IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 3, 2020, accepted November 16, 2020, date of publication November 19, 2020,
date of current version December 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3039333

Learning to Detect Anomalous Wireless Links in IoT Networks

GREGOR CERAR[1,2], (Graduate Student Member, IEEE), HALIL YETGIN[1,3], (Member, IEEE), BLAZ BERTALANIC[1,4], (Member, IEEE), AND CAROLINA FORTUNA[1]

[1]Department of Communication Systems, Jožef Stefan Institute, SI-1000 Ljubljana, Slovenia
[2]Jožef Stefan International Postgraduate School, SI-1000 Ljubljana, Slovenia
[3]Department of Electrical and Electronics Engineering, Bitlis Eren University, 13000 Bitlis, Turkey
[4]Faculty of Electrical Engineering, University of Ljubljana, 1000 Ljubljana, Slovenia

Corresponding author: Halil Yetgin (halil.yetgin@ijs.si)

This work was supported by the Slovenian Research Agency under Grant P2-0016 and Grant J2-9232.

ABSTRACT After decades of research, Internet of Things (IoT) is finally permeating real-life and helps improve the efficiency of infrastructures and processes as well as our health. As massive number of IoT devices are deployed, they naturally incurs great operational costs to ensure intended operations. To effectively handle such intended operations in massive IoT networks, automatic detection of malfunctioning, namely anomaly detection, becomes a critical but challenging task. In this paper, motivated by a real-world experimental IoT deployment, we introduce four types of wireless network anomalies that are identified at the link layer. We study the performance of threshold- and machine learning (ML)-based classifiers to automatically detect these anomalies. We examine the relative performance of three supervised and three unsupervised ML techniques on both non-encoded and encoded (autoencoder) feature representations. Our results demonstrate that; i) selected supervised approaches are able to detect anomalies with F1 scores of above 0.98, while unsupervised ones are also capable of detecting the said anomalies with F1 scores of, on average, 0.90, and ii) OC-SVM outperforms all the other unsupervised ML approaches reaching at F1 scores of 0.99 for SuddenD, 0.95 for SuddenR, 0.93 for InstaD and 0.95 for SlowD.

INDEX TERMS Anomaly detection, Internet of Things (IoT), machine learning (ML), wireless links, wireless networks.

I. INTRODUCTION

The Internet of Things (IoT) has received a plethora of attention from both industry and academia due to the market release of a variety of smart devices on a regular basis, e.g. the devices retrofitted in home appliances, wearables, healthcare, vehicles and industrial machinery, just to name a few [1]. To this end, extensive research efforts have been put forward for their active deployment and development to enable increasingly efficient and more automated operations in manufacturing, agriculture, transportation and healthcare, but also due to their massive economic contributions [2].

Valid business cases [3] and successful real-world large-scale IoT deployments are emerging as a way to improve existing business processes as well as enable new applications [2]. However, once the network of sensors is deployed, it becomes part of the operational infrastructure of a business, and needs to be maintained and serviced similar to any other infrastructure, such as legacy IT infrastructure, robots and machines just to name a few. Minimizing maintenance costs while ensuring the reliability of IoT network [4] becomes prohibitive when the number of sensors are in their thousands or tens of thousands. To efficiently manage such massive IoT networks, automatic IoT network monitoring [5] and malfunction detection [6] solutions that automatically report relevant malfunctions and filter them out without influencing the business process are required.

IoT network or node malfunctioning can also be referred to as network or node anomaly and to date, it has been defined in various ways, often from the perspective of monitored networking aspects. For instance, Sheth *et al.* [6] define and identify anomalies from the IEEE 802.11 physical layer perspective, namely, hidden terminal, capture effect, noise and signal strength variation anomalies, whereas Gupta *et al.* [7] define anomalies from multihop networking perspective with the aspects, such as black hole, sink hole, selective forwarding

The associate editor coordinating the review of this manuscript and approving it for publication was Celimuge Wu.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

and flooding. Alipour *et al.* [8] define the anomalies from IEEE 802.11 link layer security perspective with the focus on aspects, such as injection test, deauthentication attack, disassociation attack, association flood and authentication flood. Generally speaking, anomaly detection research in IoT networks can be found in the form of intrusion, fraud and fault detection, system health monitoring, event detection in sensor networks and detecting ecosystem disturbances [9], where most studies mainly concerned with a certain type of anomaly within a specific scenario.

In this paper, motivated by a real-world experimental IoT deployment, we define four types of IoT anomalies that can be identified at the link layer, namely *sudden degradation*, *sudden degradation with recovery*, *instantaneous degradation* and *slow degradation*. Rather than focusing on the cause of an anomaly as realized in [6] and [7], we focus our attention on the observable symptoms of link measurements, namely the changes in the expected received signal. Based on the type of anomaly, we identify possible root causes that may be related to hardware, firmware and the channel, and develop models for automatically classifying the introduced anomalies. By accurately detecting these four types of anomalies, a wireless network operator is able to quickly and proactively detect issues within the operation of the network without waiting to be explicitly alerted by users. Proactively detecting and mitigating malfunctions can increase user satisfaction, reduce churn and ultimately show significant improvements in business KPIs. Additionally, the detected and classified anomaly type can aid technical staff with the well-informed decisions so as to diagnose and resolve the issues. For instance, *sudden degradation with recovery* is observed frequently after updating the firmware of devices in the network, which is highly likely related to the bugs of the firmware that prevent devices from working as intended and trigger the watchdog to reset. Therefore, discriminating between four of those types of anomalies and automatizing this process can speed up the real-time resolution of the network-related issues, in turn diminishing the allotted personnel and their efforts, and network-wide operational costs of mobile operators. The major contributions of this paper are as follows.

1) We define four types of anomalies that can appear on wireless links and are representative for narrowing down the causes and enabling more efficient mitigation. Driven by a real-world operational wireless infrastructure, for each of the defined anomalies we identify their symptoms from the application perspective and potential underlying causes.
2) We study the performance of standard manually-engineered features and a proposed autoencoder-based automatic feature generation approach, and show the performance improvement brought by the latter.
3) We also analyse the relative performance of three supervised and three unsupervised ML techniques. More explicitly, we consider regression-based, tree-based and kernel-based methods as part of our supervised techniques, while nearest neighbours, tree- and kernel-based methods are leveraged as their unsupervised counterpart techniques.

Additionally, minor contributions are outlined as follows:

1) Based on the gained knowledge while operating the LOG-a-TEC wireless experimentation testbed [10], we provide an analysis on real-world operational measurements that further stresses the need for automated anomaly detection in massive IoT networks.
2) We produce a publicly available anomaly detection tool-set[1] including entire procedures, e.g., anomaly injection into trace-sets, feature generation out of data representations, and model training and development.

This paper is structured as follows. Section II summarizes the related work and Section III presents an analysis of the real-world testbed measurements motivating our contributions, while Section IV introduces the four types of IoT network anomalies. Then, Section V elaborates on various data representations that can be used to generate features for training the proposed ML models, whereas Section VI discusses the threshold-based approach as well as the selected supervised and unsupervised ML techniques. Section VII describes the relevant methodological and experimental details, while Section VIII provides thorough analyses of the results and discusses the limitations. Finally, Section IX concludes the paper.

II. RELATED WORK

We provide related work to the main contributions of this paper as follows. First, we discuss related works that define anomalies in wireless and IoT networks, then we stress on the use of autoencoders for improving various aspects of wireless networks including anomaly detection, and finally, we focus on ML models that support for improved operations of wireless networks.

A. ANOMALY DEFINITIONS IN WIRELESS NETWORKS

Generally speaking, *an anomaly* is defined as an outlier, a distant object, an exception, a surprise, an aberration or a peculiarity, depending on the domain, research community and specific application scenario [9], [11]–[15]. A widely used classification of anomalies, including in wireless sensor network research is provided in [9], [16], where three classes of anomalies are defined based on their nature; point anomalies, contextual anomalies and collective anomalies. In [14], Gupta *et al.* classify relevant studies on outlier detection for time series data, one of which is the point outlier as defined in [9], and others are subsequence outliers, global and local outliers. More recently, Lavin and Ahmad *et al.* [17] introduce a benchmark for anomaly detection, and target mainly at cloud networks and associated services, where they provide reference datasets to be used when evaluating the performance of anomaly detection algorithms. While they do

[1]Script for the design and development of anomaly detection models: https://gist.github.com/gcerar/0b03e55f41147a7b7230f45d1f1209d6

not specifically define the type of anomalies, their benchmark datasets include several anomalies.

Due to the spatio-temporal nature of wireless sensor network monitoring and data collection, Jurdak *et al.* [18] introduce temporal, spatial and spatio-temporal anomalies as well as node, network and data anomalies, followed by even finer grained anomalies, such as node resets, node failures, etc. A number of studies then introduce more focused and application specific anomalies. For instance, Sheth *et al.* [6] define and identify anomalies from the IEEE 802.11 physical layer perspective namely; hidden terminal, capture effect, noise and signal strength variation anomalies. Moreover, Gupta *et al.* [7] define anomalies with the aspects of multihop networking, such as black hole, sink hole, selective forwarding and flooding, whereas Alipour *et al.* [8] define anomalies from IEEE 802.11 link layer security aspects, such as injection test, deauthentication attack, disassociation attack, association flood and authentication flood. For further details, motivated readers are referred to [18] for the diagnosis and detection of wireless network anomalies.

B. AUTOENCODERS FOR IMPROVING WIRELESS NETWORK OPERATIONS AND ANOMALY DETECTION

With the advent of deep learning, one class of techniques belonging to this class of ML, referred to as autoencoders, has been proven to be particularly useful at performing automatic feature engineering also for time series data [19]. Autoencoders attempts to learn a lossless compression of the data and the code resulting from that compression represents a superior feature set.

Generally in wireless, autoencoders have been successfully applied by [20] and their subsequent works, such as [21] to accurately reconstruct physical layer signals and [22] signal denoising for more accurate localization. For anomaly detection in wireless and IoT networks, Wang *et al.* [23] proposed autoencoders for more accurate identification of faulty parts of WSNs, as well as faulty antennas in antenna arrays, whereas Shahid *et al.* [24] and Chen *et al.* [25] proposed autoencoders for identifying anomalies in wireless and IoT networks based on transport layer traces, and recently, Yin *et al.* [26] proposed recurrent autoencoders for time series anomaly detection for IoT networks. However, they used a synthetic dataset with metrics derived from several Yahoo services. Unlike the state-of-the-art, this work proposes autoencoders as an automatic feature generation method for link layer anomaly detection and uses a real-world wireless dataset in which the introduced four types of anomalies are synthetically injected.

C. ML TECHNIQUES FOR WIRELESS AND IoT NETWORK ANOMALY DETECTION

In the literature, it is often a good practice that when a ML solution to a specific problem is considered, several counterpart ML models are evaluated against each other for performance analyses. For instance, Kieu *et al.* [19] compare the performance of ten different ML techniques, such as Support Vector Machines, Local Outlier Factor, Isolation Forest, just to name a few, on six different datasets that are suitable for anomaly detection.

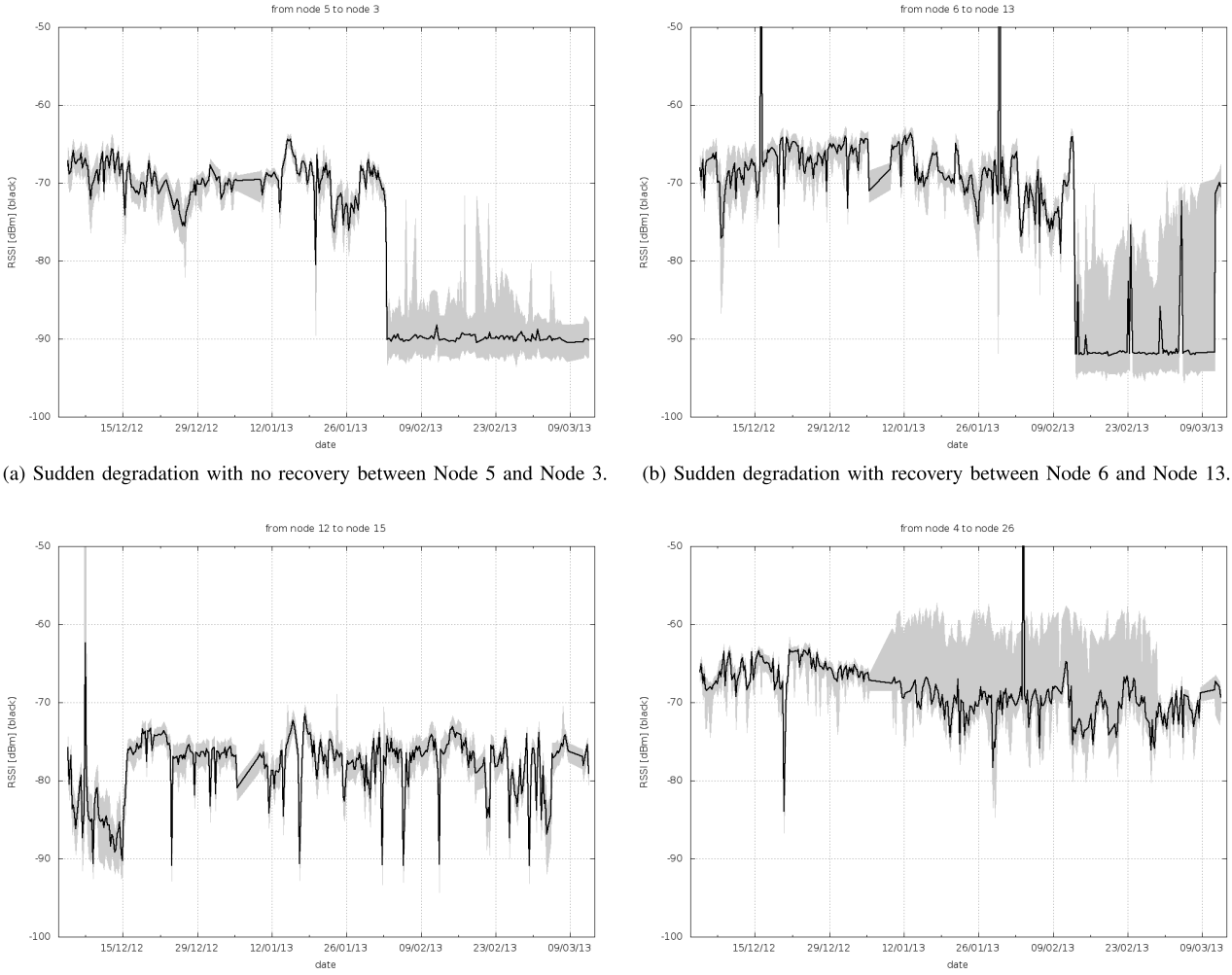
With respect to wireless and IoT network anomalies, Thing [27] evaluate the relative performance of four deep learning and one decision tree models for anomaly detection and attack classification in IEEE 802.11 networks, whereas Chen *et al.* [25] evaluate the relative performance of principal component analysis, standard and convolutional autoencoder for detecting anomalies in transport layer traces, i.e., TCP, UDP and ICMP of wireless networks. Moreover, Ran *et al.* [28] evaluate the relative performance of their proposed semi-supervised approach of IEEE.802.11 anomaly detection, and similarly Salem *et al.* [29] evaluate the relative performance of five ML techniques, i.e., SVM, decision trees (J48), logistic regression, Naïve Bayes, and Decision Table for anomaly detection in WSNs. Additionally, the previous authors [30] also evaluate the performance of their proposed algorithm against selected three ML techniques, namely linear regression, additive regression, and J48 decision tree for anomaly detection in WSNs. However, in most of the ML-based network anomaly detection research discussed in this section as well as in [31] provide only limited relative performance evaluation results. To the best of our knowledge, this paper is the first attempt to provide relative comparisons between three supervised and three unsupervised ML techniques based on various data representations and their encoded counterpart features.

