ITUJournal
Future and evolving technologies

ITU Journal on Future and Evolving Technologies, Volume 5, Issue 3, September 2024

Article

Automated PIM Detection in Wireless Networks via Smart Data-Driven Techniques

Gokcan Cantali[1,2], Eren Deniz[3,4], Hasan Ferit Eniser[5], Onur Yildirim[3], Gürkan Gür[2], Fatih Alagoz[1]

[1] Bogazici Univresity, Department of Computer Engineering, Istanbul, Turkey, [2] Zurich University of Applied Sciences (ZHAW), Institute of Computer Science (InIT), Switzerland, [3] Yupana Inc, Walnut Creek, USA, [4] Ege Univresity, Department of Computer Engineering, Izmir, Turkey, [5] MPI-SWS, Germany

Corresponding author: Gokcan Cantali, goekcan.cantali@zhaw.ch

As the number of base stations keeps increasing due to ever-growing wireless services and pervasive connectivity, the operational costs for network service providers rises proportionally. In addition, with the evolution of wireless technologies and the development of next generation networks such as 5G and 6G, network operators are inevitably expected to encounter more complex challenges. One such network issue is the Passive Intermodulation (PIM) problem that is observed in both 4G and 5G networks. It degrades the user experience and radio resource efficiency while leading to an operational overhead for detecting and mitigating it on the operator side. Although there is a significant body of work regarding PIM detection and cancellation methods, the majority of such studies depends on hardware solutions and manual investigation by network engineers, which is costly in terms of time and labor. In this paper, we propose two methods using unsupervised and semi-supervised Machine Learning (ML) approaches, namely a time-series-based anomaly detection technique and an autoencoder-based one, for identifying PIM problems in network sites. The proposed solutions utilize a set of Key Performance Indicator (KPI) data of base stations obtained from network management systems for a significantly long time interval and detect possible PIM problems without the need for a human in the loop. We measure and analyze the performance of our solutions, with the guidance of experienced network engineers, on our collected dataset.

Keywords: Machine Learning (ML) for communications, Passive Intermodulation (PIM), Quality of Service (QoS), Radio Access Networks (RANs)

1. INTRODUCTION

Wireless and mobile networks have become the backbone of our civilization with the emergence of anytime-anywhere connected services and advances in end-user devices. This phenomenon was also starkly evident during the COVID-19 pandemic when ubiquitous broadband access and applications like remote working were critical for our daily lives [1]. As a consequence, more base stations and radio access nodes are being deployed in the cellular network infrastructure to facilitate broadband connectivity and adequate capacity. This trend is accompanied by more complex radio technologies and diverse radio environments for network devices with the deployment of 5G networks. However, the increasing number of base stations also implies more labor for maintaining network sites. Due to the complex nature of the technology used in these sites, network problems nowadays have also become more complicated compared to those in the past. On top of emerging network problems, older issues that were prominent in 4G networks, such as Passive Intermodulation (PIM), still pose a significant threat to network coverage and signal quality in today's wireless networks, adversely impacting network coverage and signal quality in current wireless systems [2].

The PIM issue stems from the nonlinearity of RF antennas or passive network elements such as connectors and cables, and causes inferior signal quality and might even result in connection drops [3]. In such a setting, the nonlinear function of the passive device can be characterized by the power series $y = \Sigma_{k=1}^{\infty} a_k x^k$ where x and y represent instantaneous input and output signals, respectively. The coefficient a_k depends on the nonlinear characteristics of the device, with k being the order of the power series. For instance, when the input signal is composed of two frequencies f_1 and f_2, there will be intermodulation signals in the output signal in addition to harmonic ones due to the combination of these two frequencies as shown in Fig. 1 [4]. These intermodulation signals around the fundamental frequency lead to the PIM issue. Such nonlinearity, when caused by slow physical degradation of network equipment over time, is called *Internal PIM* [5]. Occasionally, the root cause of the PIM is not the internal components, but an external factor unrelated to the site network itself. This type of PIM, named *External PIM*, can be encountered when there are metallic objects such as a fence, billboard, or barn roof around the signal path. Unlike the Internal PIM problem, which shows its symptoms slowly over time, the External PIM problem might cause an abrupt change in signal quality and overall network performance, depending on the external factor. The reason for such a situation is that in an external PIM problem, a new object, which was not present before, is introduced into the RF transmission line. The nature of such a problem is different than that of an internal PIM, where the interference is caused by the gradual degradation of passive network elements.

Current PIM detection and cancellation methods typically depend on manual work and lead to further operational costs, such as artificially loading a Radio Access Network (RAN) cell to identify the problem [6]. Therefore, remote solutions relying on the processing of network monitoring data are actively investigated in the technical community. These schemes may employ a variety of approaches, such as ML and statistical analysis. In that regard, anomaly detection [7] is promising for PIM detection since PIM emerges as a performance anomaly and thus impacts related network metrics.

Without a reliable remote detection technique, RAN teams are required to perform physical site visits to confirm that there is a PIM problem in a network site. Once they arrive at the site, they try to ensure that there is actually PIM by usually increasing the load on the site via artificial load generators such as the Air-Interface Load Generator (AILG), forcing the site to display the effects of PIM under heavy traffic. Sometimes, they also use additional techniques, such as Distance-to-PIM (DTP), to pinpoint the location of the PIM when they decide that the network site suffers from PIM. Although remote solutions cannot be used to solve the PIM completely, they can greatly make the process of detection more efficient.

Anomaly detection is already a critical component of monitoring and maintaining the performance of mobile networks. The rapid growth in the number of mobile devices and the multiplying complexity of mobile network architectures have made it increasingly difficult to identify and troubleshoot issues in a timely manner. Anomaly detection methods use statistical models and ML techniques [8, 9] to automatically identify abnormal behavior in the network, including PIM issues. By detecting anomalies in a timely manner, network operators can take proactive measures to prevent or mitigate problems, improving the overall reliability and performance of the network for end users. As such, anomaly detection is an essential tool for ensuring efficient and secure operation of mobile networks.

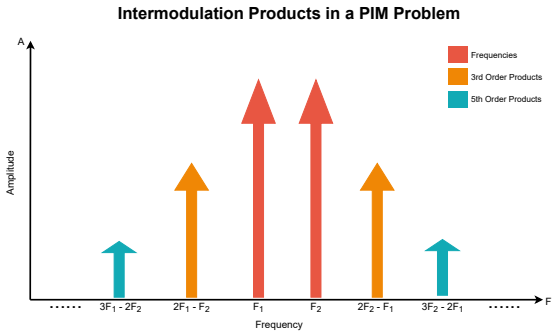


Figure 1 – Intermodulation products for two signals at F1 and F2 in PIM.

In this study, we expand our recent work [2], which presented an anomaly detection solution for detecting possible PIM occurrences in RAN, namely Temporal Analysis-Based Anomaly Detection (TABAD). A complete unsupervised approach is useful for the PIM detection challenge since obtaining a large dataset of labeled sites is quite difficult for this problem. However, in some cases, a network vendor can access a certain amount of labeled site data that could be utilized for PIM prediction. Therefore, in this work, we introduce an additional technique denoted as Classifiers with Autoencoder Preprocessing Stage (CAPS) that utilizes strengths of both supervised and unsupervised methods. Our proposed techniques utilize a set of Key Performance Indicator (KPI) data, obtained from the site base station in a time-series format. Therefore, these solutions do not require manual input to determine when the site starts to display the effects of PIM. We measure the performance of our proposals on a collected KPI dataset of base station cells in a region.

The rest of the paper is organized as follows. In Section 2, we discuss the related work on various solutions for PIM detection in the literature. Section 3 starts with

providing background information on anomaly detection techniques and autoencoder-based approaches and then proceeds to introduce our proposed solutions in detail. In Section 4, we first explain our methodology for data collection and preparation stages and then present the performance results of both of our solutions on the collected dataset. We discuss the differences between our proposed solutions and mention some identified research challenges in Section 5 and Section 5.2, respectively. Finally, Section 6 concludes the paper and delineates future work.

2. RELATED WORK

Although this section mostly covers previous studies regarding the PIM problem, we also provide a brief overview of the works on anomaly detection techniques and autoencoder usage in wireless networks, given that we provide two separate PIM detection solutions, an anomaly detection method, and an autoencoder-based classification technique.

Due to the high number of undesired effects of emerging PIM problems, several studies have been proposed in the past in an attempt to tackle the issue in various aspects. Some of these studies propose solutions for the early stages of a new site, to solve possible PIM issues that might arise in the design phase. One such study by Chen et al. [10] provides an approach to reduce the PIM severity via the finite element simulation method, resulting in a more stable antenna design. A similar PIM simulation technique is presented by Wu et al. in [11], for detecting PIM issues specifically for mesh reflector antennas, allowing a design with lower levels of PIM. Yang et al. consider the possibility of non-linearity of flange-mounted coaxial connectors in their study [12], where they introduce an improved design to reduce PIM problems and increase the reliability for such components. The underlying physical mechanisms of the PIM problem are studied by Wang et al. in [13], where the authors explain possible root causes of non-linearity in the RF transmission line as guidance for optimal antenna design.

