs for the operator. For instance, consider the case of a remotely-repairable failure that is wrongly classified as a failure requiring on-site intervention; in this case the operator is incurring unnecessarily in the much higher cost of on-site intervention. Similarly, the opposite mis-classification would lead to excessive delays for failure reparation. Therefore, ML models are not only required to be accurate, but, for possible-expensive decisions, they must allow the operator to scrutinize the confidence of the decision before trusting the decision. In other words, it is decisive to know if and when to trust model's decisions and when not, and to understand the driving factors of the decisions.

In this paper, in order to understand the main driving factors leading to ML algorithms' decisions for failure-cause identification, we first show three ML algorithms, namely, Random Forest (RF), Artificial Neural Networks (ANN) and eXtreme Gradient Boosting (XGB), for automated failure-cause identification and compare their performance in terms of classification accuracy. Then, we apply two XAI frameworks, namely, SHAP [4] and LIME [3], for generating "global" explanations of models' behavior and for explaining reasons behind wrongly classified instances, respectively. Such explanations go beyond, for instance, a pseudo-code describing the algorithm, as they correlate model features to decisions, al-

Omran Ayoub, Francesco Musumeci and Massimo Tornatore are with Politecnico di Milano, Italy. Claudio Passera is with SIAE Microelettronica, Italy. Fatima Ezzeddine is with the Lebanese University, Lebanon.

Corresponding author email: omran.ayoub@polimi.it.

lowing practitioners to gain insights on driving factors behind decisions. We believe this description of our application of XAI in the context of microwave networks can be of help to stimulate further investigation on the application of XAI in network failure management.

We can summarize the contribution of this paper as follows: 1) we show that the different ML algorithms rely on different sets of features to identify failure causes; 2) we evaluate the relevance of the features for the different failure causes and show how the obtained results can be used to validate the classifier decisions and 3) by explaining wrongly classified instances, we show how to extract insights on the reasons underlying a wrong failure classification.

The paper is organized as follows. Sec. II provides background on XAI, in general, and on the two XAI frameworks applied in our work, in particular. Sec. III discusses preliminary concepts for microwave networks and their most common types of failures. In Sec. IV we qualitatively state the failure-identification problem, and we elaborate the specific research questions addressed in our work. Sec. V describes data and supervised ML models used in this work. Sec. VI presents numerical results, and shows how we address the research questions introduced in Section IV.

## II. Explainable Artificial Intelligence

In this section we first provide a brief overview on XAI and then we describe the XAI frameworks applied in our work. For further reading, we refer the reader to comprehensive surveys on interpretable ML and XAI [5]–[7], [12].

### A. Overview

When examining literature on XAI, we notice that *explainability* and *interpretability* are two terms that are used interchangeably by researchers, although some works have put effort to identify differences between them [8]–[10]. For either terms, no rigorous mathematical definitions that allow to measure them exist. Instead, measuring interpretability of an AI model has been defined qualitatively by researchers as the degree to which a human can understand the cause of a decision of that model [11]. In contrast, explainability is associated with humans' understanding how the internal logic of the model can be explained.

In our work, we are interested in explaining pre-trained "black-box" models with the aim of understanding the internal logic of a ML model. This process is performed by using existing XAI frameworks in a *post-hoc* manner, i.e., after the model has taken its decision (see Figure 1). XAI frameworks can be either model-specific, i.e., their application is restricted to specific ML models, or model-agnostic, i.e., they can be applied to any ML model. Explanations are also divided into two classes, global and local. A global explanation explains the whole model's behavior, while a local explanation provides explanation to a specific observation. Depending on the aim, both types of explanations can be necessary to explain the behavior of a model. In our work, we apply two model-agnostic methods, SHAP and LIME, in a post-hoc manner, with the aim of explaining the behavior of ML models.

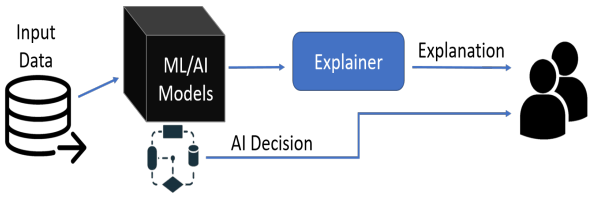


Fig. 1: Post-hoc explanation.

### B. Applied XAI Frameworks

We now provide an overview on the two model-agnostic XAI methods used in this paper, namely, SHAP and LIME [3]. Specifically, we first use SHAP to evaluate the contribution of the various features to model's decision, in order to identify which features are most relevant to model's decision to each of the classes of failure. Then, we use LIME to generate local explanations of selected wrongly-classified observations to understand why the model misclassifies them.