III. MOTIVATION

Our lab runs the LOG-a-TEC [2] testbed that has empowered wireless experimentation for more than ten years. The first version of the testbed comprised of our custom embedded platform [32] was mounted on public light poles in a small municipality of Slovenia [33]. It included more than fifty nodes, most of which were situated in hard-to-reach locations. A sensor management system [10] is used to keep the record of each node for its hardware and software versions, configurations, and locations. This system also performs a number of management and diagnosis related tasks to monitor the operation of the devices.

Over time, the users of the testbed had difficulties in reaching some of the nodes or noticed unexplainable measurements collected during their testbed experimentation. For instance, the transceivers on some of the nodes were degraded significantly for their receiver sensitivity and transmit power performances, and in some cases to such a degree that they became inoperative. As depicted in Figure 1a, third node (ID-3) sensed transmissions from fifth node with received signal strength indicator (RSSI) of about -70 [dBm] on average till 2nd February of 2013. Following that, either fifth node's transmit power or third node's receiver sensitivity was degraded significantly, which was reduced to about -90 [dBm] on average. After investing a good amount of time

[2]LOG-a-TEC testbed with sensor platforms `http://log-a-tec.eu`



(a) Sudden degradation with no recovery between Node 5 and Node 3.

(b) Sudden degradation with recovery between Node 6 and Node 13.

(c) Spike-like instantaneous degradation between Node 13 and Node 15.

(d) Slow degradation between Node 4 and Node 26.

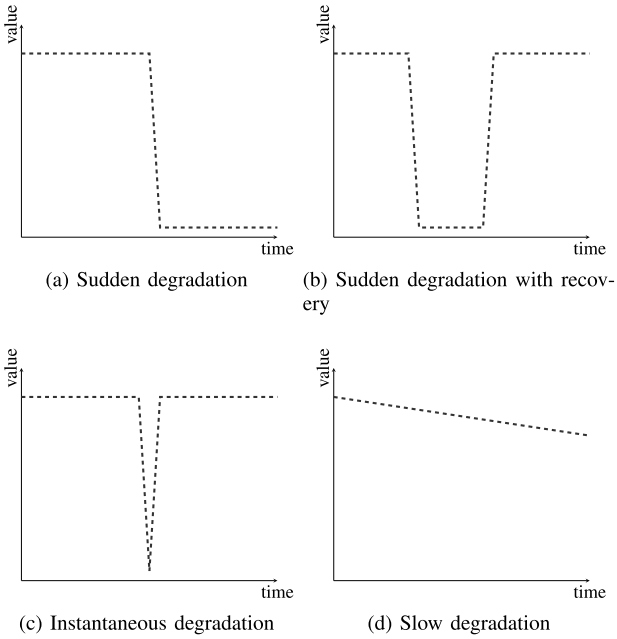
FIGURE 1. Anomalies observed in operational environment, where solid black lines represent average RSSI and greyed areas show maximum/minimum values.

and effort in understanding and reproducing the anomaly, the fifth node was diagnosed with a hardware failure, and it could only be restored to normal operation by replacing the integrated circuit for transceiver (TI CC2500).

Similarly, another anomaly type is experienced in Figure 1b with a sudden degradation and there were several recovery attempts between February 15th and March 9th 2013. In this particular case, we figured out that the sixth node was accidentally downgraded in February to an older version of the firmware that had a bug in the spectrum sensing code, which directly affected the operations of the sixth node and degraded its transmit power. Figure 1c presents several spike-like instantaneous degradation anomalies between nodes 12 and 15. We were not able to discover anything technically wrong with these respective nodes. Therefore, we assumed that these anomalies were probably due to weather and/or large objects moving around the radios, since these two devices were mounted in an industrial zone, where moving large trucks and massive long-term standing objects were not an uncommon occurrence, which can indeed incur spikes due to the instantaneous non-line-of-sight channels experienced. Finally, Figure 1d also exhibits two distinguishable rapid drops and climbs, but most importantly, on average, shows a slightly degrading performance in sensitivity and/or transmit power between nodes 4 and 26 after December 2012. We were not able to readily justify such behaviour of the device, but ageing of electronic components may induce such behaviour, which is a well-known issue [34].

IV. WIRELESS NETWORK ANOMALIES

Wireless networks are designed to exchange data between two communicating parties, e.g., video, voice and sensor measurements. As long as the network remains functional and is not interrupted, all the devices within the network are considered ordinarily operable. When the devices are compromised as exemplified in Section III, then a degradation in the service quality is experienced. The way how anomalies affect the user's service quality experience is stringently



(a) Sudden degradation (b) Sudden degradation with recovery

(c) Instantaneous degradation (d) Slow degradation

FIGURE 2. Visual representation of anomalies abbreviated as; a) SuddenD, b) SuddenR, c) InstaD, d) SlowD.

associated with the type of anomaly. Therefore, in this section, we introduce four types of anomalies that can be observed in communication links of wireless networks, which were mainly discovered in our evaluation of a real-world experimentation, as discussed in Section III: a) sudden degradation, b) sudden degradation with recovery, c) spike-like instantaneous degradation and d) slow degradation.

A. SUDDEN DEGRADATION (SuddenD)

The sudden degradation anomaly can be mathematically represented by a step function with decreasing slope, as depicted in Figure 2a. In our case, this represents a sudden persistent change in the state of a link. While this sudden change with an increasing slope is also possible in theory, typically it will only lead to a more reliable link, therefore they are not accounted as an anomaly.

Symptom: From the perspective of a user, services may become unavailable, offline and unreachable. From the perspective of a network, either the transmitter stops generating electromagnetic field or the receiver is unable to receive data.

Possible causes: Such sudden degradation can be induced by a transceiver failure as discussed in Section III and depicted in Figure 1a, a significant and sudden change in the position of one or both of the communicating parties leading them to remain disconnected, moving from line-of-sight to a non-line-of-sight environment with obstacles preserving electromagnetic shielding materials, and a significant hardware or software failure where built-in recovery mechanisms, such was watchdogs cannot be triggered.

B. SUDDEN DEGRADATION WITH RECOVERY (SuddenR)

The sudden degradation with recovery anomaly can be mathematically represented by a step function with decreasing slope, as depicted in Figure 2b. In this case, the state of a link suddenly changes, stays in the new state for a longer period of time and ultimately returns to the previous state. In sudden degradation with recovery, communication is interrupted for a certain period of time.

Symptom: From user's perspective, provided services may become sluggish and unavailable for a certain period of time and later resume back to their regular operations. From the perspective of the network, in the case of sudden degradation with recovery, either transmitter temporarily stops generating electromagnetic field or the receiver temporarily is unable to receive it.

Possible causes: This type of degradation can be caused by buffer congestion and software bug, as discussed in Section III and depicted in Figure 1b, where watchdog performs reboot after a certain timeout, a radio remaining in excessive active state and requiring recalibration, an obstacle blocking the communication for some time, and a signal jammer equipped on a military vehicle that is passing by.

C. INSTANTANEOUS DEGRADATION (InstaD)

The instantaneous degradation anomaly can be mathematically represented by a step function with steeply decreasing slope, forming a sudden spike, as depicted in Figure 2c. In this case, the state of the link changes suddenly, but instantaneously returns to its previous state. The instantaneous degradation anomaly may appear as an information loss.

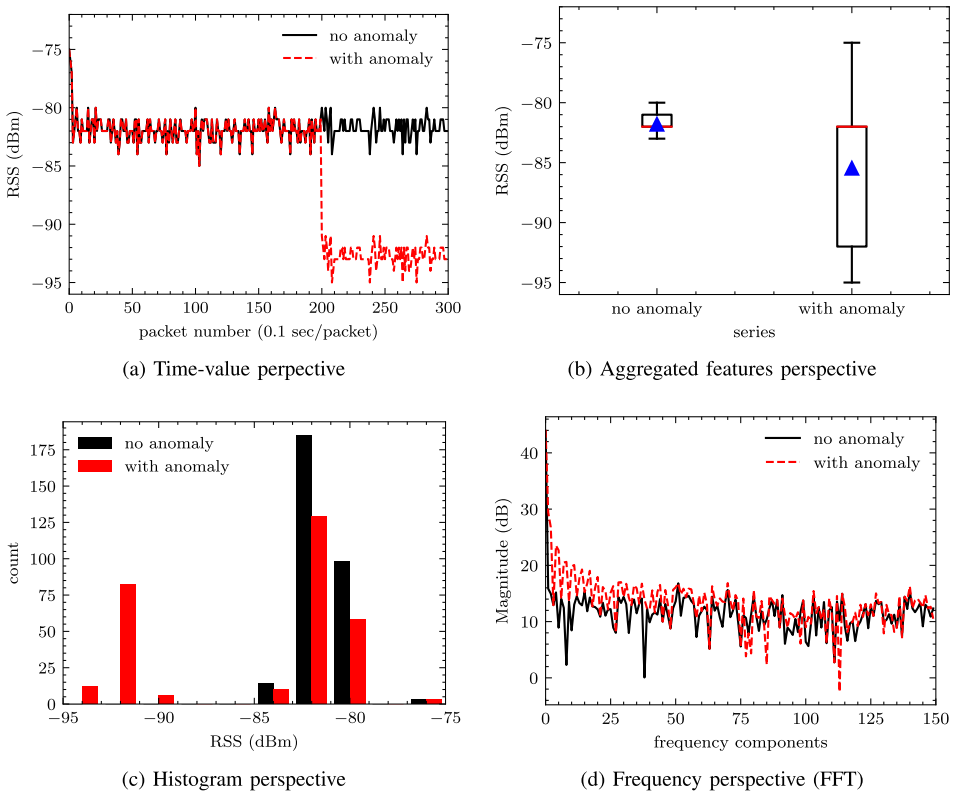
Symptoms: From user's perspective, a real-time service may experience instant lags, while other non-real-time services may work unaffected. From the perspective of the network, either transmitter experiences a deep fading instance or the receiver becomes unable to receive data due to an instant exposure to excessive noise or interference.

Possible causes: This type of degradation can be caused by an instant interference, collision, quantization errors, value reading errors or sudden saturations in the transceiver's electronic components, as discussed in Section III and depicted in Figure 1c, where anomaly can be stringently induced by the issues related to the propagation environment, such as an external device communicating on the same frequency, excessive background noise and multipath fading, just to name a few.

D. SLOW DEGRADATION (SlowD)

The slow degradation anomaly can be mathematically represented as a normalized linear function with slightly decreasing slope, as depicted in Figure 2d. In this case, the state of the link undertakes slight and unnoticeable changes for a longer period of time and it may never resume to its original state. The slow degradation anomaly may commence triggering information loss and interruptions after a certain amount of time.

Symptom: Slow degradation anomaly could go unnoticed for a very long time, where users may not even notice any difference in service quality immediately. When relevant thresholds are triggered, users commence experiencing deteriorated



(a) Time-value perpective (b) Aggregated features perspective

(c) Histogram perspective (d) Frequency perspective (FFT)

FIGURE 3. Distinct representations of the data for sudden degradation anomaly (SuddenD).

service quality. After employed compensation methods are exhausted (e. g., buffers, queues, bandwidth preservation strategies), communication may be interrupted and intended services may become unavailable. From the perspective of the network, either transmitter gradually stops generating sufficient electromagnetic field to satisfy a received signal-to-noise ratio threshold or the receiver is not able detect or collect enough electromagnetic radiation to decode the information, which can also be induced by the aging of electronic components.

Possible causes: This type of degradation may be caused by easier aging of electronic components in extreme working conditions (e. g., high moisture and heat) as it is discussed in Section III and depicted in Figure 1d, where it reflects a gradual but permanent impairment to the hardware or, slowly increasing obstacle such as a building being slowly built or vegetation growing.

V. DATA REPRESENTATION

Sections III and IV provided real-world anomaly examples and formalized wireless link anomalies, respectively. In the following, we provide five distinct ways to represent data that can be used as features while training the machine learning model.

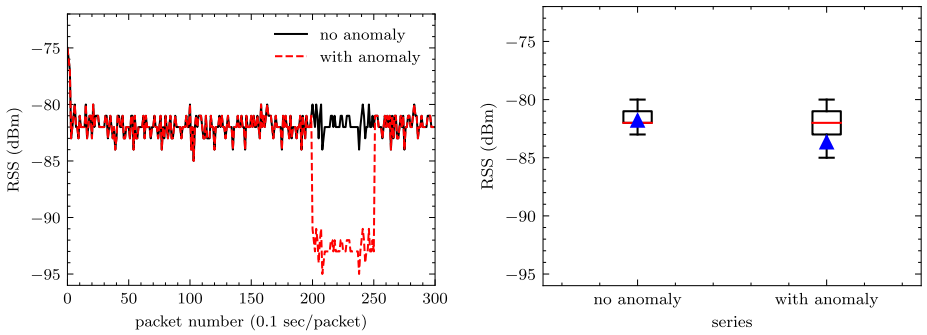
A. TIME-VALUE REPRESENTATION

The anomalies appearing in time series of RSSI values and in Figures 1 and 2 are recorded as raw time-ordered values, thus forming a time series. We refer to this time-ordered values as *time-value* representation. In Figures 3a, 4a, 5a and 6a, the time-value representation of an ordinary link is depicted with solid black lines and its anomaly injected counterpart, as per the definition from Section IV is depicted with dashed red lines.

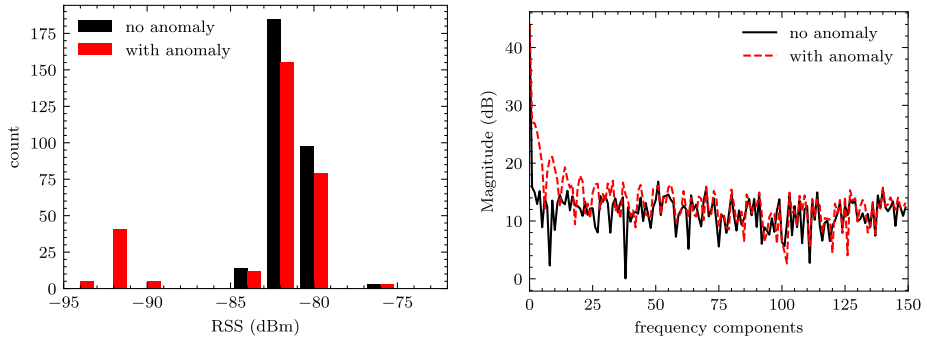
However, through mathematical transformations, time series can be represented in other domains that, in some cases may be more suitable for the analysis of anomaly or pattern recognition. Motivated readers are referred to [35] for a comprehensive taxonomy of time series representation. In addition to the time-value representation, in this study, we also consider an aggregated representation, a histogram representation, a frequency domain representation and an automatically encoded representation.