Apart from improving the design of new network base stations, a significant amount of work has been conducted to detect and eliminate the symptoms of PIM for already operating sites. For the sites that suffer from PIM, many operators utilize special hardware equipment such as Range-to-Fault (RTF) by Kaelus [14] and Distance-to-PIM (DTP) by Anritsu [15], to locate the exact source of the PIM. However, there are some cases where PIM analyzers might fail to produce results that are accurate enough to locate the exact component causing the issue. That kind of limitation has motivated researchers to adopt other approaches, such as near-field scanners by Shitvov et al. [16], a vibration modulation method by Yang et al. [17] and a super-resolution imaging technique by Li et al. [18] to achieve a more precise measurement and decrease the number of potential elements in the transmission line. Given that most PIM analyzers are designed for narrowband systems, Zhu et al. [19] follows a different approach and presents a novel technique for detecting PIM issues in ultra-wideband systems, using the principle of cooperative RF-baseband nulling. Moss et al. [20] puts an emphasis on the degradation of RF switches and introduces a process for testing and measuring the non-linearity of such devices in an efficient manner to avoid potential PIM problems. Implementing a novel method for PIM evaluation is also a part of the FutureCom project [21].

Solving the PIM problem is the next challenge after finding out the location of the faulty component. For instance, Lampu et al. focus on air-induced external PIM issues and propose PIM cancellation techniques specifically for two carrier components and four carrier components in [22] and [23], respectively. In a more recent work, the same authors also provide a more efficient solution [24] for the matter. In addition to air-induced problems, a general perspective for PIM issues is considered by Waheed et al. [25], where they introduce a cancellation method for 4G and 5G sites that operate on frequency-division duplexing technology. Furthermore, Waheed et al. also propose a compilation of PIM cancellation methods in [26].

The aforementioned studies focus on the sites that are already known to exhibit the effects of PIM. However, determining whether an RF antenna actually suffers from PIM, is not always a trivial matter. More than frequently, detecting the PIM issue on a network site is a challenging task, given that PIM symptoms are less visible when the base station receives low traffic [5]. Under such circumstances, the effects of PIM will become apparent in case the site starts to receive a higher volume of traffic than usual, causing the network operators to abruptly encounter coverage issues. To overcome such unwanted scenarios, artificial load generators, such as Air-Interface Load Generator (AILG) [27], have been proposed. Such load generators operate by creating an artificial load to temporarily increase the traffic volume in an attempt to investigate coverage issues in a clearer way and determine whether the site really suffers from PIM or not. Despite providing accurate detection results, these load generators bring along some negative side effects, such as a decrease in signal quality of the base station, while the tests are being executed. Moreover, AILG or similar tests require network engineers to be physically relocated to the target network site, incurring additional logistic costs.

Considering the disadvantages of artificial load generators and physical site visit requirements, a need for

remote PIM detection techniques by mobile network operators has emerged. Researchers who specialize in the area of wireless network technologies, specifically in PIM detection, have tried various approaches to tackle this aspect. A fairly recent patent work has proposed to incorporate a rule-based mechanism into statistical data of a network site's KPI values, to determine whether or not a site suffers from PIM [28]. ML models are also employed in the PIM detection problem, as Jochems et al. present a study focusing on detecting and clearing the self-interference on network sites, which is considered a superset for the PIM problem, using a novel ML technique called canonical system identification [29]. Supervised ML methods are also commonly used in the domain of wireless networks, hence PIM problem detection, such as recursion learning neural networks [30] and linear regression models [31] for satellite networks and 6G networks, respectively.

2.1 Anomaly detection for PIM problem

Since PIM can be considered an unusual state that only exists due to abnormal conditions on the network site, treating PIM-related issues as an anomaly becomes a rational option. This choice allows researchers to utilize anomaly detection techniques for detecting PIM problems. Based on this view, network KPI data can be obtained from the related base station in a time-series format, and anomaly detection techniques can be applied to this data. This idea, with the addition of Fourier feature mapping, is used in the work of Banerjee et al. [32], to forecast possible PIM issues in 4G and 5G networks. Ranjani et al. [33] propose a similar but different method that is mainly an ensemble technique where the time-series KPI data is used to train random forest models to detect PIM problems in 4G and 5G network sites. The study differs from other approaches in the sense that the state of the neighbor sites is also considered during the evaluation of PIM possibility in a site.

Another study by Banerjee et al. takes an entirely different approach and utilizes self-supervised reinforcement learning to proactively predict PIM problems before they occur [34], utilizing the historical data of some external factors such as weather conditions. In our previous work [2], we also propose a time-series-based anomaly detection technique for identifying PIM problems in 4G sites and test the performance of our model on KPI data collected over a long period of time.

2.2 Autoencoders in wireless networks

Due to the drastic increase in the number and size of collected datasets, deep learning has become a significantly popular concept among machine learning studies during the last two decades. The basis of deep learning algorithms, Artificial Neural Networks (ANNs), was inspired by the structure of neurons in a brain and modeled accordingly, with the assumption that simulating the human brain would enable the machine to learn new pieces of information in a more human-like manner [35]. ANNs are widely used in various applications, based on different machine learning techniques. For instance, Convolutional Neural Networks (CNNs) are supervised ANN models that are intensely utilized in image and video recognition [36]. Similarly, Recurrent Neural Networks (RNNs) can be used in a reinforcement learning paradigm for solving a vehicle routing problem [37]. In addition, ANNs can also be used in an unsupervised setup in the form of autoencoders, which were first indirectly mentioned by Rumelhart et al. in [38] and then by Ballard in [39]. An autoencoder is a special type of neural network that is trained to copy its input to its output. Given an input with n dimensions, an autoencoder first encodes the input into a compressed representation and then decodes that representation back to an output that is supposed to be equal to the input. Therefore, an autoencoder learns an encoder and a decoder. Autoencoders are commonly used in multiple applications, including, but not limited to, dimensionality reduction [40] and lossy image compression [41].

Autoencoders are also used in the domain of wireless networks, mainly for detecting novelties in a network. For instance, Dawoud et al. present a study where autoencoders are used as a type of intrusion detection system to detect abnormal traffic in the network that might be caused by potential malicious intruders [42]. Another study is conducted by Basak et al. in [43], where the authors propose an autoencoder-based framework, called DEFEND, for both identifying signals from known drones and detecting unknown drone signals. While the former goal is achieved through a semi-supervised classification method, the latter task is performed via an unsupervised novelty detection technique, both of which utilize autoencoder models.

Another work that utilizes deep learning concepts for wireless networks is presented by Ahasan et al. [44], in which autoencoder models are used to detect outliers in the performance data fetched from 4G networks and tries to obtain the optimal configuration for such models. Autoencoders are also applied in the network security domain, such as detecting DDoS attacks in an efficient manner by using time-based features of the network and treating attacker activities as an anomaly [45].

Given that autoencoders are mostly used in an unsupervised method, such as novelty detection, comparing them to traditional anomaly detection techniques becomes an interesting research topic. Smolen and Benova [46] compare the performance of autoencoders and isolation forest, on a dataset with a large number of sam-

ples, for spotting malicious cyber-activities, specifically zero-day attacks, within the network. A similar study is conducted by Hindy et al. [47], in which an autoencoder-based model is proposed to detect zero-day attacks, and the model is evaluated against a One-Class SVM as a benchmark.

Although some of these studies do not have a direct relation with the problem of PIM detection, the insights they provide can be useful in this domain as well. In the end, the existence of a PIM problem is an anomaly that causes a drop in signal quality. In that regard, a DDoS attack aiming to degrade the service of a network entity or an unauthorized signal causing additional, undesired load in the network, can be interpreted as a similar research problem. Of course, the solutions that follow the detection of these problems are very different compared to a PIM mitigation technique. Nevertheless, the scope in this study is limited to finding the existence of a PIM issue in a wireless network.

3. PIM DETECTION FROM TWO PERSPECTIVES: STATISTICAL LEARNING VS SEMI-SUPERVISED APPROACHES

In this work, we propose two different techniques, one based on statistical analysis, i.e., TABAD, and another using semi-supervised ML, i.e., Classifiers with Autoencoder Preprocessing Stage (CAPS). For the former approach, we formulate PIM detection as an anomaly detection problem, whereas deep learning and autoencoders are the focus for the latter approach.

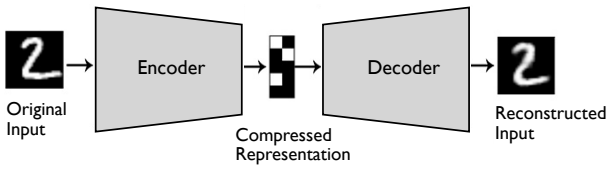


Figure 2 – Autoencoder operation.