*1) Local Interpretable Model-agnostic Explanations (LIME):* ML models are widely applied to solve particular tasks such as classification and regression. In most cases, models with high predictive capacity, such as ANNs, are preferred. However, such models are not easily interpreted by humans. To increase interpretability of these complex models, other simpler models, referred to as surrogate models, can be used, which are constrained by design to be interpretable. The role of any interpretable surrogate model is to imitate the behavior of a more complex ML model while providing a description of its own behavior, consequently explaining the behavior of the complex model. For instance, to interpret an ANN, a logistic regression model, for example, can be used as a surrogate model, to explain decisions boundaries and provide a description of model's behavior. In this case, the decision boundary of the non-linear model will coincide with that of the linear model in a local space in proximity of the instance whose prediction is explained, and therefore, the behavior of the linear model can be used to explain that of the non-linear model.

Two kinds of surrogate models exist, global and local. In a part of our work, we rely on local surrogate models and specifically on LIME. LIME is a model-agnostic technique (explains any machine learning model, and hence the name *explainer*) used to generate explanations of local decisions. LIME was proposed in [3], and it explains single predictions relying on easily interpretable models such as linear regression or decision trees. A LIME explanation is generated as follows.

- Select an instance $x$ of the dataset $X$ and its predicted target value to be explained
- Perturb dataset $X$ (i.e., change features' values of data points in $X$) to generate a new data set $Z$ of a larger size with respect to original dataset $X$. Perturbation of original data is performed to generate new observations similar to original ones to be additionally considered

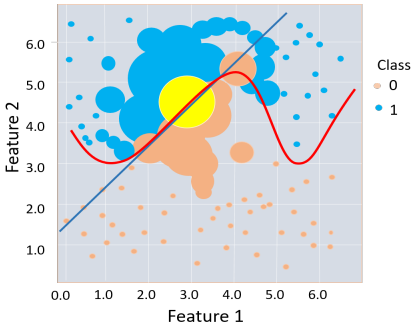



Fig. 2: Graphical representation of LIME algorithm.

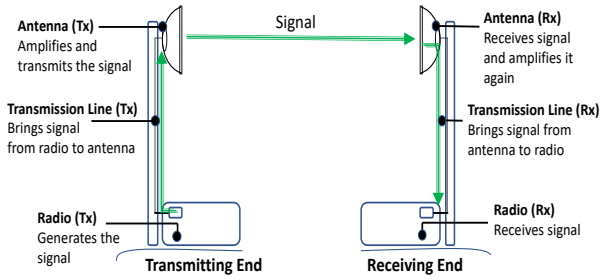



Fig. 3: Basic components of a microwave link.

when generating explanations, with the aim of better describing the behavior of the black-box model.

- Using original black box model, predict target values for all instances in $Z$.
- Weight elements in $Z$ with respect to the proximity (also referred to as *neighborhood*) to instance $x$. Note that the neighborhood is determined by giving data points weights according to their proximity to the instance to be explained.
- Train a surrogate explainable model $g$ on $Z$ and respective predictions.
- Return an explanation for explainable model $g$ for instance $x$.

Figure 2 shows a visualization of main components in the above procedure for a binary classification problem. Instance $x$ to be explained is in yellow, data points in proximity to instance $x$ are given higher weights (represented by larger points). Decision boundary of the original black box model is represented by a black curve while the decision boundary of the surrogate model is represented by a red line. The decision boundaries of the two models coincide locally, i.e., in proximity to observation to be explained, however they are significantly distant globally.

*2) SHAP: SHapley Additive exPlanations:* In 1952, economist Lloyd Shapley proposed a method from coalitional game theory to assign fair payouts to players based on their contribution to the total payout. In this method, players cooperate in a coalition and receive a certain profit from this cooperation. Then, a value, referred to as *Shapley value*, is computed as the average marginal contribution of a player across all coalitions. In the context of explainability, SHAP estimates the Shapley value (i.e., the marginal contribution or importance value) of each input feature of an instance to the prediction by iterating through all permutations of the input features, where each feature is a player in a game and the prediction is the payout to be distributed. For each classified instance, SHAP calculates the contribution of each feature to the classification value, i.e., to model's decision. Following this method, SHAP explains predictions of an observation by computing the contributions of each feature to model's decision.

As we will see later, examining the Shapley values of features (i.e., the contributions of features) to model's output allows to understand which features contribute most to deciding in favour of a particular class of failures. In other words, examining explanations of SHAP may allow operators to confirm the correct behavior of the model (by confirming that model's decision for a specific class of failure are driven by features relevant to that class), and even to discover whether a feature is relevant to a particular class of failure.

## III. Microwave Link Failures

This section describes the main building blocks of microwave links and details the various failure causes in microwave networks and discusses the typical countermeasures adopted to contrast them.