B. AGGREGATED REPRESENTATION

This representation contains seven statistical aggregates computed from the time-value representation, namely average, standard deviation, and all five quantile (Q) values, such as zeroth quantile (minimum), first quantile, second quantile (median), third quantile, and fourth quantile (maximum). This representation is depicted in Figures 3b, 4b, 5b and 6b for each anomaly type, where they present values belonging to middle quantiles (Q1-Q3) as a box shape, first quantile (Q0-Q1) and third quantile (Q2-Q3) are marked as separate whiskers on top and the bottom, median value (Q2) is shown as a red bar within the box shape (–), and finally, average is portrayed as a blue triangle shape (▲).

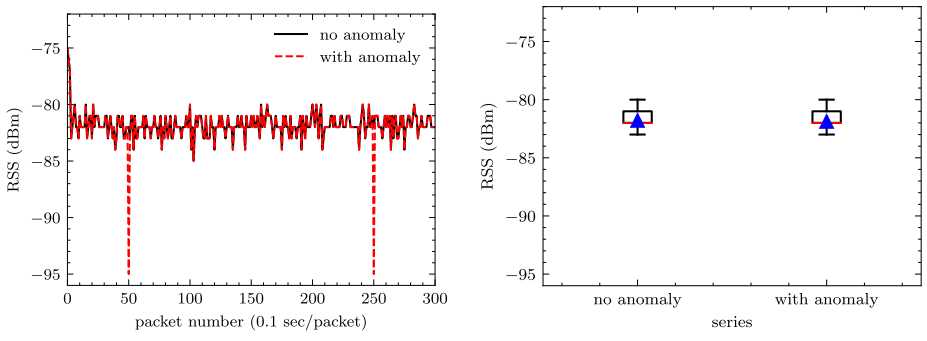


(a) Time-value perpective (b) Aggregated features perspective

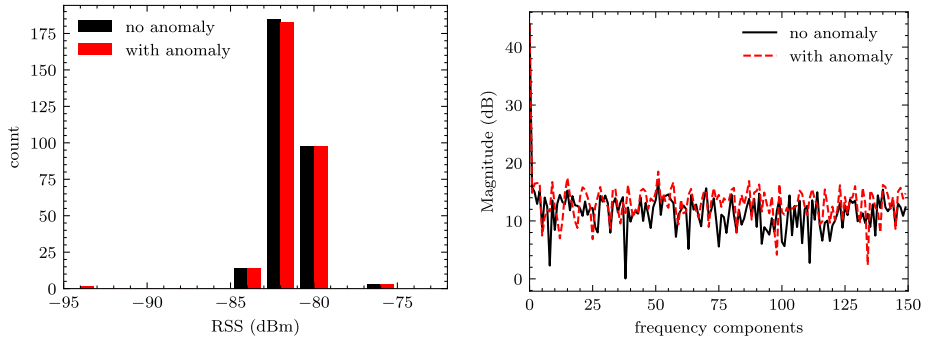


(c) Histogram perspective (d) Frequency perspective (FFT)

FIGURE 4. Distinct representations of the data for sudden degradation with recovery anomaly (SuddenR).



(a) Time-value perpective (b) Aggregated features perspective

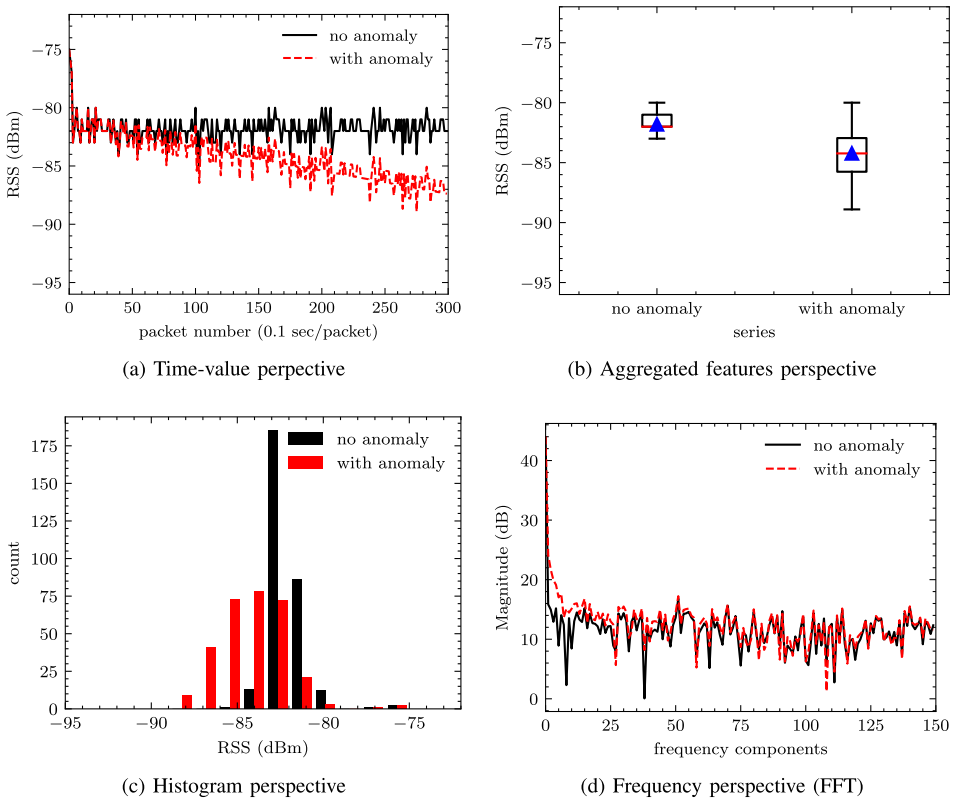


(c) Histogram perspective (d) Frequency perspective (FFT)

FIGURE 5. Distinct representations of the data for spike-like instantaneous degradation anomaly (InstaD).

C. HISTOGRAM REPRESENTATION

The histogram representation observed in Figures 3c, 4c, 5c and 6c is performed via splitting the range between (global) minimum and maximum values into ten equally-sized bins. More explicitly, this representation exhibits the percentage of values allotted in each bin.



(a) Time-value perpective

(b) Aggregated features perspective

(c) Histogram perspective

(d) Frequency perspective (FFT)

FIGURE 6. Distinct representations of the data for slow degradation anomaly (SlowD).

D. FFT REPRESENTATION

The frequency domain representation provided in Figures 3d, 4d, 5d and 6d utilizes absolute value of complex transformation, which is presented using log-scale for better contrasting ''with anomaly'' scenario against the ''no anomaly'' one.

E. ENCODED REPRESENTATION

A recent revolution of deep learning techniques, namely autoencoders, exhibits great performance returns in a diverse set of problems. To contrast against the above-mentioned traditional representations, we propose automatically generated encoded (autoencoder) representations for all anomaly types introduced in Section IV.

Autoencoders [16], [36], [37] are neural networks which are trained to generate a representation from the reduced encoding that is very similar compared its original input. The middle layer of an autoencoder is depicted with the purple circles in Figure 7 containing the reduced version of the input data and is referred to as a code $\mathbf{h}$ whose size is expected to be smaller than the size of the input data. As portrayed in Figure 7, an autoencoder is composed of two parts; i) an encoder function $\mathbf{h} = f(\mathbf{x})$, and ii) a decoder function producing a reconstruction $\hat{\mathbf{x}} = g(\mathbf{h})$. The autoencoders thus learn to include only the most useful signals from the input data, while mitigating the unnecessary signal noise.

An undercomplete autoencoder, where code size is smaller than input size, with nonlinear activation functions presents

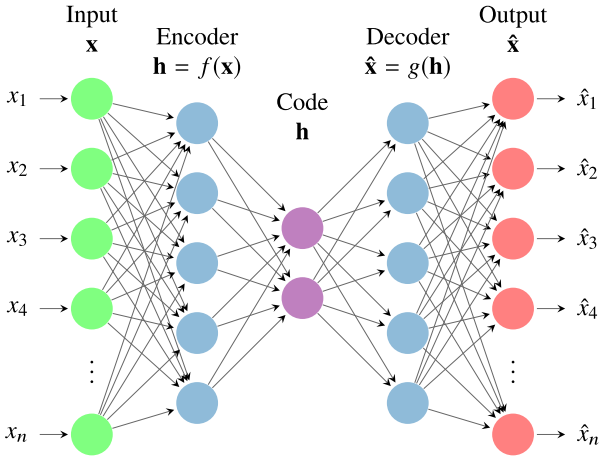
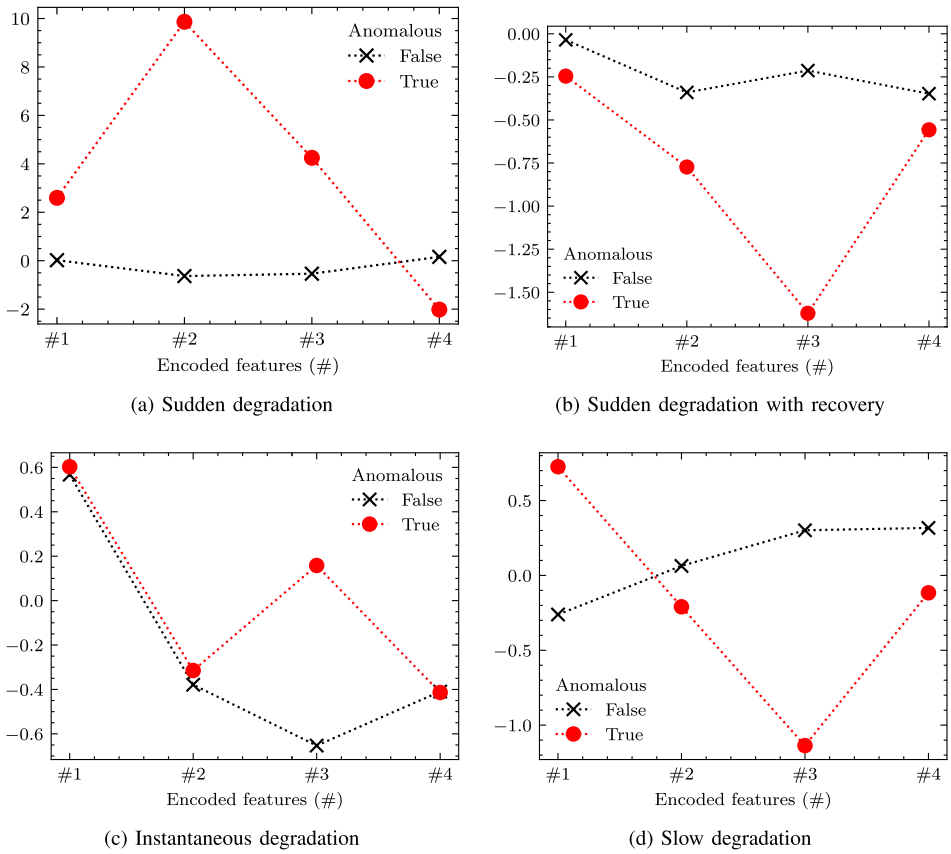


FIGURE 7. Illustration of autoencoder configuration during training process.

a generalized form of principal component analysis (PCA). Through the training process, the error between input $\mathbf{x}$ and output $\hat{\mathbf{x}}$ becomes negligible. Consequently, neural network learns a new representation of the input data, within a reduced feature-space. For example, in Figure 8a we transform time-value representation containing 300 dimensions into a newly encoded representation having only 4 dimensions. Figures 8a, 8b, 8c, and 8d present scenarios for a link with both; i) ordinary (non-anomalous) data, ii) anomaly injected (anomalous) data for SuddenD, SuddenR, InstaD



(a) Sudden degradation

(b) Sudden degradation with recovery

(c) Instantaneous degradation

(d) Slow degradation

FIGURE 8. Automatically generated features (code) exemplified for time-value representations.

and SlowD anomalies, respectively. Non-anomalous link is depicted with a solid black line, whereas anomalous link is marked with a dashed red line.

VI. APPROACHES FOR THE DETECTION OF ANOMALIES

Considering the link anomalies defined in Section IV and their corresponding representations depicted in Figures 3, 4, 5 and 6, it is clear that setting predefined thresholds for the investigated data would enable the detection of abnormal measurements and aid in treating them as an outlier. However, it has been proven that since fixed threshold-based approaches do not adapt to fluctuating behaviour of the data, selecting a threshold becomes consequential and thus may lead to poor performance, especially in real-time prediction applications [38]. On the contrary, adaptive and proactive approaches, such as deep learning neural network (DNN) and recurrent neural network (RNN) [38], can learn from regular patterns of the data and accurately identify abnormal behaviours to enable more accurate anomaly detection.

A. THRESHOLD BASED DETECTION

Considering Figure 2a, detecting SuddenD requires the diagnosis of steep falling slopes that do not recover for a relatively long, possibly predefined, period of time. Detecting SuddenR amounts to the identification of a sudden drop and later a boost in signal that resumes back to the original strength level within a predefined time window. SuddenR and InstaD are somewhat similar from application perspective. However, the distinction lies in the length of the time window at which the signal recovers back to its original levels within an instant of the time for InstaD. Detecting SlowD requires the diagnosis of a slowly but rather consistently falling slope for a relatively long, possibly predefined time window.

The time-value rules are a straightforward way to approach link-level anomaly detection. These rules may either be set based on an experienced arbitrary threshold or they can be identified using a theoretical or numerical method. However, as discussed in Section V, there are various possible ways to detect anomalies. For instance, it can be seen on Figures 3b, 4b and 6b that RSS distribution of an average healthy link is significantly different than the RSS distribution of the same link when anomaly is injected, which is readily distinguishable for SuddenD, SuddenR and SlowD anomalies at a glance. More explicitly, the spread of RSS for the anomaly injected link is wider, and its mean and median values are overwritten accordingly. Similar conclusions can be made for the respective histograms in Figures 3c, 4c and 6c. However, abnormal distributions in SlowD anomaly can only be detected with long-term observations. Moreover, sudden changes in time series can also

be detected in frequency domain, which in our case, are readily observed for SuddenD and SuddenR anomalies as larger magnitudes at lower frequencies in Figures 3b and 4b, respectively. Changes due to injected anomalies are almost indistinguishable in the case of InstaD and SlowD while leveraging frequency domain.