For the PIM detection process, we assume that the KPI data of the network can be autonomously retrieved from base stations at certain intervals, making it possible for our methods to detect a PIM issue in a real-time manner. In addition, we focus on 4G networks during the study, since our collected data contains primarily LTE sites. Our two methods, TABAD and CAPS, operate on different strategies regarding feature selection.

TABAD operates on a single feature, thus requiring us to choose one KPI among many. During the KPI selection process, we received significant assistance from the RAN engineers actively working on the PIM problem in the field. These experts explained that PIM issues primarily affect the receiver antenna due to the nature of the problem and, consequentially, advised to investigate the KPIs related to the Received Signal Strength Indicator (RSSI) of interference in the Uplink (UL) channels. Uplink channels are also categorized into two: Physical Uplink Control Channel (PUCCH) and Physical Uplink Shared Channel (PUSCH). Therefore, two target KPIs would be the RSSI_INTERFERENCE_PUCCH and RSSI_INTERFERENCE_PUSCH, which are, from now on, referred to as PUCCH and PUSCH, respectively. These KPIs are measured in the unit of dBm and their values usually vary between -100 and -120, where the higher values (e.g., -100) indicate major interference and lower values (e.g., -120) indicate minor interference. However, the expected value of these KPIs mostly depends on the properties of the respective network site, such as geographical location, nearby population, and climate. The RAN engineers explained that they have been using PUCCH and PUSCH in an attempt to identify PIM problems for a while and have obtained remarkable results. On top of that, the company also has a patented solution for PIM detection [28], which is based on these two KPIs. Considering such details, we decided to focus on these KPIs for our TABAD solution.

Our other solution, CAPS, on the other hand, operates in the combination of multiple KPIs, allowing us to feed the entire KPI set into the model. The autoencoder network inside CAPS handles the feature extraction task, in such a way that the most relevant KPIs can be used based on the input data. The flow of an autoencoder operation is shown in Fig. 2.

3.1 Temporal Analysis-Based Anomaly Detection (TABAD)

TABAD is an anomaly detection technique that is better suited for time-series data in the PIM use case. According to this technique, we analyze characteristics of previous data collected over time and compare current events against them. Well-known anomaly detection techniques, such as isolation forest, handle a dataset as separate and independent data points which includes anomalies of a certain ratio. In contrast, TABAD considers a stream of data in which we check whether the current data behaves like an anomaly based on previous data. The study in [48] indicates that using a sliding window on the data is common, given the nature of the time-series data. In addition, the study also emphasizes that seasonality is a key factor to consider during the anomaly detection process on time-series data. The seasonality has a significant effect on the PIM problems as well since the load on a network site follows a weekly pattern. For example, while a network site would not likely suffer from heavy load during the quiet hours on weekends, the network might be overwhelmed with too much load during the busy hours on weekdays. Hence, we conclude that using a window of past KPI data would

produce fewer false positives (i.e., false alarms) and thus increase the precision of the detection method, due to the seasonality of the KPI data.

Algorithm 1: TABAD algorithm

Input: *hist*, *prb* : time-series like data
Output: *true* : anomaly, *false* : normal
Function *tabad(hist, prb: Array) : bool*
    histIntr ← *splitTimeIntervals*(*hist*);
    histChrc ← *getChrcIndicators*(*histIntr*);
    baseAvg, *baseVar* ← *normal*(*histChrc*);
    prbChrc ← *getChrcIndicators*(*prb*);
    return *ZTest*(*prbChrc*, *baseAvg*, *baseVar*)

Algorithm 1 outlines the core steps of our approach. It takes an array of historical data (*hist*) that will be used to form a base for normal operation and another array of data (*prb*) which is under examination. The function starts with splitting the historical data (*hist*) into time intervals (*histIntr*) and inferring characteristics (*histChrc*) for each time interval of historical data. In the next step, we find the mean (*baseAvg*) and the variance (*baseVar*) of calculated characteristics, which lets us define a normal data distribution establishing a ground truth for normal operation. Subsequently, we calculate characteristics (*prbChrc*) of the data under examination (*prb*) and apply statistical test (*ZTest*) that checks whether given data is coming from the given distribution or not. The outcome of this test is simply the output of the function.

Algorithm 1 outlines the core steps of our approach. It takes an array of historical data that will be used to form a base for normal operation and another array of data that is under examination. The function starts with splitting the historical data into time intervals and inferring characteristics for each time interval of historical data. In the next step, we find the mean and the variance of calculated characteristics, which lets us define a normal data distribution, establishing a ground truth for normal operation. Subsequently, we calculate the characteristics of the data under examination and apply a statistical test that checks whether the given data comes from the given distribution or not. The outcome of this test is simply the output of the function.

Fig. 3 illustrates our general PIM detection scheme using TABAD. According to this, our system monitors streaming KPIs in real time and checks whether the values in the current window present an anomaly or not. Although our design allows monitoring multiple KPIs simultaneously, the decision is made separately for each KPI. Decisions for different KPIs can be aggregated in several ways. It is a new research problem on its own and requires careful analysis. In this work, we apply simple voting among multiple decisions.

TABAD can use either the mean or the variance of the

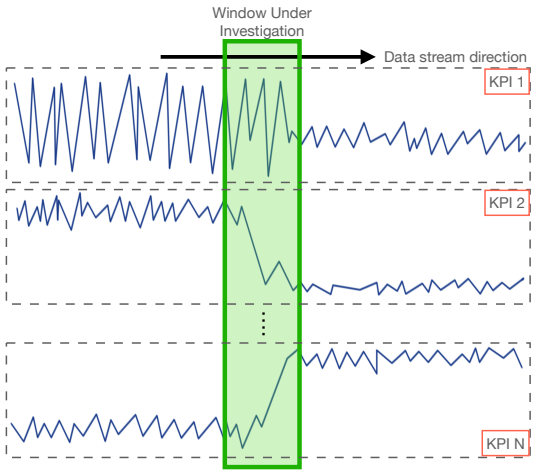


Figure 3 – Real-time PIM analysis with TABAD.

examined data to determine whether there is a possible PIM issue, because both the mean and the variance are relevant indicators for different KPIs. For example, for PUCCH-based KPIs, the mean value usually increases in cells with any PIM issue. For some other KPIs, PIM might affect the variance rather than the mean value. We start with finding the mean (or variance) of the KPI values collected over a current time window of one-week duration. We also find the mean (or variance) of values collected over previous weeks. We assume that those mean (or variation) values follow a normal distribution, and we check whether the current week's mean (or variation) is drawn from that distribution.

3.1.1 TABAD hyperparameters

TABAD approach utilizes the following three hyperparameters:

- History Size (HS)
- Window Under Investigation (WUI)
- Flag Level Threshold (FLT)

HS determines the amount of past KPI data (in days) the algorithm considers when predicting anomalies. For instance, if *HS* is 10, the algorithm will use the KPI data of the previous ten days to decide whether the site currently suffers from a PIM problem. Increasing *HS* is likely, though not guaranteed, to increase the accuracy of the predictions. However, keeping it too long might result in missing the latest trends in the data.

WUI tells the algorithm to treat a certain amount of KPI data (in days) as a whole, such that the algorithm considers this entire batch at once while determining whether there is a PIM problem. Before delving into the details, we would like to note that the PIM problem does not dissolve by itself after the occurrence. In other words, once a PIM issue arises in a network site, the problem

will persist until RAN engineers fix it manually. Therefore, providing the operator with a single alarm once the PIM issue is first detected is sufficient. Given this information, WUI acts as a throttling mechanism to prevent consecutive alarms for the operator. For instance, if *WUI* is 7, the algorithm considers the KPI data for the next seven days and either marks the entire seven days as an anomaly or does not mark any of these days. The most effective way of choosing WUI is to determine the periodic behavior of the KPI signal (e.g., weekly cyclic behavior).

The last hyperparameter is basically the threshold value *FLT* for the anomaly decision. The TABAD algorithm determines that a KPI should only be marked as an anomaly if the probability of this KPI value appearing on a healthy site is less than the value defined by this threshold. Decreasing *FLT* makes the algorithm more conservative for predictions, thus decreasing the false PIM alarms but increasing the potential misses. In contrast, increasing the *FLT* value makes the process more sensitive, increasing the correct predictions, but also causing more false alarms.

Our algorithm for predicting the existence of a PIM issue is given in Algorithm 2.