### A. Microwave Link

Figure 3 shows the basic structure of a microwave link, which can function (transmit and receive) in a bidirectional manner, from site A to site B and from site B to site A, given that link transmitting and receiving equipment is present at both sites. The link consists of three main elements, i.e., 1) the microwave radio, 2) the transmission line and 3) the antenna. The microwave radio can be placed at different locations, i.e., either inside a building (full-indoor), in proximity of the antenna (full-outdoor), or by adopting a hybrid solution, where the electronic devices are distributed between an outdoor unit (ODU) and indoor unit (IDU). At the transmitter side, the microwave radio is responsible of generating the analogue signal, while, at the receiver side, it demodulates the signal. The transmission line connects the microwave radio to the antenna. The physical medium of the transmission line is typically a coaxial cable. The antenna is usually parabolic-shaped and is characterized by its, gain, size and directivity function, i.e., the capability of concentrating the transmitted/received power to/from specific directions.

### B. Link Performance and Unavailability

The performance of a microwave link is monitored by evaluating the number of errored bits (referred to as errors) in a certain time span. The number of errored bits defines three main metrics: 1) *errored block* (EB): a block (i.e., group of consecutive bits) in which one or more bits are in error. 2) *errored Second* (ES): a one-second period with one

or more errored blocks. 3) *severely errored Second* (SES): a one-second period which contains 30% errored blocks. Specifically, according to ITU-T Recommendations G.826 and G.828 [2], when the number of consecutive SES exceeds ten in one or in both directions of the microwave link, the link is considered to be in a state of unavailability. The unavailability is then measured in terms of *UnAvailability Seconds* (UAS), which represent the amount of time (expressed in seconds) when the number of errors exceeds a certain threshold. The link is considered to be again available if, after the block of consecutive SES, no SES are present for at least ten consecutive one-second periods in both directions. Note that a microwave link can experience UAS for a period of time and then go back to normal functioning. This is because the microwave link can be frequently affected by external factors, such as the atmosphere, which may affect the functioning of the link temporarily. We discuss types of failures affecting microwave links in detail in next subsection.

### C. Categories of Failure

In this work, we consider six different failure causes, that we identify with six different classes $C_0$-to-$C_5$. Among them, the first five are propagation-related failure causes, i.e., driven by atmospheric factors or presence of temporary obstacles, while the last one consists of hardware failure, i.e., caused by equipment malfunctioning due to, e.g., aging, high temperature, etc.. In the following, we detail each failure type, and highlight the typical countermeasures adopted in each case.

1) **Deep Fading** ($C_0$) consists of a strong increase of channel attenuation causing a severe drop in signal-to-noise ratio, and can be due to many factors such as, e.g., seasonality, geographical position or radio frequency in use. It can be caused by the presence of new obstacles (e.g., growth of vegetation) or adverse meteorological phenomena, such as heavy rain, snow or fog, leading to multipath and shadowing effects. To deal with deep fading, no on-site human intervention is required. Instead, it can be automatically solved by a temporary reduction of link's modulation format.

2) **Extra Attenuation** ($C_1$) occurs when received power is well below (e.g., 6 or more dB lower) the minimum power threshold, even considering the lowest-order modulation format in the link. Extra attenuation can be caused, e.g., by path obstruction (due to the presence of permanent obstacles), antenna misalignment, mounting/screwing issues, water infiltration into waveguide used in the transmission line or damaged antenna/coupler. To deal with extra attenuation, either remote or on-site human intervention is required.

3) **Interference** ($C_2$) occurs when a receiving antenna receives multiple bit streams due to overlap of other transmissions at its frequency, causing it to fail to distinguish the bit stream destined to it. The multiple bit streams are caused by, e.g., unexpected reflections from other links or frequency misconfigurations. Typically, interference does not change over time, and it is typically solved by turning off the interfering link or changing its carrier frequency through human intervention.

TABLE I: Distribution of data points over failure classes.

| **Failure Cause** | **# of 45-minutes windows** |
|---|---|
| $C_0$ - *Deep Fading* | 284 |
| $C_1$ - *Extra Attenuation* | 581 |
| $C_2$ - *Interference* | 49 |
| $C_3$ - *Low Margin* | 190 |
| $C_4$ - *Self-Interference* | 187 |
| $C_5$ - *Hardware Failure* | 1222 |

4) **Low Margin** ($C_3$) occurs when the link configuration parameters have not been chosen adequately, i.e., they do not correspond to the ones recommended by the manufacturer, causing UAS events to occur. To address low margin failure types, remote human intervention is required to correctly configure link's parameter.

5) **Self-Interference** ($C_4$) occurs when the link is operated in full-duplex, and the transmission line, which is shared between the two streams and connects the antenna to two radio components, creates local signal reflections and spurious signals which are propagated to the receiver radio component. Self-interference is a propagation problem that can be due to degradation of the hardware (e.g., amplifiers and/or filters) used to eliminate signal reflections, and it typically causes random UAS on the link, even when the link is working at nominal received power level and no fading event is occurring. To eliminate self-interference, on-site human intervention is required to substitute hardware components and re-configure link parameters.