Details of the threshold strategy are provided in Section VIII. For time-value perspective, we consider D'Agostino-Pearson's normality statistical test [39], [40]. The test assesses whether certain set of points come from normal distribution or not. If the p value is below threshold, it is likely that the measurements do not come from normal distribution. Notice that Pearson's normality test is not sufficient condition for normality claims. Although, the approach may work fine for our limited line-of-sight scenario, it will not work for mobile or non line of sight scenario. For aggregated perspective, we consider for a link to have an anomaly two separate criteria. One criterion is based on the difference between mean and median values, which (if we assume normal distribution) are fairly close. The second criterion is how much can values deviate in standard deviation. Either of them has to be true for a link to be marked to have an anomaly. For histogram perspective, we define and arbitrary threshold. Anything below that is marked as an anomaly.

B. MACHINE LEARNING-BASED DETECTION

A ML model is expected to distinguish between anomalous and ordinary behaviours of a link, thus requires to solve a binary classification problem. There are two ways to train a ML model to identify such distinctions. The first one is based on a supervised training approach where all anomaly data are labelled, although in many practical applications, producing a reliable training dataset is expensive and it can inevitably cover only the type of anomalies that are present in the training dataset, which then cannot cope with the abnormal link behaviours in a comprehensive manner. For this reason, training a ML model in an unsupervised way is more practical, where learning from patterns of the overall link operations so as to distinguish the abnormal behaviours of a link from the anticipated behaviours is provoked, which is referred to as the automated detection of an outlier [41] or an anomaly [16] using ML models.

In addition to baseline threshold-based approach discussed in Section VI-A, we also consider three supervised and three unsupervised ML techniques as elaborated in the following sections.

1) SUPERVISED APPROACHES

To evaluate the performance of selected supervised ML techniques against each other and against the threshold-based approach, we opt for a set of candidate supervised approaches leveraging one representative technique from three different classes: i) Logistic Regression from Regression Analysis [42], ii) Random Forest from tree ensemble class [43] and iii) Support Vector Machines (SVM) from kernel-method class [43].

Logistic Regression [42] is a modified linear regression able to work on classification problems. In linear regression the goal is to fit a line to data samples and minimize loss. Similarly, logistic regression aims for fitting sigmoid function with the goal to minimize loss at predicting any two classes. Logistic regression also includes a generalized form suitable for high-dimensional input data and multi-class rather than binary classification.

Random Forests [44] is an ensemble method that uses a number of decision tree classifiers followed by a voting mechanisms to perform multi-class classification. The trees are learnt by randomly splitting a relatively large feature space into smaller subspaces. Each tree provides a class in which a specific data point falls into, the class corresponds to the ''vote'' of that tree. The final outcome of the classifier then uses a mechanism, such as majority voting to provide the final result.

Support Vector Machine [45] is a learning algorithm that belongs to the family of kernel methods. Roughly speaking, SVMs attempt to learn a hyperplane that best splits a set of data into two classes. The shape of the hyperplane depends on the type of kernel function selected for the algorithm. When the kernel function is linear, so is the learnt hyperplane. When non-linear kernels are chosen, for instance RBF kernel [46], then the hyperplane is non-linear therefore better suited to approximate or discriminate non-linear random variables.

2) UNSUPERVISED APPROACHES

The cost of producing labels for supervised learning is discussed in Section VI-B. As a countermeasure, we also consider a set of candidate unsupervised approaches for developing anomaly detection models [43], where we leverage one representative technique from three different classes: i) Local Outlier Factor from Nearest Neighbour (NN) class [43], ii) Isolation Forest from tree ensemble class [43] and iii) one-class Support Vector Machines (SVM) from kernel-method class [43].

Local Outlier Factor [47] belongs to the k-Nearest Neighbour (kNN) family of algorithms, which rely on the computation of the distance between data points of the feature space. The feature vectors with smaller distance are alike and thus clustered together. One drawback for this family of algorithms is that as the dimensionality of the training data grows, the computational complexity evolves exponentially. However, there have been attempts in circumventing this exponential complexity, e. g., Ball Tree.

Isolation Forest [48] belongs to tree-based ensemble methods, and works in a roughly similar way as Random Forests as described above. Essentially, it represents a Random Forest adapted so that it optimizes outlier detection rather than multi-class classification of majority of data it sees. Based on certain metrics and distinct criteria, the algorithm decides whether particular subspaces contain any abnormal samples, namely anomalies.

Support Vector Machine, as described at the end of supervised approaches, can also be used in an unsupervised mode

for anomaly detection. In fact, most ML techniques can be used in both supervised and unsupervised mode. With this one-class approach, the model is expected to distinguish data as negative or positive instances. Then, the model can learn the boundaries of the data so as to detect the points that lie outside the boundary exposed as anomalies or outliers.

VII. METHODOLOGY AND EXPERIMENTAL DETAILS

Before we proceed with the analysis of the relative performance of the wireless link anomaly detection approaches proposed in this paper, we provide relevant methodological and experimental details.

A. TRAINING DATASET GENERATION

For our experimental evaluation, we consider a real-world measurement dataset, i.e., Rutgers [49], which contains measurements from 29 nodes at 5 different noise levels and each record has 300 measurements. Although every link is measured at five different noise levels, wee consider each recording as a different link and we assume that there is no correlation. On this existing real-world dataset we synthetically inject the four types of anomalies proposed in this paper as follows. First, we only pick the links without packet loss. This reduces our dataset from 4 060 to 2 123 ($\approx$ 52%) of independent links. Second, by means of applying one anomaly type at a time, we randomly pick 33% of these links, at which the anomaly is injected according to guidelines in Table 1, while the remaining is left intact.

TABLE 1. Artificial anomaly injections for each anomaly scenario.

Type	Links	Affected	Appearance	Persistence
SuddenD	2 123	33% (700)	once, [200th, 280th]	for ∞
SuddenR			once, [25th, 275th]	for [5, 20]
InstaD			on $\approx$1% of a link	for 1 datapoint
SlowD			once, [1st, 20th]	for [150, 180]†

† RSSI(x, start) $\leftarrow$ RSSI(x) + min(0, $-$rand(0.5, 1.5) $\cdot$ (x $-$ start))

The suddenD anomaly, observed in Figure 2a, on the affected link appears arbitrarily between 200th and 280th packet and it persists indefinitely. In case of suddenR, observed in Figure 2b, anomaly applied on the link appears only once with a random start from 25th to 275th packet, where it persists for an arbitrary duration between 5 to 20 measurements. For InstaD of Figure 2c, the anomaly can appear anywhere in the entire series with 0.01 probability, which means that each anomaly on the affected link appears three times on average. Finally, SlowD anomaly of Figure 2d appears arbitrarily between 1st and 20th measurements, where it commences with a random degrading pace of duration between 150 and 280 packets. In a nutshell, anomaly injection details are provided in Table 1.

B. COMPUTING STANDARD AND ENCODED REPRESENTATIONS

Once anomalies are injected as specified in Table 1, we compute four different data representations described in Section V. The first one, namely time-value representation of Section V-a, converts each link into a single feature vector containing 300 features. The second one, the so-called aggregated feature, summarizes each link with 7 features, which are described in Section V-b. The third one, namely histogram feature discussed in Section V-c, defines ten equally spaced bins, which are then presented to a model as a feature vector containing 10 features. The forth one, namely frequency feature elaborated in Section V-d, gives the model a large feature vector of frequency-domain representation summing up to nearly 150 features. As we compute four representations for each of the four types of anomalies, we generate 16 candidate datasets.

Next, we also consider autoencoders for each anomaly scenario and each of the four standard representations. As any other deep neural network, autoencoder also requires many iterations of training. To produce credible results with autoencoder, we build the generic model in two steps. In the first step, we split the dataset into training and test groups with a 60:40 ratio, respectively. In the second step, when the weights of the autoencoder are converged, we perform an end-to-end evaluation on the test group. Relevant autoencoder configurations are provided in Table 2, where the layers and their required parameters are outlined for the encoder and the decoder. Although recent trends in DNNs go towards the use of convolutional layers, a convolution layer would make sense only in case of time-value and frequency perspective, due to their reasonable size and correlated neighbouring vector values. Therefore, our decision is to go with fully connected (dense) layers. For the activation part, we use batch normalization (BN) followed by Leaky Rectified Linear Unit (leaky ReLU, or LReLU) with $\alpha = 0.2$ coefficient for negative values. While plain ReLU is most widely used

TABLE 2. Autoencoder configurations.

Role	Layer	Notes
Encoder	Input(*)	
	Dense(128)	
	BN + LeakyReLU(α = 0.2)	
	Dense(64)	
	BN + LeakyReLU(α = 0.2)	
	Dense(32)	
	BN + LeakyReLU(α = 0.2)	
	Dense(4)	no activation
Decoder	Input(4)	
	Dense(32)	
	BN + LeakyReLU(α = 0.2)	
	Dense(64)	
	BN + LeakyReLU(α = 0.2)	
	Dense(128)	
	BN + LeakyReLU(α = 0.2)	
	Dense(*)	no activation

* input/output size depends on feature vector
- Implementation of autoencoders in TensorFlow/Keras is available at: https://gist.github.com/gcerar/5e4e53902493632a3cfb5cc06c3317b7

non-linear activation function, its leaky version has shown several benefits and minor overall improvements [50].

To produce the encoded representations, we feed the 16 datasets corresponding to the representation provided in Sections V-(a),(b),(c),(d) into the autoencoder, resulting in additional 16 candidate datasets. Therefore, to continue with the anomaly detection, we train both supervised and unsupervised ML models on a total of 32 datasets, 16 corresponding to the four standard representations of each anomaly and the other 16 corresponding to the encoded representations.

C. PERFORMING AUTOMATIC ANOMALY DETECTION

Next, we compute the performance of the threshold, three supervised and three unsupervised ML techniques described in Section VI on the 32 generated datasets corresponding to the proposed anomalies and representations. Each approaches' output is compared to a label to identify whether the link actually contains anomalies or not.

1) THRESHOLD APPROACH

Descriptive details of leveraging certain thresholds for each anomaly can be found in Section VI-A. The utilized experimental threshold parameters are listed in Table 3. The threshold for the time-series representation that uses the D'Agostino-Pearson's normality statistical test [39], [40] is $p < 10^{-3}$. The threshold for the aggregated representation assumes the absolute difference between mean and median is higher than $3dB$ or that the double of the standard deviation is higher than $2.5dB$. The threshold for the histogram representation is set at $RSSI < -85dBm$ while threshold selection for the FFT and encoded representations were infeasible to find using our trial-and-error approach. The differences in the FFT representation are not easily visible or detectable using simple methods while the encoded representations cannot be easily interpreted, therefore also deriving an appropriate threshold is not possible.

TABLE 3. Predetermined anomaly thresholds.

Features	Anomaly thresholds
Time-series	Normality test [39], [40], when p < 10^{-3}
Aggregated	(|mean - median| > 3 dB) **OR** (2·stdev > 2.5 dB)
Histogram	RSSI < -85 dBm

2) MACHINE LEARNING-BASED APPROACHES

For each of the six selected ML techniques, we use standard ML cross-validation.[3] We train the models using shuffled data split into training and test sets with a 80:20 ratio, respectively. Model is trained with the training set and evaluated using the test set in order to ensure credible results. We use standard metrics for evaluating classifiers: precision, recall and F1 score. Precision measures how many of the instances detected as class A actually belong to class A, expressed as; Precision $= \frac{TP}{TP+FP}$, whereas recall measures how many of the instances belonging to class A were actually detected, expressed as; Recall $= \frac{TP}{TP+FN}$, where TP, FP and FN stand for true positives, false positives and false negatives, respectively. F1 score is quantified by the harmonic mean of the precision and the recall, where larger values indicate better classifiers with balanced and higher precision and recall performances.

For each of the ML techniques selected in Section VI, Table 4 lists the respective implementations and parameters used in the experiments. For instance, for logistic regression we use the LogisticRegression implementation available in the Python Scikit Learn toolbox.[4] As the LogisticRegression implementation enables setting 12 different parameters that influence the final model, we generally select standard values that have been proven to work on large number of cases and datasets by the ML community. However, we identify selected parameters that should be optimized, such as the regularization strength C in this case. We search for the best configuration by adapting an array of possible values $C \in [10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}]$ and ultimately select the best performing regularization factor C among them. For instance, Figure 9 presents the scenario where a model is trained using LR on time-value representation for SuddenD anomalies and based on robust scaler. For this particular scenario, the best $F1$ score of this model is attained by means of setting C to any value that is larger than 1. For the results presented in the next sections, we only account for the best $F1$ scores obtained after searching for such near-optimal regularization parameter values.

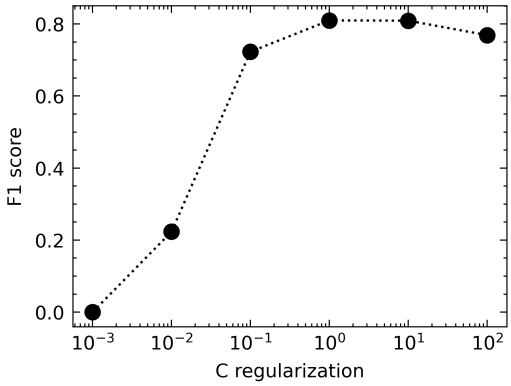


FIGURE 9. Regularization parameter (C) search for selecting the best performing model that is, for example, trained using LR on time-value representation for SuddenD anomalies and based on robust scaler.

The implementations chosen for the remaining algorithms also include over ten possible input parameters. For LOF, we vary the number of neighbours, algorithm and leaf size for finding the best performing model. For RForest and IForest, we vary the number of base estimators, whereas for SVM and

[3]Stratified K-Fold cross validation is implemented by using StratifiedKFold parameter in Python Scikit Learn toolbox https://scikit-learn.org/stable/

[4]https://scikit-learn.org/stable/

TABLE 4. ML techniques and their relevant parameters.

Approach	Technique	Implementation	Parameters and their range
Supervised	Logistic Regression (LR)	LogisticRegression from sklearn	penalty='l2', dual=False, tol=1e-4, C= (1e-3, 1e-2, 1e-1, 1.0, 10., 100.) fit_intercept=True, intercept_scaling=1, class_weight=None, solver='lbfgs', l1_ratio=None
	Random Forest (RForest)	BaggingClassifier from sklearn	base_estimator=None, n_estimators=[10, 20, 30, 40, 50, 70, 100], max_samples=1.0, max_features=1.0, oob_score=False, intercept_scaling=1,
	Support Vector Machine (SVM)	SVC from sklearn	C=(1e-3, 1e-2, 1e-1, 1.0, 10., 100.), kernel=('linear', 'rbf'), gamma=('auto', 'scale'), tol=1e-3, decision_function_shape='ovr', break_ties=False
Unsupervised	Local Outlier Factor (LOF)	LocalOutlierFactor from sklearn	n_neighbors=[5, 10, 20, 40, 50, 80], algorithm=['ball_tree', 'kd_tree', 'brute'], leaf_size=[10, 30, 50, 80], p=[1, 2] metric_params=None, contamination="auto",
	Isolation Forest (IForest)	IsolationForest from sklearn	n_estimators=[10, 20, 30, 40, 50, 70, 100], max_samples='auto', contamination='auto', max_features=1.0, bootstrap=False,
	Support Vector Machine (OC-SVM)	OneClassSVM from sklearn	nu=[0.10, 0.3, 0.5, 0.70, 0.90, 1.0], kernel=('linear', 'rbf'), gamma=('auto', 'scale'), coef0=0.0, tol=1e-3,

OC-SVM, we vary the regularization factor C, the kernel and the kernel coefficient *gamma* for the *rbf* kernel, respectively.