Algorithm 2: Anomaly Alarm Decision

Input: $zScore$: $prob.$, $fltRed$: $prob.$, flt : $prob.$
Output: *PIM* : PIM exists, *NoPIM*: PIM does not exist
Function *decidePIMAlarm(zScore: float, flt: float)* → *pimAlarm: string*
  pimAlarm ← *null*;
  if $zScore <= flt$ **then**
    pimAlarm ← *"PIM"*;
  else
    pimAlarm ← *"NoPIM"*;
  return *pimAlarm*

3.2 Classifiers with Autoencoder Preprocessing Stage (CAPS)

Our second proposed algorithm, CAPS, is a PIM detection method in the form of a semi-supervised classification model, consisting of an autoencoder preprocessing stage and a classifier, as shown in Fig. 4. We first train an autoencoder model using all of the available KPIs (37, to be exact) of the given training dataset. Then, using that trained autoencoder model, we encode the KPI values of the training dataset to obtain encoded KPI values. This step concludes the unsupervised phase of the model. Then, by feeding these encoded KPI values into a classifier, we train the model via the labels corresponding to each site and time period. In other words, the encoded KPI values and their corresponding labels are used as input for training a classifier, which is basically the supervised phase of our CAPS solution. The classifier's predictions, in the end, are used to test whether a target site suffers from PIM or not. Thus, the entire model is considered a semi-supervised solution for detecting PIM problems. We integrate four different classifiers, namely *SVM*, *Random Forest*, *MLPClassifier*, and *Naive Bayes*.

4. EVALUATION

4.1 Benchmark

We have collected KPI values of 37 4G RAN cells over a period of 18 months. Our aggregated data contains 37 KPIs. However, we consider two different KPI sets for our techniques in this work, since the two solutions differ in terms of their operation principles. We consider only PUCCH and PUSCH-based KPIs for TABAD, whereas the entire KPI set is utilized for CAPS. Although our data set contains a relatively small number of cells in a zone, they were monitored over a very long period. This is necessary because PIM usually occurs over such long time horizons. All of our 37 cells are known to have the PIM problem at some point within these 18 months. In addition to this situation, these cells were physically visited by field teams to fix their PIM problem after its occurrence as part of RAN maintenance work.

4.2 Experimental design

Unfortunately, it is not always possible to identify exactly at which point PIM emerges, even with expert views from maintenance teams with comprehensive hands-on experience. On the other hand, we have cell visit time information that corresponds to the point where PIM is fixed. Our history-based anomaly detection technique, i.e. TABAD, is sensitive to any change regardless of PIM emergence or PIM resolution. For this reason, we consider site visit dates as ground truth anomalies (i.e., PIM fix events), and we check whether our technique can identify them or not. Consequently, we report True Positive (TP), False negative (FN), False Positive (FP), precision and recall as performance metrics for TABAD, where precision represents the percentage of correct PIM alarms over the number of all alarms and recall represents the percentage of correct PIM alarms over all the PIM incidents.

On the other hand, CAPS accepts a list of KPIs as input and predicts whether the given data is anomalous (PIM-positive) or normal. In our evaluation, we train one autoencoder with daily KPI data collected from multiple cells over around 18 months. Later, we train a classifier per cell with daily KPI data whose labels are available (e.g. day_1 is PIM-positive, day_24 is normal, etc.).

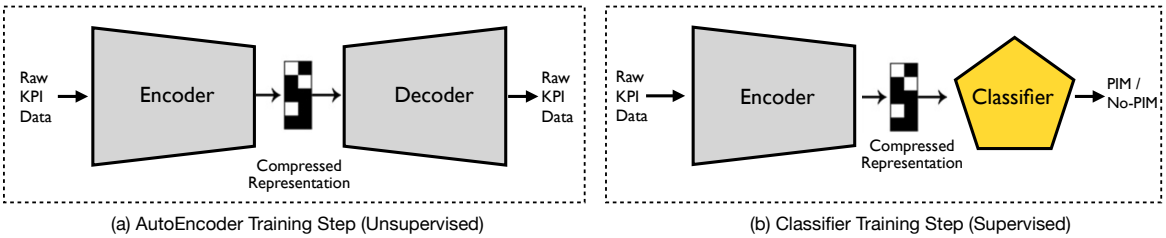


Figure 4 – Overview of the CAPS training procedure.

Thus, final predictions of CAPS refer to whether the cell was PIM-positive or not on a given day. For this reason, all TP, FP, TN, and FN are relevant, and we report accuracy, F1 score, precision, and recall, which summarize those raw numbers from various aspects.

4.2.1 *Baseline method for TABAD: isolation forest*

From the perspective of TABAD, the PIM problem is a rare occurrence that should normally not exist in healthy sites, making TABAD essentially an anomaly detection method that operates in wireless networks. In that regard, TABAD can be compared to other anomaly detection techniques that are mainly used in wireless networks. Therefore, we compare TABAD's performance to a baseline method, namely isolation forest, in our performance evaluation.

Many anomaly detection algorithms construct a profile for normal behavior, which serves as a ground truth when deciding whether an event is an *anomaly*. However, isolation forest [49] takes a different route to isolate anomalies from normal events, where isolation refers to separating an event from the rest. At its core, it takes advantage of anomalies being "few and different" and constructs a tree structure that enables identifying anomalies in its leaves.

Isolation forest recursively generates partitions on the dataset by randomly selecting a feature and then randomly selecting a split value for that feature. Presumably, the anomalies need fewer random partitions to be isolated compared to normal points in the dataset, so the anomalies will be the points that have a smaller path length in the tree, the path length being the number of edges traversed from the root node.

In the following, we provide a step-by-step procedure of an isolation tree construction:

1. Given a dataset, a random sub-sample of the data is selected and assigned to a binary tree.
2. Branching of the tree starts by selecting a random feature of the input. Then, branching is applied on a randomly determined threshold value of the selected feature.
3. If the value of an input is less than the selected threshold, it goes to the left branch, else to the right.
4. This process is continued until each data point is completely isolated or the maximum depth threshold is reached.

The foregoing steps are repeated to construct a number of random binary trees (i.e., a forest). After an ensemble of trees is created, model training is complete. During scoring, a data point is traversed through all the trees that were trained earlier. An 'anomaly score' is assigned based on the depth of the node that the data point reaches. The overall score is an aggregation of the depth obtained from each of the trees in the trained model.

4.3 Experimental results

In this section, we present and analyze our experimental results to investigate the following research questions:

1. **TABAD and CAPS performance:** *What is the performance of TABAD and CAPS in detecting PIM via KPI signals?*
2. **Comparison of TABAD to a baseline method:** *How well does TABAD perform compared to a baseline anomaly detection method?*
3. **Impact of hyperparameters on TABAD and CAPS:** *How do the performances of TABAD and CAPS get affected by changes in its hyperparameters (i.e., configuration)?*
4. **TABAD vs CAPS:** *How do CAPS and TABAD perform, with regard to each other, for a PIM detection problem?*

4.3.1 *Anomaly detection approach: TABAD performance*

First, we start with the evaluation of our TABAD model and delve into the research questions specific to that model.

4.3.1.1 Overall TABAD performance

For our study, as described in Section 4.1, the expert RAN engineers from the field teams provided us with a dataset of 37 network sites, which used to suffer from a PIM problem before a site visit. According to the explanation by experts, 25 of these sites show improvements in their KPIs after a site visit. However, 12 of them do not display an apparent change in their KPIs due to being located in a sparsely populated area, hence receiving low traffic demand all the time. Therefore, we were advised to exclude these 12 sites from our experiment, to avoid having corrupted/biased data impacting the results.

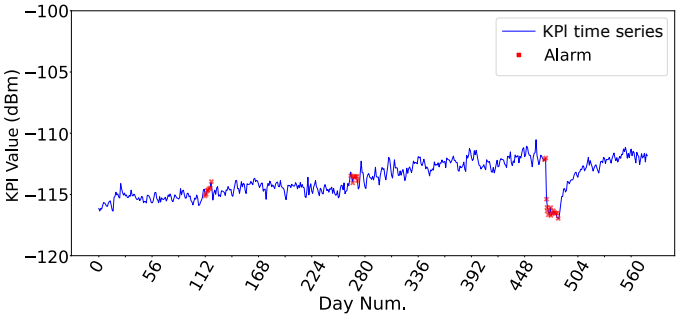
We measure the performance of our method by checking whether an abnormal change occurs right after a site visit. We consider an alarm raised right after that visit an anomaly detection. If there are any other alarms besides the site visit date, it is considered a false alarm. Similarly, if the algorithm misses the alarm at the visit date, it is considered a false negative.

Based on the simulation results conducted for 25 network cells, our method correctly identifies 20 of the 25 site visits, while five of the site visits are missed. Therefore, we have 20 true positives and 5 false negatives. On the other hand, our method triggers 40 false alarms irrelevant to site visit dates. Computing true negatives in this setup is a tricky task because the data is almost always considered normal except for the site visit date. Therefore, we omit this metric here. Using the formulas, our approach achieves a recall of 80% and a precision of 33.3%.