6) **Hardware Failure** ($C_5$) refers to the cases of link unavailability that are not directly related to propagation problems, including failures due to equipment failure. Such failures can be either temporary or permanent, and in both cases, they require on-site human intervention to replace hardware equipment causing the failure.

## IV. Problem Statement and Research Questions

We model the problem of failure-cause identification in microwave networks as a supervised multi-class classification problem. As input, the supervised ML model takes a 45-minutes window observation on a microwave link, consisting of three 15-minutes windows in which the last window suffers from at least one UAS event. For a given link in a given 45-minutes window, a total of 35 features, describing link's design parameters and performance metrics, are used to model data points input. As output, the model provides a label corresponding to one of the 6 failure causes discussed previously. After ML classifiers have been trained in a supervised manner, we apply XAI frameworks to explain model's global behavior and local decisions with the aim of addressing the following Research Questions (RQs):

- *RQ1: Are the lists of most important features the same among the ML models? In other words, do the ML models considered have the same list of most important features?*
- *RQ2: Which features are most influencing model's decision for each failure class and how? Do the features*

TABLE II: Features describing a 45-minute window of the radio link. Feature names with '*' are measurement features with three different values, one for each 15-minutes slot.

| Type | Feature | Name | Description |
|---|---|---|---|
| Link Characteristics | $f_1$ | LowThr | Minimum received power tolerated on the link with any modulation format used (dBm) |
| | $f_2$ | Ptx | Nominal transmitted power when the minimum modulation format is used (dBm) |
| | $f_3$ | Thr_min | Minimum received power threshold tolerated by the link with its current modulation format (dBm) |
| | $f_4$ | RxNominal | Nominal received power at the maximum modulation format (dBm) |
| | $f_5$ | acmEngine | A flag which indicates if the Adaptive Code Modulation (ACM) is enabled on a given microwave link |
| G.828 metrics | $f_6$, $f_7$, $f_8$ | ES* | Number of one-second periods with at least one ES in the 15-minutes slot |
| | $f_9$, $f_{10}$, $f_{11}$ | SES* | Number of one-second periods with at least one SES in the 15-minutes slot |
| Power values | $f_{12}$, $f_{13}$, $f_{14}$ | txMaxA* | Maximum power transmitted from site A in in the 15-minutes slot (dBm) |
| | $f_{15}$, $f_{16}$, $f_{17}$ | txminA* | Minimum power transmitted from site A in the 15-minutes slot (dBm) |
| | $f_{18}$, $f_{19}$, $f_{20}$ | rxmaxA* | Maximum power received at site A in the 15-minutes slot (dBm) |
| | $f_{21}$, $f_{22}$, $f_{23}$ | rxminA* | Minimum power received from site A in the 15-minutes slot (dBm) |
| | $f_{24}$, $f_{25}$, $f_{26}$ | txMaxB* | Maximum power transmitted from site B in the 15-minutes slot (dBm) |
| | $f_{27}$, $f_{28}$, $f_{29}$ | txminB* | Minimum power transmitted from site B in the 15-minutes slot (dBm) |
| | $f_{30}$, $f_{31}$, $f_{32}$ | rxmaxB* | Maximum power received at site B in the 15-minutes slot (dBm) |
| | $f_{33}$, $f_{34}$, $f_{35}$ | rxminB* | Minimum power received from site B in the 15-minutes slot (dBm) |

*of the 15-minutes windows preceding the window in which failure has occurred have an influence on model's decisions for any of the failure classes?*

- *RQ3: Can we determine why the model systematically misclassifies instances of one class as instances of another particular class?*

To answer RQ1 and RQ2, we specifically exploit SHAP. Specifically, we generate SHAP summary plots to examine the list of features (and their values) that influence most model's decisions for each class of failure. To address RQ3, we use LIME to explain model's decisions for wrongly classified observations.

## V. Data Description and ML Models

### A. Data Description

The dataset used in this work is collected from more than 10 thousand point-to-point links of a real microwave network for a duration of 18 months. For each link, several performance metrics are collected for both sites at fixed time-steps of 15 minutes via a network management system of SIAE Microelettronica. As a data point, we consider 45-minute windows constituted by three consecutive slots of 15 minutes. Each 45-minutes window in our dataset is characterized by at least one UAS event in the last 15-minutes slot. This means that we consider, in addition to the window suffering from UAS, the two previous 15-minute slots. This consideration is based on the knowledge of domain experts, who affirm that a 45-minutes time span is deemed sufficient to capture temporal dynamics of failure causes in microwave links. A total of 2513 45-minutes windows have been manually labeled by domain experts with a label representing one of the failure causes described previously. The manual labelling of all data points required the effort of two domain experts for two weeks. We assume the manual labelling represents the ground truth. For each data point, we consider 35 features. The features, described in Tab. II include: 5 features ($f_1$ - $f_5$) describing design parameters, and hence, they do not depend on the 15-minutes slots considered in the 45-minutes window; 6 features ($f_6$ - $f_{11}$) representing G.828 performance measures ES and SES for the three 15-minutes slots; and 24 features ($f_{12}$ - $f_{35}$) representing the minimum/maximum received and transmitted power values for each side of the link (i.e., site A and site B) and for each of the three 15-minutes slots. All features passed to the ML models are normalized to make sure the model is less sensitive to the scale of data. To normalize the features, we calculate the mean $\bar{f}_i$ and the standard deviation $\sigma_i$ for each feature, considering all data points in the dataset $X$, and then, for each data point $j \in X$, we obtain the standardized features ($f_i^j$, $i = 1, 2, ..., 35$) as follows:

$$f_i^j \leftarrow \frac{f_i^j - \bar{f}_i}{\sigma_i} \tag{1}$$

Labels are distributed among the 6 classes as shown in Tab. I. A severe unbalance between some classes can be observed, due to the fact that some failure causes, such as, e.g., *interference*, are less frequent. This means that it is necessary to inspect the set of most influential features per class (and not globally by aggregating importance of all features among all points) to better understand the behavior of the model.

### B. Supervised ML Models

We consider three different ML algorithms, namely, Artificial Neural Network (ANN), Random Forest (RF) and Extreme Gradient Boosting (XGB) for failure-cause identification. In particular, ANN and RF were adopted in our previous work [1], where details on hyperparameter selection can be found. Similarly, also for XGB algorithm we tested different combinations of hyperparameters and used the classifier with highest classification accuracy. We vary *eta* parameter (learning rate) and *subsample* between 0.1 and 1 with a step of 0.1, and vary *max depth* between 1 and 10 with a step of 1. For *XGB*, the hyperparameters selected are *eta* = 0.3, *max depth* = 7 and *subsample* = 0.9.

## VI. Results and Discussion

In this section we first perform numerical evaluations of the supervised failure-cause identification and then discuss findings of applying XAI frameworks to our case study with the aim of addressing the RQs presented in Sec. IV.

TABLE III: Performance metrics of each of the three models considered in our study.

| Model | Accuracy | Precision | Recall | F1-Score | F1-score per class | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | $C_0$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ |
| RF | 0.93 | 0.94 | 0.93 | 0.93 | 0.85 | 0.91 | 0.88 | 0.76 | 0.97 | 0.97 |
| ANN | 0.88 | 0.84 | 0.84 | 0.82 | 0.69 | 0.90 | 0.89 | 0.73 | 0.97 | 0.94 |
| XGB | 0.93 | 0.93 | 0.93 | 0.93 | 0.85 | 0.92 | 0.88 | 0.76 | 0.98 | 0.98 |

TABLE IV: Confusion matrix and per-class F1-score obtained with XGB classifier.

| | | Predicted Label | | | | | |
|---|---|---|---|---|---|---|---|
| | | $C_0$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ |
| True Label | $C_0$ | 50 | 1 | 0 | 4 | 0 | 2 |
| | $C_1$ | 5 | 105 | 0 | 5 | 0 | 1 |
| | $C_2$ | 0 | 0 | 8 | 0 | 1 | 1 |
| | $C_3$ | 5 | 2 | 0 | 30 | 0 | 1 |
| | $C_4$ | 0 | 0 | 0 | 0 | 37 | 0 |
| | $C_5$ | 0 | 3 | 0 | 1 | 0 | 241 |
| F1-score | | 0.85 | 0.92 | 0.88 | 0.76 | 0.98 | 0.98 |

### A. Comparing Supervised ML Algorithms

The supervised ML algorithms used to perform failure-cause identification are compared in Tab. III, where different classification metrics are shown showing in particular *F1-score* per class. Results show that, in general, the three algorithms have a comparable performance with a slight advantage for XGB. Specifically, XGB has the best accuracy (93.6%) outperforming RF and ANN that have 93.04% and 88.66%, respectively. In terms of Precision, Recall and F1-Score, the XGB and RF algorithms show a similar performance (see Tab. III) outperforming the ANN which shows a performance 10% lower for all metrics. In terms of F1-Score for the various failure classes, XGB and RF show similar F1-score values ranging between 73% and 98%. ANN, on the contrary, suffers from a relatively low F1-score for class $c_0$ (69%) and class $c_3$ (73%). We also show in Tab. IV the confusion matrix of the best performing classifier (XGB) for one case when used on a test set of 20% of the dataset. We can see that in some cases the model predicts a class of failure that requires human intervention (classes $C_2$, $C_4$ and $C_5$), while in fact the true label corresponds to a class of failure that does not (classes $C_0$, $C_1$ and $C_3$). Such misclassifications can result costly if the operator takes actions accordingly. Leveraging on local explanations of misclassified points, we examine the reasons why the model misclassifies these points, with the aim of deriving guidelines that can help the operator to know when not to trust model's decision.