As some of the models are sensitive to scaling, we also consider training on data that is; i) not scaled, ii) scaled by using mean values, iii) scaled using mean and deviation, and iv) scaled using min-max. The entire procedure and parameters can be readily found and used in the existing public open source repository.[5] Six selected ML techniques with the associated parameter tuning are trained over the 32 datasets, totalling at more than 40,000 anomaly detection models.

VIII. EVALUATION

In this section, we evaluate the relative performance of various data representations discussed in Section V and of approaches discussed in Section VI for detecting four types of anomalies introduced in Section IV. The methodological and experimental details utilized for obtaining the results are elaborated in Section VII.

A. PERFORMANCE ANALYSES OF DATA REPRESENTATIONS

In this section, we first provide insight into how a model learns to classify by discussing the importance of various features resulting from the four manually generated and interpretable representations for discriminating the four types of anomalies defined in Section IV. Next, we discuss the influence of the five data representations, including those four manually generated ones and the automatically generated (autoencoder) one, as elaborated in Section V, on the performance of the learnt models. This entire subsection focuses on the influence of representations on the final models, while the influence of the ML approaches is analysed in Section VIII-B.

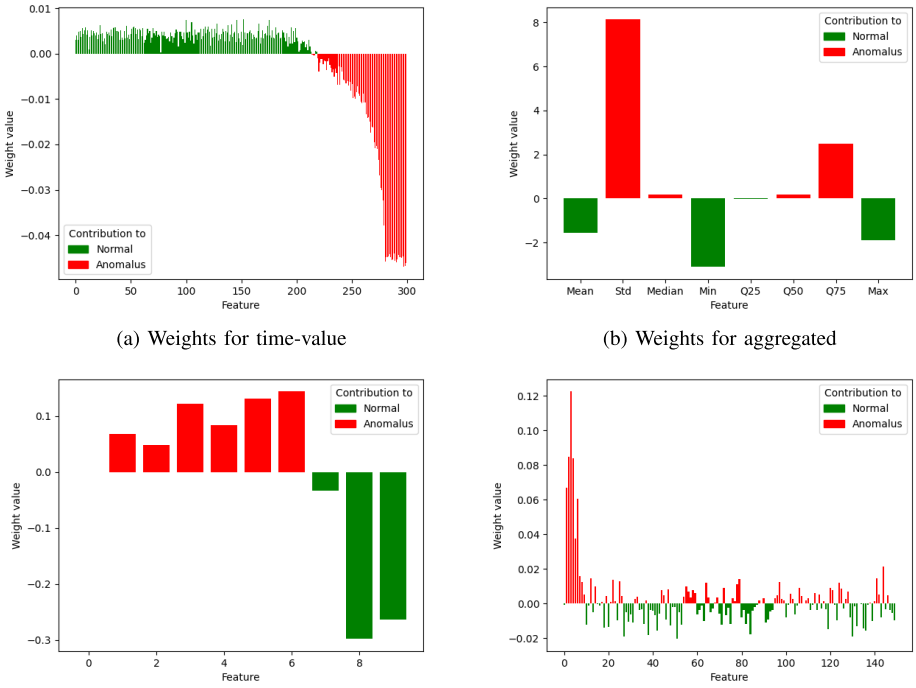
[5]Script for the design and development of anomaly detection models excluding data preprocessing is available at: https://gist.github.com/gcerar/0b03e55f41147a7b7230f45d1f1209d6

1) ANALYSING THE DISCRIMINATIVE IMPORTANCE OF FEATURES

For analysing the discriminative power of the features in learning to classify the four anomaly types, we choose LR for its simplicity and reasonable tractability. As explaining the meaning of the automatically generated features is infeasible, we exclude them from this part of the analysis, without loss of generality.

Figures 10, 11, 12 and 13 depict the weights learnt by the LR on the representations discussed in Section V. Each set of figures corresponds to an anomaly type, namely SuddenD, SuddenR, InstaD and SlowD. In the above-referred figures, the green weights depict the features that are important for identifying normal links, whereas the red weights are important for detecting the anomalous links. Using these learnt features, it is possible to look at the LR as a linear function with as many variables as the length of the feature vector, e.g., 300 for time-values representation and 8 for the aggregated. Each point in the feature vector has its corresponding weight with which it is multiplied. When all multiplications (weight * variable(n)) are summed up, a positive or a negative value corresponding to one of the two classes are obtained, i.e., normal or anomalous links.

For the case of the *time-value representation* of the SuddenD anomaly from Figure 3a, it can be seen that the points depicted with red, mostly starting from somewhere after feature 200 play a more important role when making the decision on whether an input feature vector contains an anomaly or not. The reason why LR learns that these features are the most important ones can be explained from the way the SuddenD anomaly is injected in the training dataset. According to Table 1, SuddenD is injected randomly between packets 200 and 280. Therefore LR learns that those points are more discriminative for the anomalies. Simplistically, when multiplying the anomalous vector from Figure 3a with the weights in Figure 10a, and subsequently summing up, the result will become positive, and hence the input will be classified as



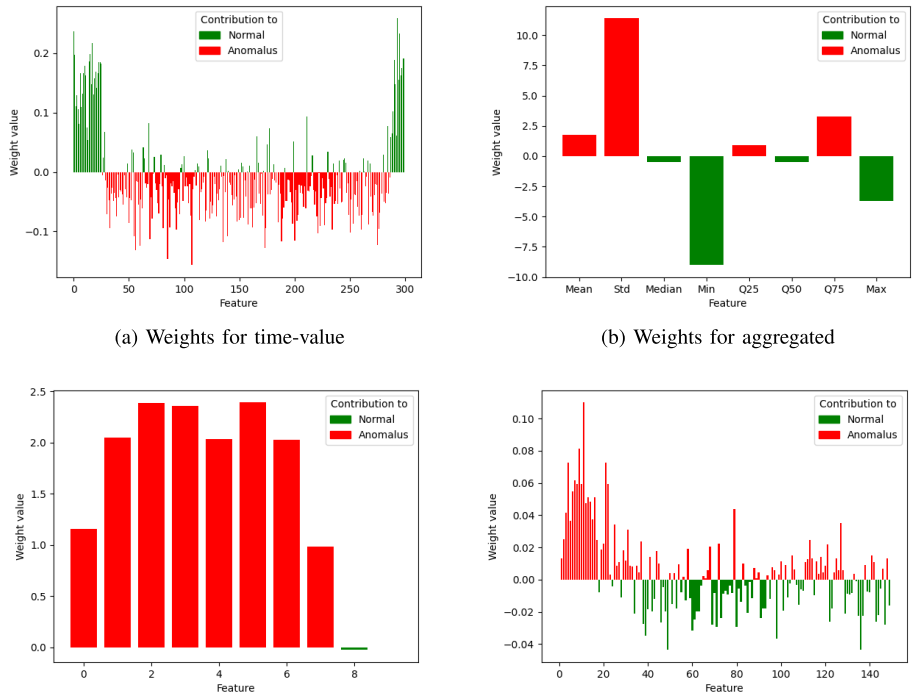
(a) Weights for time-value

(b) Weights for aggregated

(c) Weights for histogram

(d) Weights for frequency (FFT)

FIGURE 10. Learnt feature importance for distinct representations of the data for sudden degradation anomaly (SuddenD).



(a) Weights for time-value

(b) Weights for aggregated

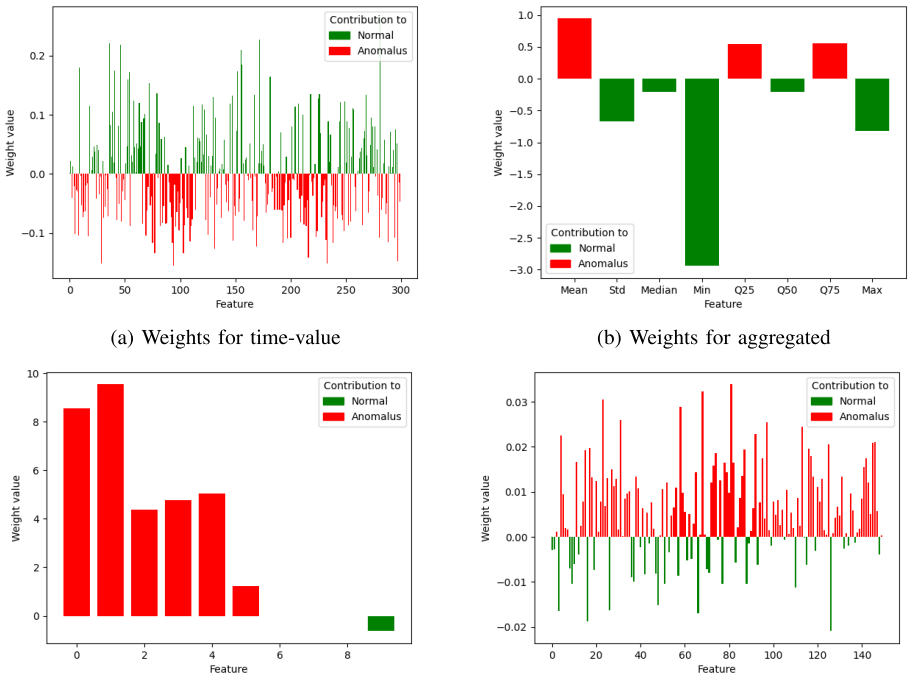
(c) Weights for histogram

(d) Weights for frequency (FFT)

FIGURE 11. Learnt feature importance for distinct representations of the data for sudden degradation with recovery anomaly (SuddenR).

anomaly. On the other hand, when the normal vector from Figure 3a is multiplied with the weights in Figure 10a, upon summing them up, the result will become negative, thus the vector will be classified as normal.

Similar discussions over time-value representations can be made for all the other anomalies. SuddenR anomaly is randomly injected between packets 25 and 275 of the time-value representation as per Table 1, and it can be seen from



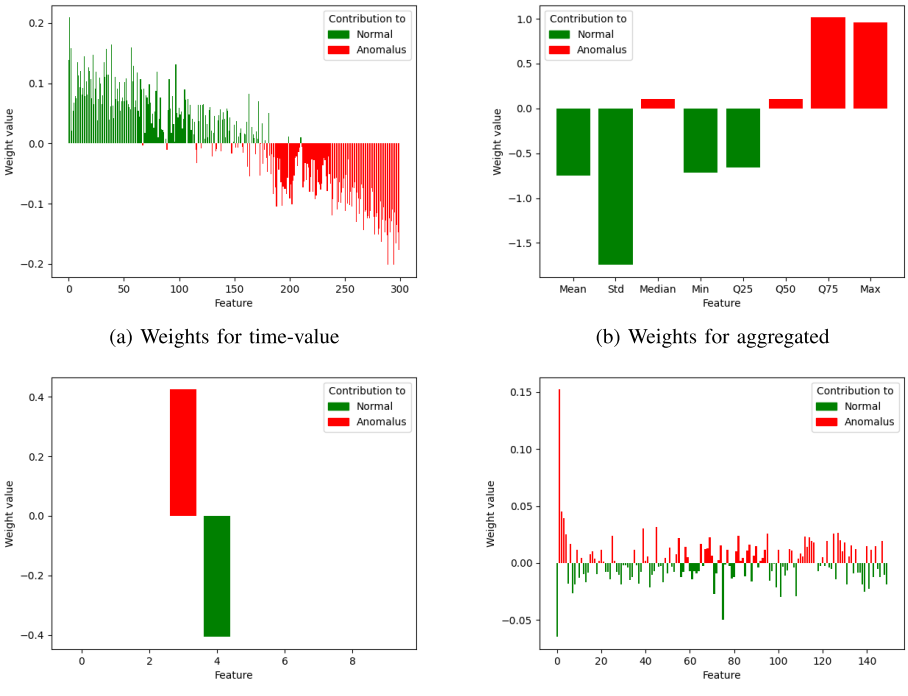
(a) Weights for time-value

(b) Weights for aggregated

(c) Weights for histogram

(d) Weights for frequency (FFT)

FIGURE 12. Learnt feature importance for distinct representations of the data for spike-like instantaneous degradation anomaly (InstaD).



(a) Weights for time-value

(b) Weights for aggregated

(c) Weights for histogram

(d) Weights for frequency (FFT)

FIGURE 13. Learnt feature importance for distinct representations of the data for slow degradation anomaly (SlowD).

Figure 11a that the most important features for detecting the anomaly, represented with red, lie within this range. The importance of features for the spike anomaly that is quite random in nature and also occurs often in the data due to the nature of the wireless channel is depicted in Figure 12a. Finally, the importance of the features for detecting SlowD

TABLE 5. Performance of detecting sudden degradation (SuddenD) anomalies.

Approach	Technique	time-value features			aggregated features			histogram features			frequency domain		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Baseline	Threshold (Tab. 3)	0.66	1.00	0.79^{1}	0.97	1.00	0.98^{1}	0.44	1.00	0.61^{1}	-	-	-
Supervised	LR	1.00	1.00	$\mathbf{1.00}^{1}$	1.00	1.00	$\mathbf{1.00}^{2}$	0.99	0.99	0.99^{1}	1.00	1.00	$\mathbf{1.00}^{1}$
	encoder + LR	1.00	1.00	$\mathbf{1.00}^{6}$	1.00	1.00	$\mathbf{1.00}^{3}$	1.00	1.00	$\mathbf{1.00}^{6}$	1.00	1.00	$\mathbf{1.00}^{1}$
	RForest	1.00	1.00	$\mathbf{1.00}^{1}$	0.99	0.99	0.99^{4}	0.99	1.00	$\mathbf{1.00}^{4}$	1.00	1.00	$\mathbf{1.00}^{1}$
	encoder + RForest	1.00	1.00	$\mathbf{1.00}^{1}$	1.00	1.00	$\mathbf{1.00}^{4}$	1.00	1.00	$\mathbf{1.00}^{4}$	1.00	1.00	$\mathbf{1.00}^{1}$
	SVM	1.00	1.00	$\mathbf{1.00}^{1,7}$	1.00	1.00	$\mathbf{1.00}^{4,7}$	0.99	1.00	$\mathbf{1.00}^{5,8}$	1.00	1.00	$\mathbf{1.00}^{1,7}$
	encoder + SVM	1.00	1.00	$\mathbf{1.00}^{1,8}$	1.00	1.00	$\mathbf{1.00}^{4,8}$	1.00	1.00	$\mathbf{1.00}^{4,7}$	1.00	1.00	$\mathbf{1.00}^{1,7}$
Unsupervised	LOF	0.36	0.53	0.43^{1}	0.51	0.38	0.43^{6}	0.88	0.67	0.76^{4}	1.00	0.20	0.33^{5}
	encoder + LOF	0.85	0.25	0.38^{3}	0.65	0.16	0.26^{4}	0.65	0.19	0.29^{1}	0.59	0.19	0.29^{2}
	IForest	0.98	0.48	0.64^{1}	0.90	0.77	0.83^{4}	0.91	0.60	0.72^{4}	0.89	0.47	0.61^{2}
	encoder + IForest	0.94	1.00	$\mathbf{0.97}^{3}$	0.89	0.86	$\mathbf{0.88}^{2}$	0.94	1.00	$\mathbf{0.97}^{3}$	0.94	0.99	$\mathbf{0.97}^{5}$
	OC-SVM	0.87	0.93	$\mathbf{0.90}^{4,8}$	0.81	0.86	$0.83^{1,8}$	0.94	0.99	$\mathbf{0.96}^{5,8}$	1.00	0.96	$\mathbf{0.98}^{2,7}$
	encoder + OC-SVM	1.00	0.84	$\mathbf{0.91}^{3,7}$	0.99	0.93	$\mathbf{0.96}^{5,7}$	1.00	0.84	$\mathbf{0.91}^{1,7}$	0.98	0.99	$\mathbf{0.99}^{3,7}$

is higher in the second half of the feature vector as depicted in Figure 13a since that's where the degradation becomes more evident.