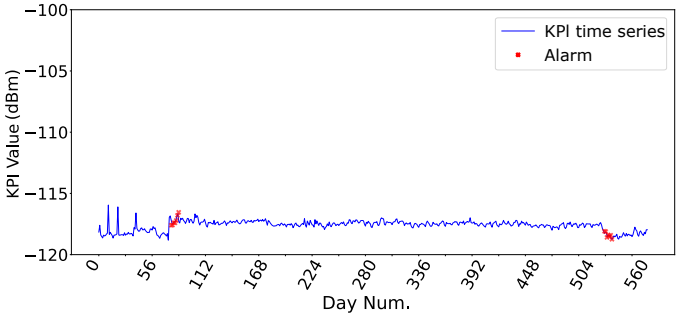
4.3.1.2 Comparison with isolation forest

When we simulate the same setup with the isolation forest method, we obtain 14 correctly identified and 11 missed sites, which has a recall of 56%. However, isolation forest produces more than 300 false alarms for the tested 25 sites. Therefore, the precision value of the method is less than 5%. One can observe from the results displayed in Table 1 that our method performs significantly better for both precision and recall compared to the isolation forest method.

One can observe from Fig. 5a that TABAD successfully detects the PIM start point at Day ~282 where the PUCCH value is approximately -114 dBm and the PIM resolution point at Day ~476 where the PUCCH value is approximately -118 dBm. TABAD also makes a mistake by marking Day ~112 as an anomaly, even though that is not where the PIM starts or ends. Similarly, in Fig. 5b, TABAD can be seen detecting the correct PIM start date at Day ~84 with a PUCCH value of -117 dBm and the correct PIM end date at Day ~522 with a PUCCH value of -119 dBm.



(a) TABAD method on Cell 1



(b) TABAD method on Cell 2

Figure 5 – TABAD method marks anomalies in PUCCH signals in two different cells. Experts confirm that the first mark corresponds to the point where PIM starts. Also, the last mark is where the PIM is solved by field teams.

Table 1 – Performance results for TABAD and isolation forest

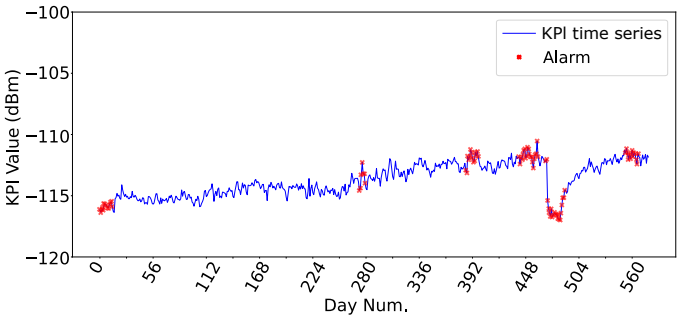
	Isolation Forest	TABAD
Precision	< 0.05	0.33
Recall	0.56	0.80

Isolation forest, on the other hand, correctly determines the PIM start and end dates, as can be seen in Fig. 6a. Nevertheless, the algorithm also gives out numerous false alarms throughout the duration. Especially in Fig. 6b, isolation forest marks from Day 0 to Day ~84 as anomalous, which is clearly a wrong conclusion.

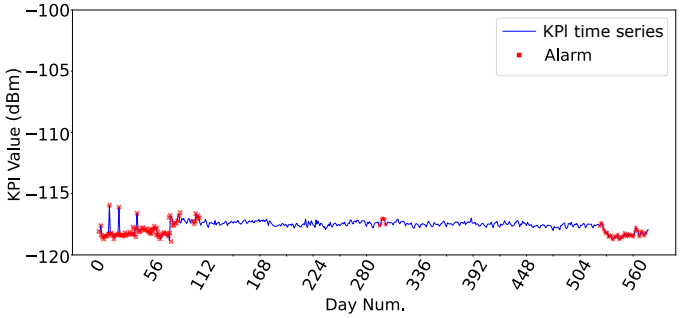
4.3.1.3 Impact of TABAD hyperparameters

During our TABAD experiments, we used various values for our hyperparameters, *HS*, *WUI* and *FLT*. We used grid-search to check through a range of values for each hyperparameter, and we empirically observe that our algorithm performs best when the hyperparameters are set to values provided in Table 2. These values achieve a precision of 33% and a recall of 80%, as mentioned before.

When we increase the value of *HS*, we observe that the number of false positives decreases significantly, at the expense of increasing false negatives. Surprisingly, using a *HS* value lower than the optimum causes an in-



(a) Isolation forest on Cell 1



(b) Isolation forest on Cell 2

Figure 6 – Isolation forest anomaly detection method marks anomalies in PUCCH signals in two different cells. It marks a lot of unrelated points in addition to the correct anomaly points, which greatly reduces its practical usage.

Table 2 – Optimal values for TABAD hyperparameters

Hyper-Parameter	Optimal Value
HS	10 weeks
WUI	1 week
FLT	0.01

crease in both false positives and false negatives. Examples of different *HS* values are given in Table 3, along with corresponding performance results, where other hyperparameters are set to their optimum values.

When we use a *WUI* higher than the optimum value, the number of false negatives increases while the number of false positives decreases in a drastic manner. Using a *WUI* lower than the optimum value increases both false negatives and false positives. A pair of different *WUI* values and the resulting performance results are given in Table 4. Again, other hyperparameters are set to their optimum values during these experiments.

Table 3 – Different values for TABAD hyperparameter HS

HS Value	TP	FN	FP	Precision	Recall
5 weeks	12	13	120	0.09	0.48
15 weeks	8	17	20	0.28	0.32

Table 4 – Different values for hyperparameter WUI

WUI Value	TP	FN	FP	Precision	Recall
3 days	14	11	150	0.08	0.56
2 weeks	12	13	10	0.54	0.48

Table 5 – Different values for hyperparameter FLT

FLT Value	TP	FN	FP	Precision	Recall
0.005	13	12	30	0.30	0.52
0.025	21	4	60	0.26	0.84

Since the *FLT* parameter directly acts as a threshold, we expect an increase in false negatives and a decrease in false positives when *FLT* becomes stricter. Similarly, when *FLT* is loosened, an increase in false positives and a decrease in false negatives is expected. The results of our trials suggest that our expectations are correct, as one can observe from Table 5, where other hyper-parameters are set to their optimum values.

4.3.2 *Switching to semi-supervised learning: CAPS performance*

After explaining the results of the TABAD technique in detail, we now present an extensive evaluation of the CAPS model.

4.3.2.1 Overall CAPS performance

For the experiments on our CAPS model, we utilized a subset of the TABAD dataset, entailing 17 network sites with KPI data over around 1.5 years. Nevertheless, one crucial difference between CAPS and TABAD is that we can utilize all 37 KPI features for CAPS as a significant benefit of an autoencoder-based approach. The experiment details, such as the number of datapoints used in (un)supervised training and test, are provided in Table 8.

By using all the available KPI features in the dataset, we train an autoencoder model and obtain the encoder component of the model. Then, we retrieve the output of the encoder, which is basically an encoded format of the KPI values of the sites. Then, we proceed to train different classifiers that require labels. Since we know, for each site, which periods of time it suffers from a PIM problem (i.e. before a site visit), we are able to determine the correct label for each duration as either PIM or healthy. Based on the determined labels, along with a subset of the encoded KPI data, we train various classification models. After, we use the remaining subset of the encoded KPI data as the test data and evaluate the performance of each classifier separately.

Table 6 – CAPS performance with best autoencoder configuration (<80,60,40,20,10>).

Classifier	Accuracy	F1 Score	Precision	Recall
RandomForest	0.89 (±0.07)	0.83 (±0.15)	0.86 (±0.12)	0.83 (±0.18)
MLPClassifier	0.76 (±0.13)	0.60 (±0.35)	0.67 (±0.40)	0.65 (±0.33)
NaiveBayes	0.81 (±0.09)	0.74 (±0.17)	0.78 (±0.18)	0.76 (±0.21)
SVM	0.86 (±0.09)	0.71 (±0.32)	0.71 (±0.34)	0.78 (±0.3)

Table 7 – CAPS performance with best classifier (random forest).

Classifier	Accuracy	F1 Score	Precision	Recall
<80,60,40,20,10>	0.89 (±0.07)	0.83 (±0.15)	0.86 (±0.12)	0.83 (±0.18)
<60,40,20,10>	0.89 (±0.05)	0.82 (±0.2)	0.81 (±0.2)	0.84 (±0.2)
<40,20,10>	0.9 (±0.05)	0.83 (±0.2)	0.83 (±0.2)	0.83 (±0.2)

Table 8 – Dataset used in CAPS model training and test phases.