### B. XAI-Assisted Failure-Cause Identification

We now address the RQs formulated in Sec. IV.

*RQ1: Are the lists of most important features the same among the ML models?*

To address RQ1, we show in Tab. V the 10 most important features for each ML algorithm obtained with SHAP by considering all data points in dataset. Overall, results show that the models share 5 features among the 10 most important ones. More specifically, XGB shares 8 features among the most important 10 with RF, with slight differences in the order of features in terms of importance. For instance, the first 3 most important features are identical in both cases. This can be explained by the fact that both XGB and RF are decision-tree algorithms, hence, similar behavior is expected. As for XGB and ANN, they share 6 features among the list, however with notable differences in their order of importance (only one feature is common among the first 6 most important features). Moreover, with ANN, $Thr_{min}$ is the second most important feature while it is not present in the list of XGB. Similarly, with XGB, *rxminBN* is the most important feature, while with ANN it is the tenth. The case for RF and ANN is similar to that of XGB and ANN, confirming that similarities, in terms of most important features and their order, are more evident between decision-tree-based algorithms. A main point to highlight is the importance of link characteristic features for each of the models. With XGB, *lowThr* and *RxNominal* have high importance. With RF, two link characteristics features are present among the 12 most important features, *lowThr* and *acmEngine*. For ANN, on the contrary, three link characteristic features are among the 12 most important features, $Thr_{min}$ (second most important feature), *RxNominal* and *lowThr*. While this confirms the importance of link characteristics features, it also shows that each model considers a slightly different set of those features among the most impacting features. This type of insight can be leveraged by the network operator to a-posteriori verify if the ML models rely on the same set of features that would be used by experts on the problem at hand, thus contributing to the selection of the most suitable and trustable ML model.

*RQ2: Which features are most influencing model's decision for each failure class and how? Do features of 15-minutes windows preceding the window in which failure has occurred have an influence on model's decisions for any of the failure classes?*

While field experts know, at a global level, that performance metrics corresponding to windows prior to failure occurrence are necessary for failure-cause identification, discovering which performance metrics (features) are linked to specific failures is decisive for implementing AI-driven solutions for predictive maintenance, for instance. To address RQ2, we consider the case of XGB and we use SHAP to show a *summary plot* for each class of failure in Figure 4 (features with names ending with *-1* and *-2* correspond to the first and second window preceding that suffering from UAS). A summary plot combines feature importance with feature effects to explain model's behavior. The y-axis lists features according to their importance, and each point of the summary plot is a

TABLE V: Ordered list of 10 most important features according to SHAP for XGB, RF and ANN. Features with names ending *(N-1)* and *(N-2)* correspond to first and second window preceding the failure). Features with names in bold correspond to important features present in the lists of all three models.

| Feature Rank | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | XGB | **rxminBN** | **rxminAN** | esN | **lowthr** | **rxmaxBN-2** | rxmaxBN | RxNominal | rxmaxAN-2 | **rxmaxAN** | rxminBN-1 |
| | RF | **rxminAN** | **rxminBN** | esN | **rxmaxBN-2** | rxmaxAN-2 | rxmaxAN-1 | **rxmaxAN** | rxmaxBN-1 | **lowthr** | acmEngine |
| | ANN | **rxminAN** | Thr_min | rxminAN-2 | **rxmaxAN** | rxminAN-1 | rxmaxAN-1 | RxNominal | **rxmaxBN-2** | **lowthr** | **rxminBN** |