Moving to *aggregated representations*, it can be seen from Figure 10b that standard deviation (Std) and the last quantile (Q75) are the most important features for detecting the anomaly, with minor contribution from the median and Q50. This is because standard deviation increases when SuddenD anomaly is present while the count of high RSSI values in the last quantile is smaller when this anomaly is present. Next, for SuddenR, the two main features remain the same as the shape is very similar to the SuddenD as can be seen in Figure 11b, albeit the duration differs leading to a more prominent influence of the mean for discrimination. For InstaD, that can be seen as a very narrow SuddenR randomly appearing on 1% of the link, Std looses importance while the mean and two quantiles become more predictive as depicted in Figure 12b. For SlowD, the model learns that features which inform about the slope that appears and increases, therefore Q75 counting high RSSI values and the maximum (max) are predictive. The median and Q50 that capture the intermediate values of the slowly increasing slope also add minor discriminative power, as portrayed in Figure 13b.

In the case of *histogram representation*, the first bins where *cumulated* low RSSI values corresponding to SuddenD, SuddenR and InstaD anomalies are the most important ones according to Figures 10c, 11c and 12c. For the case of SlowD presented in Figure 13c, one of the middle bins that capture intermediate values is the most discriminative while the other bins seem to not contribute to either class.

Finally, the importance of features in the case of *frequency representation* presents a similar line of reasoning as for the other representations. For SuddenD and SuddenR anomaly amplitudes at low frequencies that introduce a major shift in the mean are the most important features, as portrayed in Figures 10d and 11d. For InstaD there is no clear importance pattern as shown in Figure 12d, whereas for SlowD the feature amplitudes around 0 are the most prominent ones as illustrated in Figure 13d.

2) THE INFLUENCE OF THE REPRESENTATIONS ON THE PERFORMANCE OF THE LEARNT MODELS

The best performing results of the classification with respect to F1 score are presented in Table 5 for SuddenD, Table 6 for SuddenR, Table 7 for InstaD and Table 8 for SlowD. The first column of the tables lists the approach, the second column outlines the used ML techniques, while columns 3 to 6 list the results for time-value, aggregated, histogram and FFT representations, respectively.

The encoded representation introduced in Section V-e and employed according to the methodology in Section VII-B is inserted into the above-mentioned performance tables with the name of respective ML technique using the term "encoder". More precisely, referring to the rows corresponding to the ML technique, say IForest, the performance results are implemented for the four mentioned representations for the IForest ML technique. Additionally, at the row entitled "Encoder + IF", the numerical results refer to the IForest ML technique that is applied to the codes generated from the four representations, respectively. Finally, the superscripts identify the scaling methods utilized. The three highest F1 scores for supervised approaches and the three highest F1 scores for unsupervised approaches are delineated in bold font.

With respect to the data representations, from the results listed in Tables 5, 6, 7 and 8, two high level observations are outlined as follows.

- None of the four manually generated features clearly dominates the remaining ones in terms of anomaly detection performance.

TABLE 6. Performance of detecting sudden degradation with recovery (SuddenR) anomalies.

Approach	Technique	time-value features			aggregated features			histogram features			frequency domain		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Baseline	Threshold (Tab. 3)	0.66	1.00	0.79^{1}	0.97	0.97	0.97^{1}	0.44	1.00	0.61^{1}	-	-	-
Supervised	LR	0.92	0.86	0.89^{3}	0.99	0.99	$\mathbf{0.99}^{3}$	1.00	0.98	$\mathbf{0.99}^{2}$	1.00	1.00	$\mathbf{1.00}^{2}$
	encoder + LR	0.99	0.98	$\mathbf{0.99}^{2}$	0.99	0.99	$\mathbf{0.99}^{6}$	1.00	0.99	$\mathbf{0.99}^{2}$	1.00	1.00	$\mathbf{1.00}^{2}$
	RForest	0.96	0.96	0.96^{3}	0.99	0.99	$\mathbf{0.99}^{2}$	1.00	0.99	$\mathbf{0.99}^{5}$	0.99	0.99	0.99^{5}
	encoder + RForest	0.99	0.99	$\mathbf{0.99}^{5}$	0.99	0.98	$\mathbf{0.99}^{6}$	1.00	0.99	$\mathbf{0.99}^{6}$	1.00	1.00	$\mathbf{1.00}^{1}$
	SVM	0.98	0.96	$0.97^{2,8}$	0.99	0.99	$\mathbf{0.99}^{5,8}$	0.99	0.99	$\mathbf{0.99}^{3,8}$	1.00	1.00	$\mathbf{1.00}^{1,7}$
	encoder + SVM	0.99	0.98	$\mathbf{0.99}^{5,7}$	0.99	0.99	$\mathbf{0.99}^{6,8}$	1.00	0.99	$\mathbf{0.99}^{2,7}$	1.00	1.00	$\mathbf{1.00}^{2,8}$
Unsupervised	LOF	0.88	0.99	$\mathbf{0.93}^{5}$	0.53	0.39	0.45^{6}	0.98	0.97	$\mathbf{0.98}^{2}$	1.00	0.39	0.56^{5}
	encoder + LOF	0.95	0.61	0.74^{1}	0.67	0.16	0.26^{2}	0.79	0.27	0.40^{3}	0.80	0.29	0.43^{1}
	IForest	0.48	0.20	0.28^{1}	0.95	0.62	0.75^{1}	0.99	0.26	0.41^{1}	0.94	0.49	0.64^{6}
	encoder + IForest	0.98	0.97	$\mathbf{0.98}^{5}$	0.93	0.98	$\mathbf{0.95}^{2}$	0.95	0.96	$\mathbf{0.95}^{5}$	0.97	0.97	$\mathbf{0.97}^{3}$
	OC-SVM	0.92	0.98	$\mathbf{0.95}^{2,8}$	0.98	0.82	$\mathbf{0.89}^{2,7}$	0.93	0.95	$\mathbf{0.94}^{5,8}$	1.00	0.83	$\mathbf{0.91}^{2,7}$
	encoder + OC-SVM	0.81	0.87	$0.84^{5,8}$	0.76	0.81	$\mathbf{0.78}^{2,8}$	0.74	0.96	$0.83^{5,7}$	0.73	0.98	$\mathbf{0.84}^{6,7}$

TABLE 7. Performance of detecting spike (InstaD) anomalies.

Approach	Technique	time-value features			aggregated features			histogram features			frequency domain		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Baseline	Threshold (Tab. 3)	0.64	0.97	0.77^{1}	0.94	0.67	0.78^{1}	0.42	1.00	0.60^{1}	-	-	-
Supervised	LR	0.92	0.87	0.89^{1}	0.96	0.99	0.97^{2}	0.95	0.95	0.95^{1}	0.97	0.91	0.94^{1}
	encoder + LR	0.95	0.92	$\mathbf{0.94}^{4}$	0.98	0.98	$\mathbf{0.98}^{5}$	0.98	0.95	$\mathbf{0.97}^{3}$	0.98	0.94	$\mathbf{0.96}^{4}$
	RForest	0.98	0.86	0.91^{3}	0.98	0.96	0.97^{6}	0.98	0.95	$\mathbf{0.96}^{2}$	0.96	0.89	0.92^{1}
	encoder + RForest	0.96	0.91	$\mathbf{0.94}^{4}$	0.97	0.97	0.97^{6}	0.98	0.95	$\mathbf{0.96}^{3}$	0.97	0.93	$\mathbf{0.95}^{4}$
	SVM	0.96	0.91	$\mathbf{0.94}^{3,8}$	0.97	0.98	$\mathbf{0.98}^{5,8}$	0.97	0.96	$\mathbf{0.96}^{6,8}$	0.97	0.91	$0.94^{1,8}$
	encoder + SVM	0.95	0.93	$\mathbf{0.94}^{4,7}$	0.98	0.98	$\mathbf{0.98}^{4,7}$	0.98	0.95	$\mathbf{0.97}^{6,8}$	0.99	0.93	$\mathbf{0.96}^{4,8}$
Unsupervised	LOF	0.79	0.86	$\mathbf{0.82}^{2}$	0.60	0.32	0.42^{3}	0.94	0.85	$\mathbf{0.89}^{5}$	0.90	0.25	0.39^{4}
	encoder + LOF	0.89	0.28	0.43^{5}	0.58	0.26	0.36^{6}	0.65	0.27	0.38^{3}	0.45	0.11	0.17^{3}
	IForest	0.29	0.16	0.21^{1}	0.67	0.73	0.70^{1}	0.98	0.34	0.50^{5}	0.97	0.42	0.59^{2}
	encoder + IForest	0.91	0.82	$\mathbf{0.86}^{2}$	0.87	0.97	$\mathbf{0.92}^{1}$	0.80	0.96	$\mathbf{0.87}^{1}$	0.82	0.97	$\mathbf{0.89}^{1}$
	OC-SVM	0.77	0.87	$\mathbf{0.82}^{5,8}$	0.99	0.82	$\mathbf{0.90}^{2,7}$	0.76	0.82	$0.79^{5,8}$	0.82	0.90	$0.86^{6,7}$
	encoder + OC-SVM	0.76	0.85	$0.80^{3,8}$	0.71	0.91	$\mathbf{0.80}^{4,7}$	0.70	0.80	$0.75^{3,8}$	0.92	0.95	$\mathbf{0.93}^{1,7}$

- In most cases, automatically generated encoded data representation improves anomaly detection performance compared to the same non-encoded counterpart.

a: SuddenD ANOMALIES

For *SuddenD* observed in Table 5, all representations produce nearly perfect F1 scores of above 0.99 with all supervised ML approaches. Moving to unsupervised approaches, it can be readily seen that the histogram representation works best with LOF, however the F1 score of 0.76 is modest. The aggregated features with $F1 = 0.83$ work best with IForest followed by the histogram features with $F1 = 0.72$. The encoded representations surpass all non-encoded ones with this approach reaching F1 scores up to 0.97. All but the manual aggregated features yield good F1 scores of above 0.9 with OC-SVM, however the frequency representation dominates with F1 score of above 0.98. The encoded representations improve the anomaly detection performance in three of the four possible cases.

b: SuddenR ANOMALIES

For *SuddenR* observed in Table 6, almost all representations produce high F1 scores of above 0.9 with all supervised ML approaches. The time-value representation is slightly inferior to the other manual and autoencoded representations, producing 0.89 F1 score with LR, 0.96 with RForest and 0.97 with SVM.

TABLE 8. Performance of detecting slow degradation (SlowD) anomalies.

Approach	Technique	time-value features			aggregated features			histogram features			frequency domain		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Baseline	Threshold (Tab. 3)	0.17	0.10	0.13[1]	0.47	0.03	0.06[1]	0.31	0.57	0.40[1]	-	-	-
Supervised	LR	0.97	0.96	0.97[3]	0.44	0.16	0.91[4]	0.92	0.87	0.90[2]	0.92	0.89	0.90[6]
	encoder + LR	0.99	0.98	**0.99**[2]	0.99	1.00	**1.00**[3]	1.00	1.00	**1.00**[1]	0.98	0.98	**0.98**[4]
	RForest	0.98	0.98	**0.98**[5]	0.98	0.99	0.99[4]	0.99	0.99	0.99[6]	0.92	0.92	0.92[3]
	encoder + RForest	0.98	0.98	**0.98**[4]	0.99	1.00	**1.00**[3]	1.00	1.00	**1.00**[6]	0.97	0.97	**0.97**[4]
	SVM	0.97	0.97	0.97[2,7]	0.98	0.99	0.99[6,8]	1.00	1.00	**1.00**[4,8]	0.93	0.94	0.94[2,8]
	encoder + SVM	0.98	0.99	**0.99**[4,7]	1.00	1.00	**1.00**[3,7]	1.00	1.00	**1.00**[6,7]	0.98	0.98	**0.98**[4,7]
Unsupervised	LOF	0.36	0.37	0.36[3]	0.28	0.23	0.26[4]	0.32	0.26	0.29[5]	0.43	0.02	0.04[2]
	encoder + LOF	0.59	0.12	0.20[4]	0.36	0.18	0.24[4]	0.29	0.20	0.23[3]	0.59	0.11	0.18[1]
	IForest	0.29	0.20	0.24[1]	0.74	0.55	**0.63**[3]	0.33	0.13	0.18[6]	0.30	0.09	0.14[1]
	encoder + IForest	0.86	0.97	**0.91**[4]	0.40	0.41	0.40[6]	0.49	0.58	**0.53**[6]	0.64	0.61	**0.63**[1]
	OC-SVM	0.46	0.81	0.59[5,7]	0.41	0.92	**0.56**[4,7]	0.65	0.69	**0.67**[1,7]	0.69	0.73	**0.71**[6,7]
	encoder + OC-SVM	0.71	0.76	**0.73**[4,8]	0.90	1.00	**0.95**[4,7]	0.77	1.00	**0.87**[6,7]	0.63	0.91	**0.75**[5,7]

[1]No-scaling [2]Only mean scaling (by standard scaler) [3]Mean and deviation scaling (by standard scaler) [4]Only mean scaling (by robust scaler with respect to values between Q25 and Q75) [5]Mean and deviation scaling (by robust scaler with respect to values between Q25 and Q75) [6]Min-Max scaler [7]Linear kernel [8]RBF kernel

For unsupervised approaches, unlike in the case of SuddenD, the time-series and histogram representations work best with LOF, with high F1 scores of above 0.93. Similarly, the aggregated features with $F1 = 0.75$ work best with IForest followed by the frequency representation with $F1 = 0.64$ for SuddenD anomaly. The encoded representations surpass all non-encoded ones with this approach reaching F1 scores up to 0.98. The manual features yield good scores of above 0.89 with OC-SVM, however the time-value and histogram representations dominate with F1 score of above 0.94. The encoded representations do not improve the anomaly detection performance for this anomaly type using OC-SVM.

c: InstaD ANOMALIES

For *InstaD* observed in Table 7, almost all representations produce high F1 scores of above 0.9 with all supervised ML approaches. The time-value representation is slightly inferior to the other manual and autoencoded representations, producing 0.89 F1 score with LR, 0.91 with RForest and 0.94 with SVM. While for the previous SuddenD and SuddenR the remaining three representations yielded comparable F1 scores with all ML approaches, for InstaD anomaly, frequency domain representation is less suitable when compared to histogram, and histogram features are less suitable than the aggregated features in terms of the anomaly detection performance.