Model Phase	# of Pos. Datapoints (PIM)	# of Neg. Datapoints (No PIM)	Total # of Datapoints
Unsup. Train.	N/A	N/A	6347
Sup. Train.	4236	4264	8500
Testing	647	662	1309

As shown in Table 6, we investigate CAPS performance for changing classification models, random forest, MLP, naive Bayes, and SVM. Results for each classifier are provided in four separate metrics: accuracy, F1 score, precision, and recall. These results are obtained by utilizing the optimum autoencoder configuration (see Table 7). The values given in the table represent the average metric scores across 17 cells whose labels were available whereas the values in the parentheses represent the standard deviation of the scores. As one can observe from the results, the random forest model performs the best in all four metrics, while MLP performs the worst. There is no obvious winner between SVM and naive Bayes, as each of them outperforms the other in a different metric. Our results show that the best performing model, random forest in this case, achieves an 80+% score in every metric, which includes a remarkable 83% F1 score.

4.3.2.2 CAPS hyperparameters

CAPS involves complex components in its pipeline and these components have their own hyperparameters. In our evaluation, we do not explore every single hyperparameter of each component. Instead, we explore the performance by replacing components with their alternatives. We make this hyperparameter search in two dimensions. In the first dimension, we take a look at the performance of CAPS for changing autoencoder complexities. For this, we create autoencoder networks of different layers and neurons. In the second dimension, we inspect the performance of CAPS with different classifiers such as random forest and SVM, which were given in Table 6.

For the autoencoder complexity dimension, Table 7 shows the results of CAPS performance under autoencoders with varying numbers of layers and neurons. For this evaluation, we fixed the classifier to random forest which we have found to be the most successful one in the aforementioned four metrics. We create autoencoders with different numbers of layers and neurons. The notation $< n_1, n_2, , n_k >$ represents an autoencoder with k layers where n_i indicates the number of neurons in the i^{th} layer. For example, <40,20,10> refers to an autoencoder where its encoder consists of three layers with 40, 20, and 10 neurons. Note that the decoder is set as the mirror of the encoder. According to Table 7, we do not observe huge differences in the performance, yet we still find that <80,60,40,20,10> is the most reliable one as standard deviations are smaller. The reason for that is, although the accuracy of <40,20,10> is slightly better compared to <80,60,40,20,10>, the precision is a little worse. Furthermore, the standard deviation of most metrics in <40,20,10> is higher compared to those in <80,60,40,20,10>.

4.3.2.3 TABAD vs CAPS performance

Comparing TABAD, an anomaly detection method, to a semi-supervised technique like CAPS might not be truly fair, given that TABAD is completely unsupervised. However, for the sake of a comprehensive evaluation, we can point out the difference in results for common performance metrics.

F1 score and accuracy metrics are not utilized during the evaluation of TABAD, so we can only consider precision and recall while comparing CAPS to TABAD. From Table 1, we witness that TABAD achieves a recall score of 80% and precision score of 33%, due to a high number of false positives. Even though each classifier under CAPS achieves a higher precision value than TABAD by a significant margin, the recall score of TABAD outperforms CAPS, with the exception of the random forest classifier, which has a recall score of 83%. Our findings indicate

that, although CAPS provides better scores in general, in a scenario where false positives are much less critical than false negatives, TABAD can also be utilized.

5. DISCUSSION

In addition to the experimental evaluation, we discuss TABAD and CAPS in this section to analyze their key characteristics and delineate the main challenges identified during our research.

5.1 TABAD vs CAPS

Our proposed solutions, TABAD and CAPS, operate on different principles. TABAD considers the PIM issue as an anomaly detection problem and analyzes the time-series-based KPI data in a certain window to determine when an anomaly happens. Thus, TABAD does not require a pre-labeled dataset of network KPIs. CAPS, on the other hand, considers the pre-labeled periods of a network site and determines whether a site suffers from PIM for a specific duration of time. In other words, TABAD works in an unsupervised manner, while CAPS is considered a semi-supervised method, performing a classification process as the final result. Such distinction between TABAD and CAPS allows each of them to be useful under certain circumstances. For instance, in the scenario where we have an adequate amount of labeled data, using CAPS might produce healthier results. Meanwhile, for environments in which obtaining labeled data is difficult, an unsupervised technique like TABAD would be more useful.

In addition to the underlying methodology, TABAD and CAPS also differ heavily in their mechanisms to detect PIM issues. TABAD only considers a single KPI of a network site to determine whether PIM exists in the site. On the contrary, CAPS is based on autoencoders, a deep learning model suited for data with high dimensions; therefore, using various KPIs is possible with CAPS. In a setup where we have fewer KPIs regarding target network sites, TABAD can be considered; however, if we can acquire plenty of different KPIs, using CAPS might be a good option for exploiting all the available data.

Another difference between the two approaches is that TABAD is designed to be site specific; thus, for each target network site, a unique TABAD model needs to be instantiated. This kind of restriction is not applied to CAPS. However, given that a model based on CAPS can be used for multiple sites, assuming its parameters are configured in a reasonable way. Another feasible approach is to generate a number of prototype models, each of which is targeted for sites with similar conditions, based on conditions such as weather, geography, and traffic, and given the specifics of the target network site, the best suited model can be used for PIM detection in that site.

5.2 Challenges

PIM analysis and detection via remote and data analytics-based techniques is a multifaceted research problem. In our investigation, we have run into various challenges regarding the data collection and analysis aspects for PIM detection, which we highlight in this section:

– Data collection: Due to the nature of the PIM problem, it is difficult to obtain a 100% reliable result on whether a site suffers from PIM or not. For the most accurate and healthy conclusion, load tests should be executed per site. Thus, running these tests on every site is not always possible and when a site is tested, the result is usually not positive, since PIM is a rare anomaly in a network.

– Bias in the dataset: As part of the previously explained challenge, the number of healthy sites in the collected datasets significantly outweighs the number of faulty sites with the PIM problem, resulting in an imbalanced dataset with a strong bias for healthy sites. In an attempt to overcome this problem to some extent, we have exploited the expert-provided knowledge for labeling additional sites based on their KPI, without having to conduct load tests physically.

– Selection of right performance metrics: Finding a suitable metric for evaluating our solutions has also been a challenge. Although other metrics, e.g., F1 score, may be better for quantifying the detection performance, we have adopted precision and recall since they are widely used in the literature and more convenient for comparison with existing works.

– Data confidentiality: Some network vendors might be concerned with sharing their network site data, concerned with the possibility of a compromise of the confidential information in their system. For such cases, we alter the specific parts of the data, such as changing the site name and randomizing the site location coordinates, while keeping the KPI values intact to avoid misleading training and testing phases. Still, modifying the original data might cause side effects during the ML process, especially considering our diversity challenge, which we have previously mentioned.

6. CONCLUSION

In this work, we have proposed two different approaches for PIM detection: TABAD and CAPS. While both mod-

els operate on real-time KPI data, TABAD is an anomaly detection approach and CAPS is a semi-supervised classifier based on autoencoders. Both proposed models have outperformed their respective benchmark methods in our experiments.

In the future, we are planning to improve the precision and recall metrics of our methods by switching to a more sophisticated approach like the Kalman filter for TABAD and employing different ML models such as XGBoost for CAPS. Furthermore, we would like to design a predictive PIM detection model that can estimate the occurrence of PIM beforehand, granting network operators the capability to take proactive measures.

ACKNOWLEDGEMENT

A preliminary version of this work was presented at the Sixth International Balkan Conference on Communications and Networking (BalkanCom 2023) - DOI: 10.1109/BalkanCom58402.2023.10167980.

REFERENCES

[1] Anja Feldmann, Oliver Gasser, Franziska Lichtblau, Enric Pujol, Ingmar Poese, Christoph Dietzel, Daniel Wagner, Matthias Wichtlhuber, Juan Tapiador, Narseo Vallina-Rodriguez, Oliver Hohlfeld, and Georgios Smaragdakis. "A Year in Lockdown: How the Waves of COVID-19 Impact Internet Traffic". In: *Commun. ACM* 64.7 (June 2021), pp. 101–108. ISSN: 0001-0782. DOI: 10.1145/3465212. URL: https://doi.org/10.1145/3465212.

[2] Gokcan Cantali, Eren Deniz, Ozcan Ozay, Onur Yildirim, Gürkan Gür, and Fatih Alagoz. "PIM Detection in Wireless Networks as an Anomaly Detection Problem". In: *2023 International Balkan Conference on Communications and Networking (BalkanCom)*. 2023, pp. 1–6. DOI: 10.1109/BalkanCom58402.2023.10167980.

[3] Zhanghua Cai, Lie Liu, Francesco de Paulis, and Yihong Qi. "Passive Intermodulation Measurement: Challenges and Solutions". In: *Engineering* 14 (2022), pp. 181–191. ISSN: 2095-8099. DOI: https://doi.org/10.1016/j.eng.2022.02.012. URL: https://www.sciencedirect.com/science/article/pii/S2095809922002235.