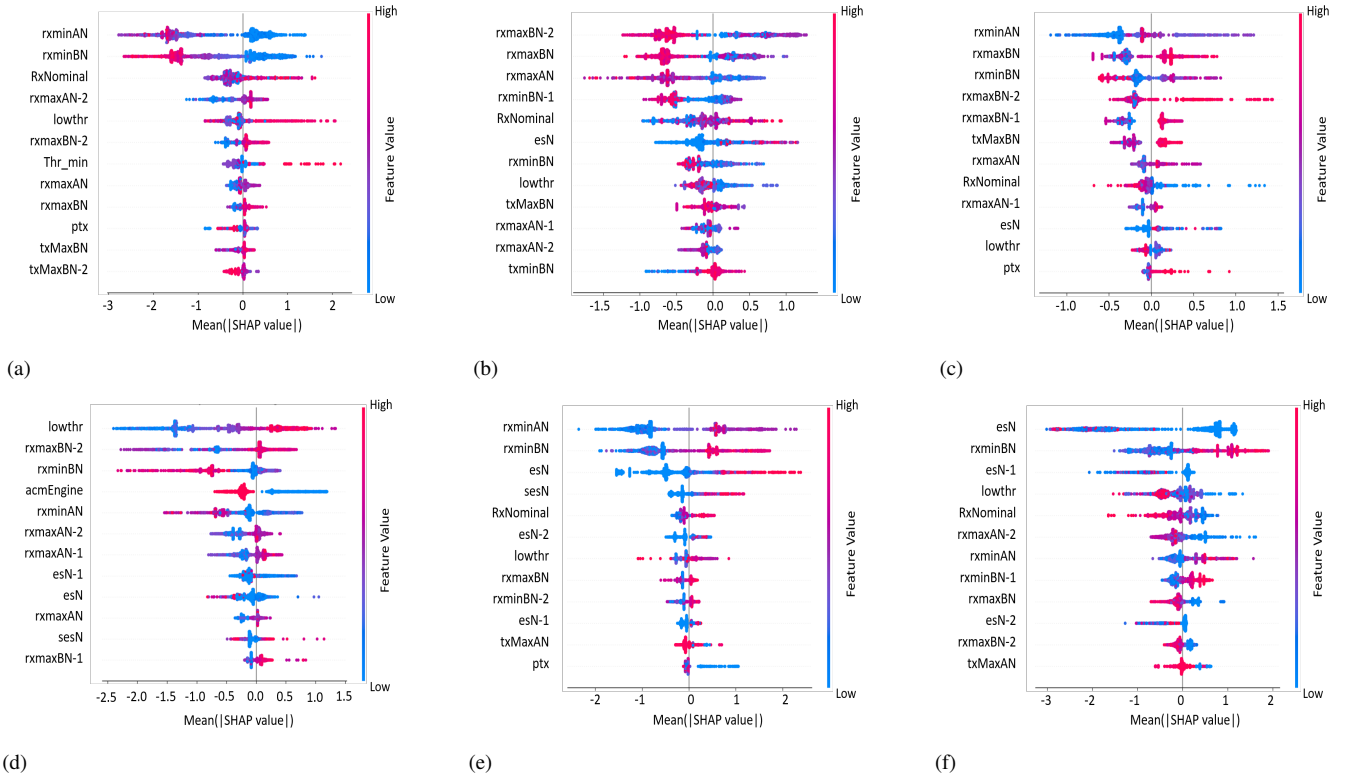
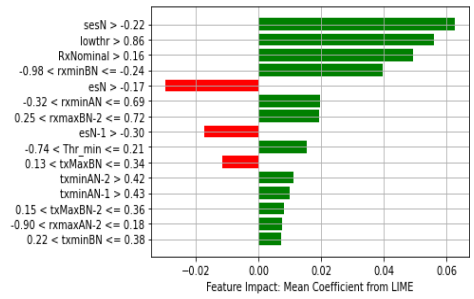



Fig. 4: Summary plot of SHAP values for ($a$) Deep Fading, ($b$) Extra Attenuation, ($c$) Interference, ($d$) Low Margin, ($e$) Self-Interference and ($f$) Hardware Failure, showing the 12 most important features per class (features with names ending *(N-1)* and *(N-2)* correspond to first and second window preceding the failure).

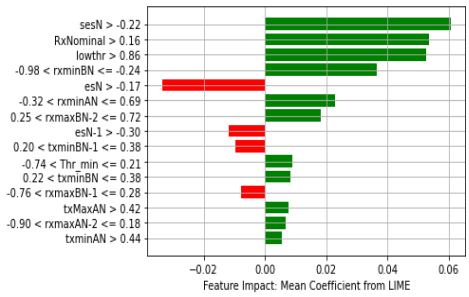
Shapley value for a given feature and a given data point (a 45-minutes window for a given link, in our case), positioned based on its Shapley value. Each point has also a color which qualitatively represents the feature value in a low-to-high scale. The overlapping points in vertical direction reflect the distribution of Shapley values for each feature. By examining summary plots of each class of failure, we understand the relationship between value of a feature and the impact on the prediction towards each class of failure. This knowledge can be leveraged to have a global understanding of model's behavior, and, when analyzed per failure class, allows to extract local insights on the behavior.

Results show that *Self-Interference*, *Deep Fading* and *Interference* rely with a limited degree on such features, as only three of them are present among the 12 most important features in each of the classes, and they have relatively low SHAP values. On the contrary, *Hardware Failure*, *Extra Attenuation* and *Low Margin*, significantly rely on such features as several of them are present among the 12 most important features. This information can be exploited by domain experts to affirm or reject model's behavior, and thus know whether to trust model's decisions. In addition, the summary plots can also be exploited to examine feature correlation with model's decision. For instance, in *Deep Fading*, low values of *rxminAN* and *rxminBN* (blue dots for first two features on y-axis) are correlated with a positive contribution while medium and high values of these features (purple and red dots) are correlated with a negative contribution (against the decision) towards *Deep Fading*. Such analysis, when performed over all features, can be used to analyze which features (and feature-values) are most influencing model's decision for each failure class, which further contributes to gain trust in the model before its application.