Considering unsupervised approaches, the more arbitrary the anomaly becomes, so the effect of the representation on the results. The time-value representation and histogram work best with LOF with F1 up to 0.89 while the encoded representation provides no additional benefit. The manual representations work poorly with RForests while the encoded ones yield F1 scores of up to 0.92. The aggregated features and encoded frequency domain representations work best with OC-SVM with $F1 = 0.9$ and $F1 = 0.93$, respectively.

d: SlowD ANOMALIES

For *SlowD* observed in Table 8, all representations produce high F1 scores of above 0.9 with all supervised ML approaches. The time-value representation performs best with LR yielding $F1 = 0.97$, while all time-value, aggregated and histogram features work well with RForest and SVM yielding an F1 score of above 0.97. This anomaly type is relatively more difficult to be detected using frequency representation when supervised approaches are considered.

For unsupervised approaches, no representation works well with LOF while all manual representations perform modestly with F1 scores of up to 0.71. However, in some specific cases, the encoded representation achieves higher detection performance. For instance, time-value encoded with IForest yields an F1 score of 0.91, while aggregated encoded yields an F1 score of 0.95 with OC-SVM. All encoded representations perform better with OC-SVM compared to their non-encoded counterparts.

B. PERFORMANCE ANALYSES OF ML APPROACHES

We now analyse the detection performance of the ML approaches described in Section VI on all the anomaly types proposed in Section IV. By using Tables 5, 6, 7 and 8 we perform an analysis across rows, unlike the cross-column analysis performed in Section VIII-A for data representations. While in Section VIII-A we already explained, as an example, how the LR approach works on our anomaly dataset, this section elaborates, as an exemplifying analysis, on what the

tree based ensemble learns. We selected the tree based ensemble as it is also easily explainable and tractable similar to LR. For the start, we remark the following major observations.

- For a given anomaly type, there is no major difference between the three selected supervised approaches.
- Among the unsupervised approaches, OC-SVM performs the best F1 scores, closely followed by IForest, whereas LOF typically performs the worst F1 scores.

1) SuddenD ANOMALIES

According to Table 5, the supervised models are able to detect SuddenD anomalies more accurately than the unsupervised models. All three supervised models have achieved near perfect F1 score of 0.99 on all data representations.

The tree based ensemble models, such as supervised RForest and unsupervised IForest, learn a set of trees and subsequently use a voting mechanism on the decision of each individual tree to determine the final class. A tree that is the fundamental part of the two ensemble models also learns which features are the most important ones. The feature with the highest discrimination power (weight) is situated at the root of the tree, then on the left and right nodes, as it can be exemplified in Figure 14, the next two important features are placed and the process follows until a certain stopping criterion is met. In our specific case, the trees learn the thresholds for particular values in the feature vector. For instance, depicted in Figure 14, it can be seen that if the value at position 290 in the time-value representation, denoted by X_{290}, is below -92.5, then the link is anomalous, otherwise it is a normal link. This simple rule is able to correctly detect $n = 596$ anomalous links and $n = 1520$ normal links while only misclassifying 7 links, thus the performance of that tree alone is $F1 = 0.99$.

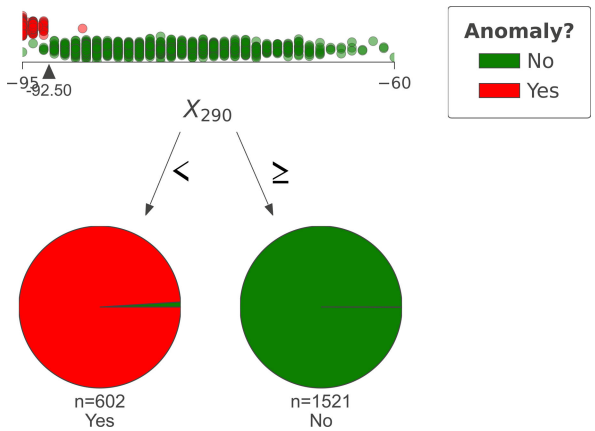


FIGURE 14. An exemplifying decision tree for the detection process of SuddenD anomaly.

The SVM models are more complex and difficult to visualize when a feature vector has more than 3 dimensions as it is the case with all manual and autoencoded representations used in this paper. SVMs essentially compute a hyperplane that attempts to separate the N-dimensional feature vector according to a criterion, such as the labels.

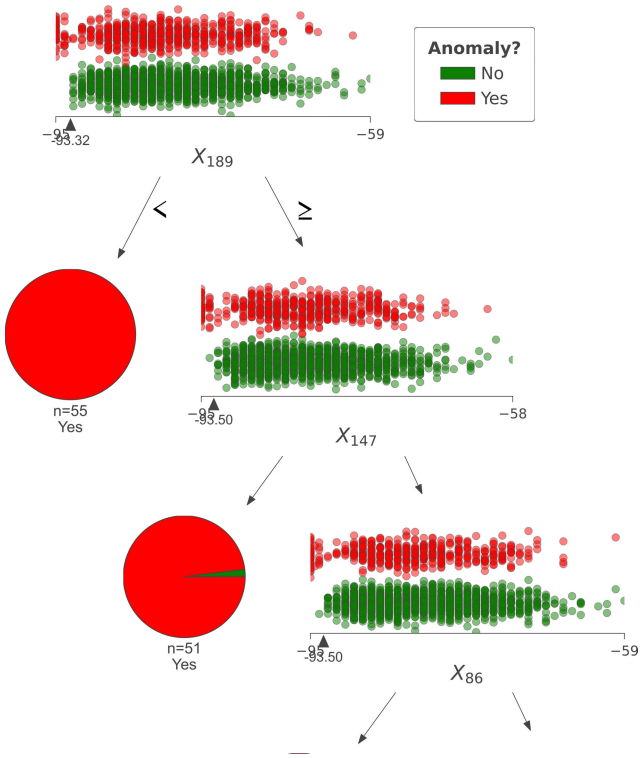


FIGURE 15. A part of the decision tree while detecting SuddenR anomaly over time-value representation.

Among the unsupervised approaches, OC-SVM is able to achieve F1 scores close to the supervised approaches, for instance 0.98, 0.96 and 0.90 on FFT, histogram and time-value representations, respectively. For OC-SVM model, with the aid of autoencoder the time-value representation is transformed to an important summary of the data by removing the noise and repetitions, leading to a performance increase from $F1 = 0.83$ to $F1 = 0.96$. Next, IForest achieved a lower performance with an F1 score between 0.61 and 0.83, the latter on the aggregated representation 0.83 while the LOF performance reached 0.76 on one occasion.

2) SuddenR ANOMALIES

Compared to SuddenD, SuddenR gains a steep recovery slope, while the duration and occurrence are more random. The results in Table 6 show that supervised models are able to detect SuddenR more accurately than the unsupervised models. F1 score of supervised models ranges from 0.89 with LR on time-value representation to near perfect F1 score for remaining supervised approaches. Using encoded representation of the time-values improves the performance also in the case of LR to 0.99, which corresponds to an about 11% improvement. For the LR case, as discussed in Section VIII-A and depicted in Figure 11, the most important features are the ones that attempt to capture the random drops between packets 25 and 275.

A decision tree representing RForest and IForest ensembles is portrayed in Figure 15 for the time-value representation of the SuddenR anomaly. It can be seen that the

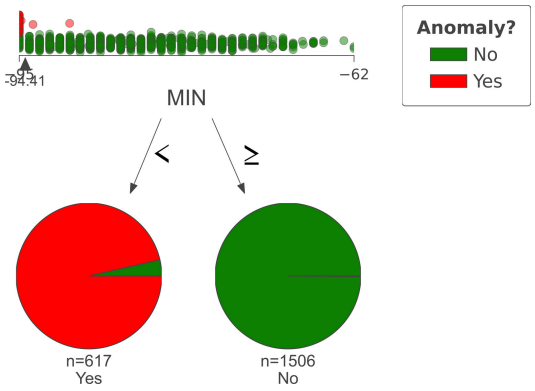


FIGURE 16. A part of the decision tree while detecting SuddenR anomaly over aggregated data representation.

most discriminative data points are X_{189}, X_{147}, X_{86} with -93.5 dBm RSSI threshold. The tree can grow very deep, eventually over-fitting the data, however, as discussed in Section VII, we undertook standard methods for avoiding that in the experimental design. Figure 16 presents an example tree learnt on aggregated feature representation. Similar to the tree in Figure 14, it is simple and effective, where it compares minimal RSSI to -94.407 dBm threshold to decide whether it is anomaly or not. Figure 17 shows a tree learnt using the histogram representation as input. While performance is similar to the previous representation, we see that using aggregated representation requires less number of decisions, i.e., depth of tree, for effective anomaly detection. Similar observations can be made for the tree learnt on fft representation for this anomaly type depicted in Figure 18.

Among the unsupervised approaches, OC-SVM, without encoded representation, is able to achieve an F1 score of around 0.90 on average through all four representations, which is almost on par with supervised approaches. IForest, on the other hand, performs much better with encoded representations, where the most significant improvement is presented on time-value representation ramping its F1 score from 0.21 to 0.86. Since SuddenR is limited in duration and thus affecting less number of features, LOF is able to pull ahead in time-value and histogram representations, where it reaches an F1 score of above 0.93.

3) InstaD ANOMALIES

In contrast to SuddenD and SuddenR, InstanD appears as an anomaly with extremely short duration (pulse). The results in Table 7 show that supervised approaches are slightly better at InstaD classification. F1 performance score of supervised approaches is slightly worse (up to 0.98) from what we have seen for SuddenD or SuddenR detection performance. Due to the arbitrary characteristics of this anomaly type, the F1 score is diminished further when the supervised approaches are

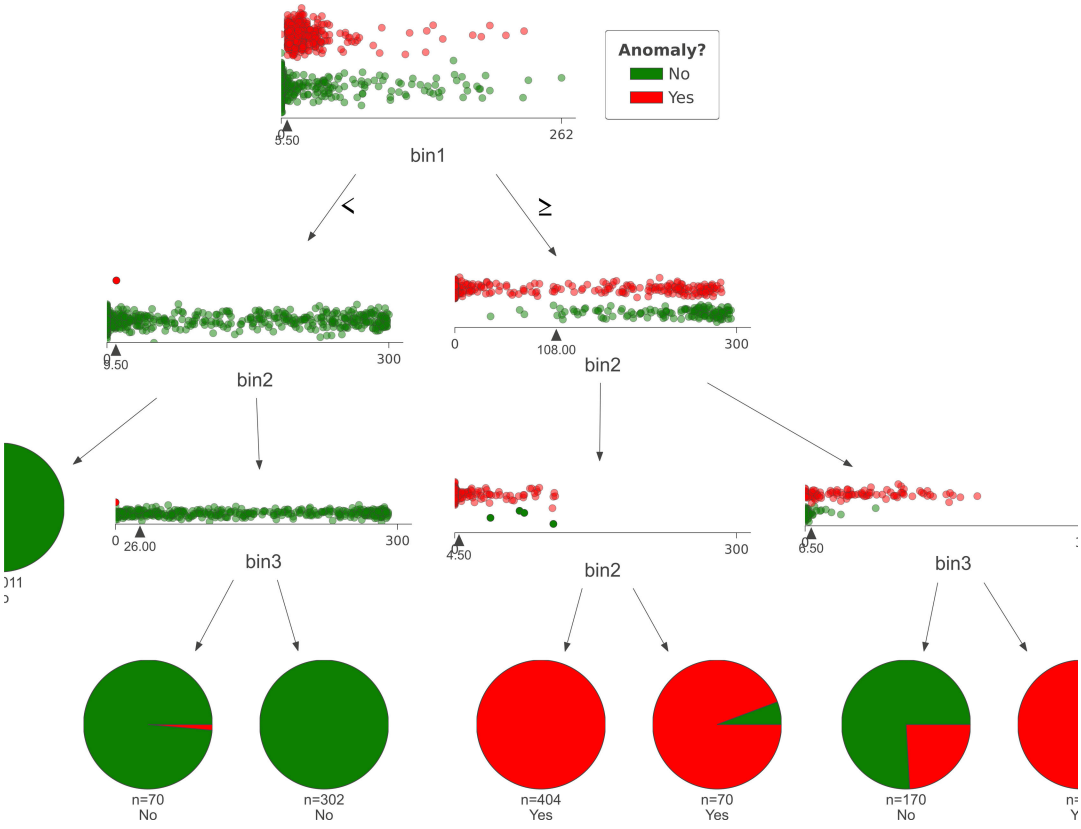


FIGURE 17. A part of the decision tree while detecting SuddenR anomaly over histogram representation.

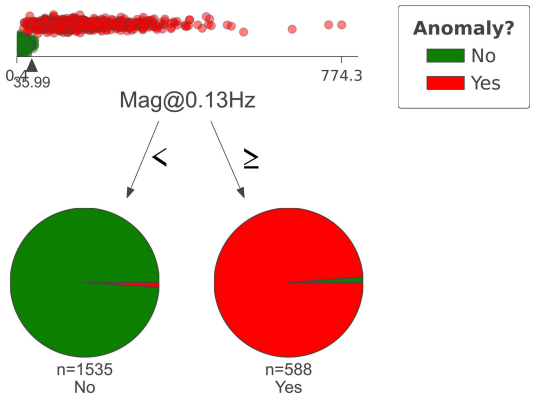


FIGURE 18. Decision tree for detecting SuddenR anomaly using FFT representation.