[4] Eren Deniz, Gokcan Cantali, Ozcan Ozay, Onur Yildirim, Gürkan Gür, and Fatih Alagoz. "Towards Data Analytics Based PIM Detection in Wireless Networks". In: *2023 IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks IEEE CAMAD 2023*. 2023.

[5] YUPANA Inc. *Passive Inter-modulation Sources and Cancellation Methods*. https://yupanatech.com/media/uploads/PIM_YUPANA.pdf. YUPANA White Paper. 2022.

[6] Hakan Murat Karaca. "Passive Inter-modulation Sources and Cancellation Methods". In: *The European Journal of Research and Development* 2.2 (June 2022), pp. 75–91. DOI: 10.56038/ejrnd.v2i2.30. URL: https://journals.orclever.com/ejrnd/article/view/30.

[7] Gopinath Muruti, Fiza Abdul Rahim, and Zul-Azri bin Ibrahim. "A Survey on Anomalies Detection Techniques and Measurement Methods". In: *2018 IEEE Conference on Application, Information and Network Security (AINS)*. 2018, pp. 81–86. DOI: 10.1109/AINS.2018.8631436.

[8] Akanksha Toshniwal, Kavi Mahesh, and R. Jayashree. "Overview of Anomaly Detection techniques in Machine Learning". In: *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. 2020, pp. 808–815. DOI: 10.1109/I-SMAC49090.2020.9243329.

[9] Song Wang, Juan Fernando Balarezo Serrano, Kandeepan Sithamparanathan, Akram Al-Hourani, Karina Gomez Chavez, and Ben Rubinstein. "Machine Learning in Network Anomaly Detection: A Survey". In: *IEEE Access* PP (Nov. 2021), pp. 1–1. DOI: 10.1109/ACCESS.2021.3126834.

[10] Chen Chen and Yangyang Gu. "A Mechanical Modelling and Simulation Method for Resolving PIM Problems in Antennas". In: *Sensors* 22.1 (2022). ISSN: 1424-8220. DOI: 10.3390/s22010294. URL: https://www.mdpi.com/1424-8220/22/1/294.

[11] Dongwei Wu, Yongjun Xie, Yong Kuang, and Liqiang Niu. "Prediction of Passive Intermodulation on Mesh Reflector Antenna Using Collaborative Simulation: Multiscale Equivalent Method and Nonlinear Model". In: *IEEE Transactions on Antennas and Propagation* 66.3 (2018), pp. 1516–1521. DOI: 10.1109/TAP.2017.2786304.

[12] Huiping Yang, Wei Huang, He Wen, and Xiu Yin Zhang. "Improved Design of Flange Mount Coaxial Connector With Low Passive Intermodulation Distortion". In: *IEEE Transactions on Instrumentation and Measurement* 71 (2022), pp. 1–7. DOI: 10.1109/TIM.2022.3150893.

[13] Wenbo Wang, Yimin Wang, Wenxuan Zang, Huali Duan, Wenchao Chen, and Wei E.I. Sha. "Physical Mechanisms of Passive Intermodulation: A Short Review". In: *2022 International Applied Computational Electromagnetics Society Symposium (ACES-China)*. 2022, pp. 1–3. DOI: 10.1109/ACES-China56081.2022.10064913.

[14] Kaelus. *Range-to-Fault (RTF)*. https://www.kaelus.com/en/test-measurement-solutions/portable-pim-testing/range-to-fault-(rtf)-en.

[15] Anritsu. *Understanding PIM*. https://www.anritsu.com/en-us/test-measurement/solutions/en-us/Understanding-PIM.

[16] Alexey Shitvov, Alexander Schuchinsky, and Dmitry Zelenchuk. "Near-field mapping of passive intermodulation in printed circuits". In: *Proceedings of the Fourth European Conference on Antennas and Propagation*. 2010, pp. 1–4.

[17] Sen Yang, Wei Wu, Shuai Xu, Yao-Jiang Zhang, Daniel Stutts, and David J. Pommerenke. "A Passive Intermodulation Source Identification Measurement System Using a Vibration Modulation Method". In: *IEEE Transactions on Electromagnetic Compatibility* 59.6 (2017), pp. 1677–1684. DOI: 10.1109/TEMC.2017.2705114.

[18] Xinjie Li, Deshuang Zhao, Yiling Wang, Jianhua Deng, Dejin Yan, and Bing-Zhong Wang. "Detection of Passive Intermodulation by Super-Resolution Imaging of Subwavelength Structures". In: *2022 International Conference on Microwave and Millimeter Wave Technology (ICMMT)*. 2022, pp. 1–3. DOI: 10.1109/ICMMT55580.2022.10022673.

[19] Chengkai Zhu, Zhenyu Chen, Bin Zhang, Xinbo Wang, Wanzhao Cui, Yun Li, Zhongbo Zhu, Xiaojun Li, Jun Wang, Song Wu, and Lixin Ran. "Testing of Passive Intermodulation Based on an Ultrawideband Dual-Carrier Nulling". In: *IEEE Transactions on Microwave Theory and Techniques* 70.8 (2022), pp. 4017–4025. DOI: 10.1109/TMTT.2022.3186362.

[20] Stephen Moss, Elanchezhian Veeramani, and Joris Angelo Sundaram Jerome. "Passive Intermodulation (PIM) Test and Measurement". In: *2020 IEEE 29th North Atlantic Test Workshop (NATW)*. 2020, pp. 1–3. DOI: 10.1109/NATW49237.2020.9153075.

[21] D. Allal, R. Bannister, K. Buisman, D. Capriglione, G. Di Capua, M. García-Patrón, T. Gatzweiler, F. Gellersen, T. Harzheim, H. Heuermann, J. Hoffmann, A. Izbrodin, K. Kuhlmann, K. Lahbacha, A. Maffucci, G. Miele, F. Mubarak, M. Salter, T.D. Pham, A. Sayegh, D. Singh, F. Stein, and M. Zeier. "RF Measurements for Future Communication Applications: an Overview". In: *2022 IEEE International Symposium on Measurements & Networking (M&N)*. 2022, pp. 1–6. DOI: 10.1109/MN55117.2022.9887740.

[22] Vesa Lampu, Lauri Anttila, Matias Turunen, Marko Fleischer, Jan Hellmann, and Mikko Valkama. "Air-Induced Passive Intermodulation in FDD Networks: Modeling, Cancellation and Measurements". In: *2021 55th Asilomar Conference on Signals, Systems, and Computers*. 2021, pp. 983–988. DOI: 10.1109/IEEECONF53345.2021.9723205.

[23] Vesa Lampu, Lauri Anttila, Matias Turunen, Marko Fleischer, Jan Hellmann, and Mikko Valkama. "Air-Induced PIM Cancellation in FDD MIMO Transceivers". In: *IEEE Microwave and Wireless Components Letters* 32.6 (2022), pp. 780–783. DOI: 10.1109/LMWC.2022.3164718.

[24] Vesa Lampu, Lauri Anttila, Matias Turunen, Marko Fleischer, Jan Hellmann, and Mikko Valkama. "Cancellation of Air-Induced Passive Intermodulation in FDD MIMO Systems: Low-Complexity Cascade Model and Measurements". In: *2023 IEEE/MTT-S International Microwave Symposium - IMS 2023*. 2023, pp. 33–36. DOI: 10.1109/IMS37964.2023.10188046.

[25] Muhammad Zeeshan Waheed, Dani Korpi, Adnan Kiayani, Lauri Anttila, and Mikko Valkama. "Digital Cancellation of Passive Intermodulation: Method, Complexity and Measurements". In: *2019 IEEE MTT-S International Microwave Conference on Hardware and Systems for 5G and Beyond (IMC-5G)*. 2019, pp. 1–3. DOI: 10.1109/IMC-5G47857.2019.9160361.

[26] Muhammad Zeeshan Waheed, Dani Korpi, Lauri Anttila, Adnan Kiayani, Marko Kosunen, Kari Stadius, Pablo Pascual Campo, Matias Turunen, Markus Allén, Jussi Ryynänen, and Mikko Valkama. "Passive Intermodulation in Simultaneous TransmitReceive Systems: Modeling and Digital Cancellation Methods". In: *IEEE Transactions on Microwave Theory and Techniques* 68.9 (2020), pp. 3633–3652. DOI: 10.1109/TMTT.2020.2996206.

[27] Anritsu. *A New Technique for Measuring Passive Intermodulation Over Fiber Optic Cables*. https://dl.cdn-anritsu.com/en-us/test-measurement/files/Technical-Notes/White-Paper/11410-01135A.pdf. AILG White Paper. 2019.

[28] Hakan Evircan, Emir Kursad Ulusoy, and Muzaffer Mete Dalan. *Remote Detection and Analysis of Passive Intermodulation Problems in Radio Base Stations*. US10009784, Jun 26, 2018, https://patentscope.wipo.int/search/en/detail.jsf?docId=US221793731, Accessed on: May 5, 2023.