*RQ3: Can we determine why the model systematically misclassifies instances of one class as instances of another particular class?*




(a)




(b)

Fig. 5: LIME explanations for a wrongly classified observation

We address RQ3 by analyzing and comparing contributions of features towards 1) the true label and 2) the predicted wrong label using LIME[1]. In particular, we consider an observation with *Low Margin* as true label that was classified as *Deep Fading*, shown in Figure 5(a) and 5(b), respectively. This analysis can be leveraged by domain experts to gain insights on the problem at hand, allowing to know when the model might misclassify one class of failure to another, and therefore derive additional guidelines that would allow to avoid taking costly wrong countermeasures in future occurrences. The explanation figure is read as follows. The y-axis lists a set features in descending order of importance (influence) on the decision and the x-axis shows the LIME coefficient (feature importance). Each of the features either has a positive (green) or a negative (red) value. A positive value indicates that the feature has supported the decision towards its predicted target class while a negative value indicates otherwise. Inspecting the explanations of the sample observation when explained towards either of the failure classes, we see that the sets of most influential features (the lists of features on y-axis) are the same and that the features have the same contribution towards both classes (if a feature is contributing positively towards *Deep Fading* it contributes positively towards *Low Margin* and vice versa). Feature *sesN* and *RxNominal* are two examples of such features, which are among the most influential features and which contribute positively to both classes of failure. Through this analysis, we can explain why the model miclassifies instances of *Low Margin* with *Deep Fading*, which is returned to the fact of having a set of features with specific values that supports the decision of the model towards both classes. Relating this with SHAP summary plots in Figure 4, we see that *sesN*, although very influential in the observations explained in Figure 5, it is not among the most influential features for *Deep Fading* (Figure 4(a)). Similarly, *RxNominal* is not among the most influential features for *Low Margin* (Figure 4(d)). This means that observations similar to those in Figure 5 are not abundant among either of the classes, explaining why influential features locally are not present among the most influential features at a class level. This also shows that local explanations can be decisive to understand better the behavior of the model and to further increase trust in its decisions.

[1]An explanation of an instance can be generated towards a particular class, i.e., finding contributions of features towards a class

## VII. Conclusion

In this work, we investigate the use of eXplainable Artificial Intelligence (XAI) for automated failure-cause identification in microwave networks. After applying existing supervised ML algorithms providing 94% classification accuracy, we explore the use of SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) to address important practical questions with the aim of achieving a trustable deployment of automated failure-cause identification in microwave networks. We answer these questions showing how to achieve a deeper understanding of the behavior of the ML algorithm adopted and we further exploit XAI frameworks to extract insights of the problem.

## References

[1] Musumeci, Francesco, et al. "Supervised and Semi-Supervised Learning for Failure Identification in Microwave Networks." IEEE Transactions on Network and Service Management, 18.2 (2020): 1934-1945.

[2] F. Coenning, "Understanding itu-t error performance recommendations,"https://www.julesbartow.com/Pictures/ITS/ITU-TErrorsApplicationNote2.pdf

[3] Ribeiro, Marco et al. "Why should I trust you?" Explaining the Predictions of Any Classifier." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

[4] Lundberg, Scott M., and Su-In Lee. "A unified Approach to Interpreting Model Predictions." Proceedings of the $31^{st}$ International Conference on Neural Information Processing Systems, 2017.

[5] Du, Mengnan, et al. "Techniques for Interpretable Machine Learning." Communications of the ACM 63.1 (2019): 68-77.

[6] Došilović, Filip Karlo, Mario Brčić, and Nikica Hlupić. "Explainable Artificial Intelligence: A Survey." 41st IEEE International Convention on Information and Communication Technology, Electronics and Microelectronics, 2018.

[7] Roscher, Ribana, et al. "Explainable machine learning for scientific insights and discoveries." Ieee Access 8 (2020): 42200-42216.

[8] Lipton, Zachary C. "The Mythos of Model Interpretability: In Machine Learning, the Concept of Interpretability is Both Important and Slippery." Queue 16.3 (2018): 31-57.

[9] Doshi-Velez, Finale, and Been Kim. "Towards a Rigorous Science of Interpretable Machine Learning." arXiv preprint arXiv:1702.08608 (2017).

[10] Gilpin, Leilani H., et al. "Explaining Explanations: An Overview of Interpretability of Machine Learning." 5th IEEE International Conference on Data Science and Advanced Analytics, 2018.

[11] Miller, Tim. "Explanation in Artificial Intelligence: Insights from the Social Sciences." Artificial Intelligence 267 (2019): 1-38.

[12] Linardatos, Pantelis, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. "Explainable AI: A Review of Machine Learning Interpretability Methods." Entropy 23.1 (2021): 18.