[29] Freek Jochems and Alexios Balatsoukas-Stimming. "Non-Linear Self-Interference Cancellation via Tensor Completion". In: *arXiv e-prints*, arXiv:2010.01868 (Oct. 2020), arXiv:2010.01868. DOI: 10.48550/arXiv.2010.01868. arXiv: 2010.01868 [eess.SP].

[30] Bizheng Liang, Xiangyuan Bu, Mucheng Li, Pengfei Guo, and Celun Liu. "A Novel RTRLNN Model for Passive Intermodulation Cancellation in Satellite Communications". In: *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. 2018, pp. 18–23. DOI: 10.1109/IWCMC.2018.8450316.

[31] Faris B. Mismar. "Intermodulation Interference Detection in 6G Networks: A Machine Learning Approach". In: *arXiv e-prints*, arXiv:2111.00524 (Oct. 2021), arXiv:2111.00524. DOI: 10.48550/arXiv.2111.00524. arXiv: 2111.00524 [cs.NI].

[32] Serene Banerjee, Raul R Martin, and Abel Pardo. "Frequency-aware Time Series Forecasting, Anomaly Detection, Classification and Granger Causality". In: *2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS)*. 2022, pp. 217–221. DOI: 10.1109/COMSNETS53615.2022.9668359.

[33] H.G. Ranjani, Sleeba P Puthennurakel, Arthur Brisebois, Serene Banerjee, and Venkatesh Umaashankar. "Time series based approach for detecting Passive Intermodulation occurrences in cellular network". In: *2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS)*. 2022, pp. 630–638. DOI: 10.1109/COMSNETS53615.2022.9668382.

[34] Serene Banerjee, Pratyush Kiran Uppuluri, Rahul Sharma N, and Subhadip Bandyopadhyay. "Self-Supervised Reinforcement Learning for Proactive Prediction of Passive Intermodulation". In: *2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS)*. 2023, pp. 309–313. DOI: 10.1109/COMSNETS56262.2023.10041311.

[35] Juergen Schmidhuber. *Annotated History of Modern AI and Deep Learning*. 2022. arXiv: 2212.11279 [cs.NE].

[36] M.V. Valueva, N.N. Nagornov, P.A. Lyakhov, G.V. Valuev, and N.I. Chervyakov. "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation". In: *Mathematics and Computers in Simulation* 177 (2020), pp. 232–243. ISSN: 0378-4754. DOI: https://doi.org/10.1016/j.matcom.2020.04.031. URL: https://www.sciencedirect.com/science/article/pii/S0378475420301580.

[37] Mohammadreza Nazari, Afshin Oroojlooy, Lawrence V. Snyder, and Martin Taká. *Reinforcement Learning for Solving the Vehicle Routing Problem*. 2018. arXiv: 1802.04240 [cs.AI].

[38] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning Internal Representations by Error Propagation". In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362. ISBN: 026268053X.

[39] Dana H. Ballard. "Modular Learning in Neural Networks". In: *Proceedings of the Sixth National Conference on Artificial Intelligence - Volume 1*. AAAI'87. Seattle, Washington: AAAI Press, 1987, pp. 279–284. ISBN: 0934613427.

[40] G. E. Hinton and R. R. Salakhutdinov. "Reducing the Dimensionality of Data with Neural Networks". In: *Science* 313.5786 (2006), pp. 504–507. DOI: 10.1126/science.1127647. eprint: https://www.science.org/doi/pdf/10.1126/science.1127647. URL: https://www.science.org/doi/abs/10.1126/science.1127647.

[41] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. *Lossy Image Compression with Compressive Autoencoders*. 2017. arXiv: 1703.00395 [stat.ML].

[42] Ahmed Dawoud, Seyed Shahristani, and Chun Raun. "Deep Learning for Network Anomalies Detection". In: *2018 International Conference on Machine Learning and Data Engineering (iCMLDE)*. 2018, pp. 149–153. DOI: 10.1109/iCMLDE.2018.00035.

[43] Sanjoy Basak, Sreeraj Rajendran, Sofie Pollin, and Bart Scheers. "Autoencoder based framework for drone RF signal classification and novelty detection". In: *2023 25th International Conference on Advanced Communication Technology (ICACT)*. 2023, pp. 218–225. DOI: 10.23919/ICACT56868.2023.10079363.

[44] Md Rakibul Ahasan, Mirza Sanita Haque, Mohammad Rubbyat Akram, Mohammed Fahim Momen, and Md Golam Rabiul Alam. "Deep Learning Autoencoder based Anomaly Detection Model on 4G Network Performance Data". In: *2022 IEEE World AI IoT Congress (AIIoT)*. 2022, pp. 232–237. DOI: 10.1109/AIIoT54504.2022.9817338.

[45] Mohammad A. Salahuddin, Vahid Pourahmadi, Hyame Assem Alameddine, Md. Faizul Bari, and Raouf Boutaba. "Chronos: DDoS Attack Detection Using Time-Based Autoencoder". In: *IEEE Transactions on Network and Service Management* 19.1 (2022), pp. 627–641. DOI: 10.1109/TNSM.2021.3088326.

[46] Timotej Smolen and Lenka Benova. "Comparing Autoencoder and Isolation Forest in Network Anomaly Detection". In: *2023 33rd Conference of Open Innovations Association (FRUCT)*. 2023, pp. 276–282. DOI: 10.23919/FRUCT58615.2023.10143005.

[47] Hanan Hindy, Robert Atkinson, Christos Tachtatzis, Jean-Noel Colin, Ethan Bayne, and Xavier Bellekens. "Utilising Deep Learning Techniques for Effective Zero-Day Attack Detection". In: *Electronics* 9 (Oct. 2020), p. 1684. DOI: 10.3390/electronics9101684.

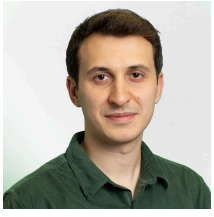
[48] Zahra Zamanzadeh Draban, Geoffrey Webb, Shirui Pan, Charu Aggarwal, and Mahsa Salehi. "Deep Learning for Time Series Anomaly Detection: A Survey". In: (Nov. 2022). DOI: 10.48550/arXiv.2211.05244.

[49] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation Forest". In: *International Conference on Data Mining*. 2008, pp. 413–422. DOI: 10.1109/ICDM.2008.17.

AUTHORS

Gokcan Cantali is a research assistant at Zurich University of Applied Sciences (ZHAW), Institute of Computer Science (InIT). He earned his bachelor's degree in 2016 and received his master's degree in 2019, both from the Computer Engineering department in Bogazici University, Istanbul. He also has over eight years of software development experience, and has taken the role of tech lead for about 2.5 years in the past. His research interests include network security and privacy, network performance optimization, and AI-based software applications.

Eren Deniz earned an B.S. degree in 2000 and an M.S. degree in 2005 in electronics engineering from Dokuz Eylul University in Izmir, Turkey. With over 23 years of experience in telecommunications, he has worked for various telcom companies, taking on technical and managerial responsibilities. Currently, he is a Ph.D. candidate in computer engineering at Ege University in Izmir, Turkey. In addition to his academic pursuits, he serves as a researcher at Yupana Inc. His research interests include machine learning, natural language processing, and knowledge graphs.



Hasan Ferit Eniser is a Ph.D. student at MPI-SWS. He graduated from Bogazici University in 2018 with an M.Sc. in computer science. Before that, he did his B.Sc. studies in computer science at the same university. Broadly, he works in the intersection of software testing/verification and artificial intelligence. He develops new testing and verification techniques to ensure the dependability of AI-enabled systems. He has authored more than 10 academic works.

Onur Yildirim is a software solutions director, with over ten years of experience on developing innovative software solutions. He earned his bachelor's degree in the Electronics and Communications department of Istanbul Technical University, in 2000. He has a total of 22 years of experience in the telecommunication industry, in which he has worked in various network and telecom companies in different countries.

Gürkan Gür is a senior lecturer at Zurich University of Applied Sciences (ZHAW), Institute of Computer Science (InIT) in Winterthur, Switzerland. He received his B.S. degree in electrical engineering in 2001 and Ph.D. degree in computer engineering in 2013 from Bogazici University in Istanbul, Turkey. His research interests include future Internet, 5G and beyond networks, information security, and information-centric networking. He has two patents (one in US, one in TR) and published more than 80 academic works. He is a senior member of IEEE and a member of ACM.

Fatih Alagoz is a professor in the Computer Engineering Department of Bogazici University, Turkey. He obtained his BSc degree in electrical engineering in 1992 from Middle East Technical University, Turkey, and DSc degree in electrical engineering in 2000 from George Washington University, USA. His current research areas include cognitive radios, wireless networks, network security and UWB communications.