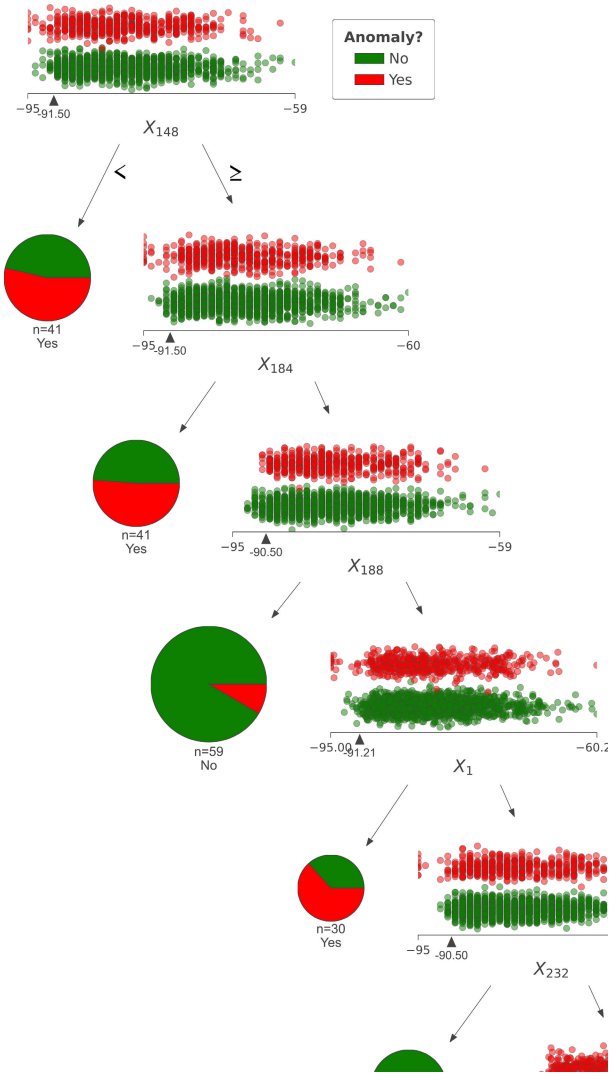


FIGURE 19. A part of the decision tree while detecting InstaD anomaly over time-value representation.

trained with the time-value and frequency domain representations as outlined in Table 7.

To better understand decision making on classifying the InstaD anomaly, we examine a decision tree representing RForest and IForest ensembles as depicted in Figure 19. Due to the random nature of this anomaly, the tree selects random points and verifies their value against a learnt threshold. For this particular tree, feature X_{148} that is compared to -91.50 dBm RSSI threshold is selected in the root. Then, it follows with the comparisons of the features in order of X_{184} and X_{188} that are compared to -91.50 dBm and -90.50 dBm, respectively and this process terminates when the final depth of the three is reached. For this anomaly type, time-series and FFT domain may not be the optimal data representations for the sake of developing a reliable and non-overfitting model.

Among the unsupervised approaches, there is no clear best approach. The top five performing models are OC-SVM using encoded FFT with 0.93 F1 score, IForest using encoded aggregated features with an F1 score of 0.92, OC-SVM using aggregated representation with an F1 score of 0.90, and LOF using histogram representation and IForest using encoded FFT, both achieving an F1 score of 0.89.

4) SlowD ANOMALIES

In contrast to SuddenD, SuddenR and InstaD, SlowD does not appear instantly, but rather gradually with random slope. The results in Table 8 show that supervised approaches are still superior to unsupervised ones. For supervised approaches, the average F1 score, ranging between 0.90 and the perfect score, is slightly better than InstaD, but slightly worse than SuddenD and SuddenR. The most notable drop in performance is observed with LR approach over aggregated, histogram and frequency representations.

To better understand the underlying reasons behind the detection performance, we visualized in Figure 20 a typical decision tree learnt on the time-value representation of this anomaly. It can be seen from the figure that the tree commences with a comparison of feature X_{282} (282nd item in time-series) to the threshold of -92 dBm. By doing so, it tries to distil anomalous samples at the end of the series, since samples with SlowD anomaly are suppose to have lower value towards the end of the time-series. However, as the first pie-chart reveals, this is not always the case, since some of the fully functioning non-anomalous (normal) links in the dataset have average RSSI close to that threshold, which leads to a high misclassification rate. In the second step of decision making, the process is repeated by comparing an earlier feature X_{64} against -89.70 dBm threshold. The tree continues to learn according to this pattern until a stopping criterion is met.

Among the unsupervised approaches, according to Table 8, the best approach is OC-SVM with best F1 scores from 0.71 to 0.95, followed by IForest with best F1 scores from 0.63 to 0.91. LOF, as an alternative unsupervised candidate, has poorly performed over all scenarios.

C. LIMITATIONS

We identify three main limitations that apply to this treatise, and to the best of our understanding also to most of the other related works in wireless network and IoT anomaly

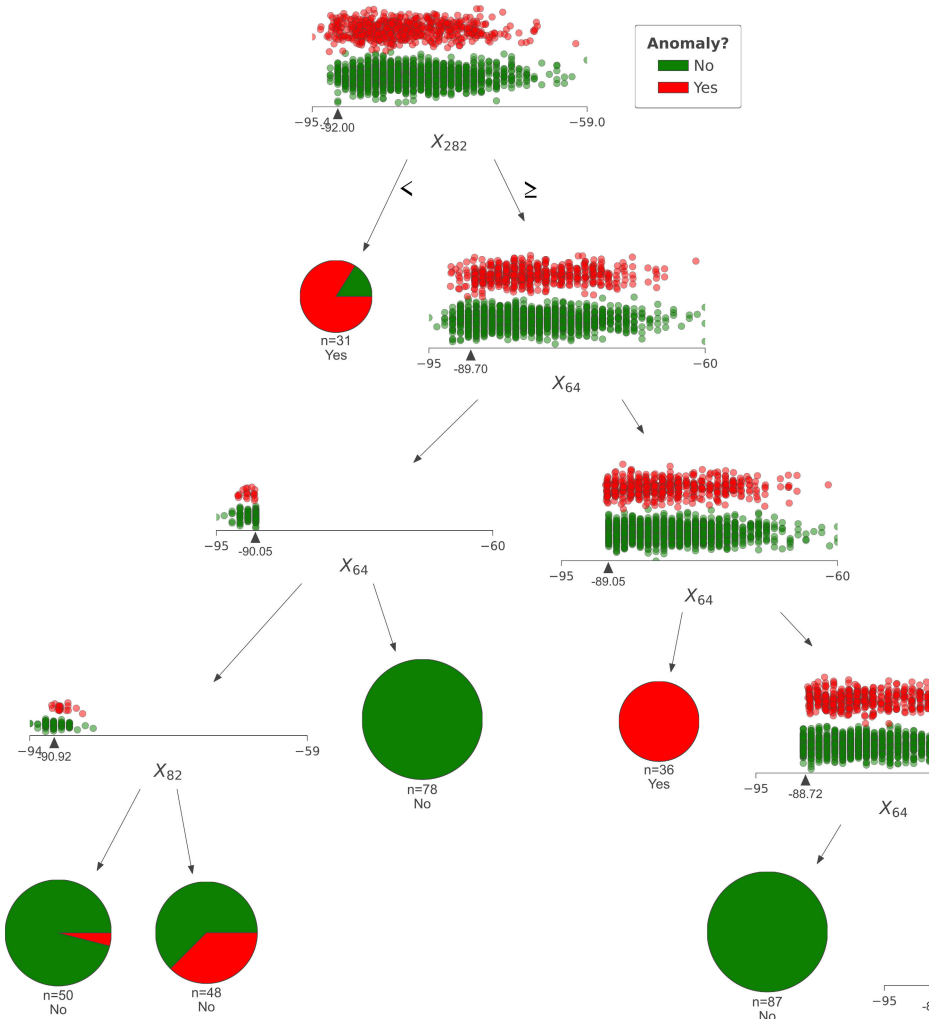


FIGURE 20. A part of the decision tree while detecting SlowD anomaly over time-value representation.

data that do not target real-world application data, such as measurements.

1) LIMITATION 1

Every ML-based tool needs sufficient data for training and evaluation. Quantifying ''sufficient'' is difficult but in general it means that the model needs to see enough training examples to be able to accurately approximate the underlying distribution. Intuition would say that the data that is ''sufficient'' to learn a normal distribution would be smaller in size than the data needed to learn an exponential distribution. While synthetic data is useful to develop a proof of concept, for anything more than that real data is required. To the best of our knowledge, only few related works consider real-world data [26] and none of them considers link layer traces.

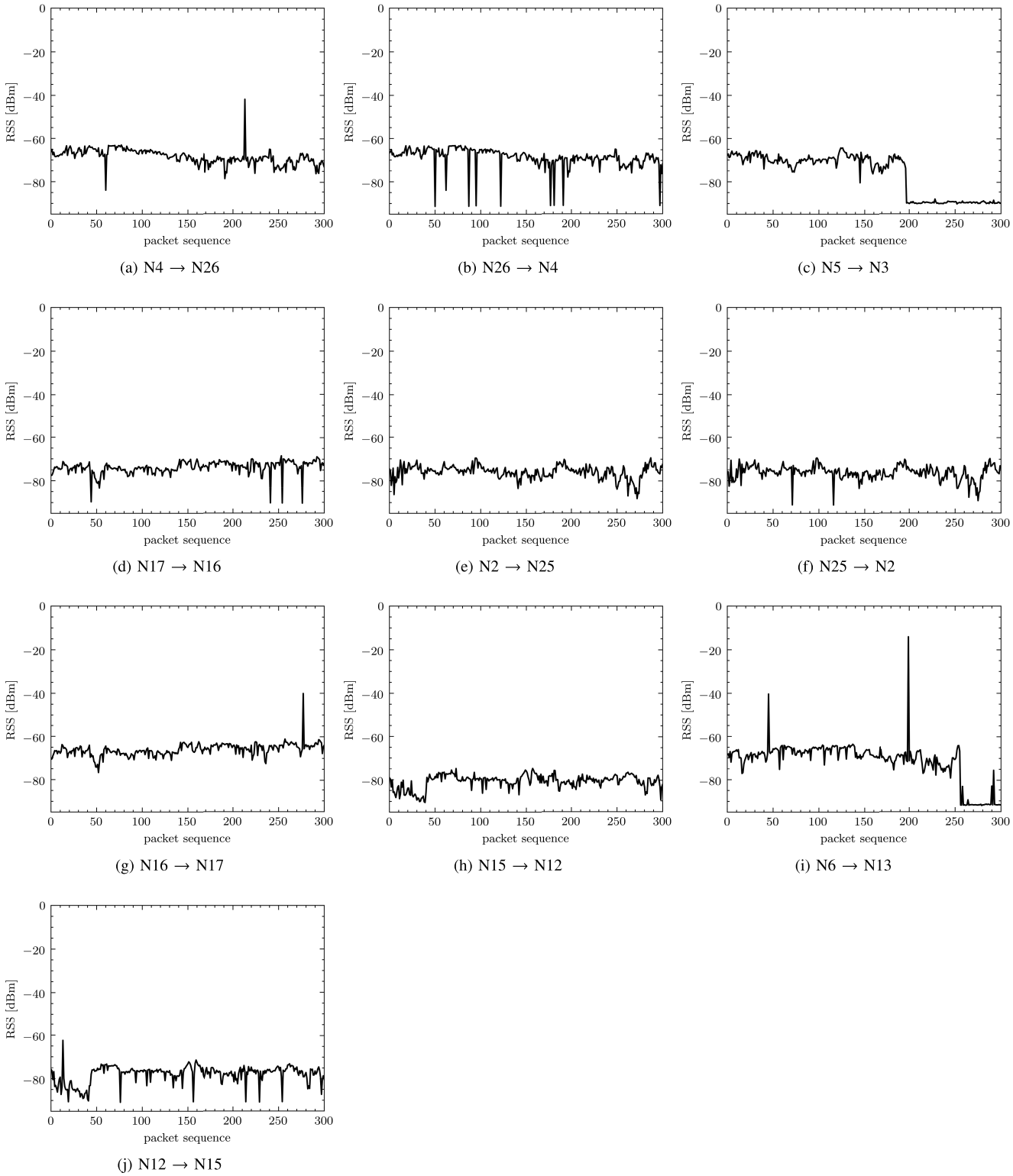
In this study, we developed the ML models using IEEE 802.11 traces available from a public dataset as the motivation data from LOG-a-TEC contains only 11 IEEE 802.15.4 traces all depicted in Figure 21. Table 9 shows how the LR model developed on IEEE 802.11 traces performs on the IEEE 802.15.4 traces. The first column of

TABLE 9. Predicted anomalies on validation data, as illustrated in Figure 21.

Model	Predicted anomalies
LR SuddenD	Figures 21c and 21i
LR SuddenR	Figures 21c and 21f
LR InstaD	Figures 21c and 21i
LR SlowD	Figures 21b, 21c, 21d, 21e, 21f and 21i

the table lists the LR model corresponding to the anomalies defined in this paper while the second includes the subfigures with links that were classified as having the respective anomalies. It can be seen from the first row of the table that the SuddenD degradations in the IEEE 802.15.4 traces are detected correctly and they appear in the links represented in Figures 21c and 21i, while for the other degradations the models seem to generate false positives.

According to the second row of Table 9, it can be seen that the links represented in Figures 21c and 21f have been classified as SuddenR. However, when visually inspecting the links in those respective subfigures it can be seen that they are both classified as false positives. It is hard to determine the reason for misclassification since none of the classified



(a) N4 → N26 (b) N26 → N4 (c) N5 → N3

(d) N17 → N16 (e) N2 → N25 (f) N25 → N2

(g) N16 → N17 (h) N15 → N12 (i) N6 → N13

(j) N12 → N15

FIGURE 21. Anomaly detection validation test employed over real-world measurements gleaned from the LOG-a-TEC testbed, where for example, as in (g) N16→N17 indicates a communication link between nodes 16 and 17.

traces even remotely resemble SuddenR anomaly presented in Figure 4a.

As per to the third row of Table 9, we observe that the two links that are detected as having InstaD anomaly are false positives. As also discussed in Section VIII-A and Figure 12a, the weights change dynamically and arbitrarily for such anomalies, and thus no distinct pattern can be readily detected.

Finally, the last row of the table shows that a large number of 802.15.4 links are falsely classified as having SlowD anomalies. While we can see that the trace in Figure 21b contains a slightly descending slope predicted to be SlowD anomaly, this model produces false positives over the other traces in Figure 21. As discussed in Section VIII-A, the discriminative importance of the features for the detection of SlowD is sought in the last part of the signal trace. This is why Figures 21c and 21i, and to some extent Figures 21e and 21f are inevitably misclassified, since they contain lower values in the last portion of the trace.

As a conclusion, the learnt models on the relatively limited IEEE 802.11 traces are not directly and reliably transferable to the IEEE 802.15.4 traces, which indicates that the developed models cannot be readily generalized across various technologies and possibly for distinct applications.

2) LIMITATION 2

The architecture of the autoencoder that learns the encoded features has been selected for a small number of candidates as a result of the trial-and-error method. Having more data would enable training an autoencoder, which then can be better generalized for even unseen examples. Autoencoder optimization and end-to-end deep learning for the proposed anomaly types might bring further insights into developing better performing and more reliable anomaly detection models. However, as hyperparameter search in deep learning is challenging and needs a large amount of training data, we leave such optimization for the future work.

3) LIMITATION 3

In this study, we only developed offline models that would need to be periodically retrained in real-world applications in order to account for the dynamically changing environments, which are the inherent characteristics of wireless networks. This leads us to online models that can learn from continuous incoming (streaming) data. Roughly speaking, offline models outperform online counterpart models in terms of the required computational power, albeit online models are able to rapidly adapt to the changes within the application environment in an automated way thus simplify the detection system that would otherwise need to periodically re-train and update the offline models.

IX. CONCLUSION

In this paper, we introduce four types of anomalies that can be present in wireless links and are useful for being detected in real-world operational IoT deployments. We demonstrated that these anomalies were exposed on a real-world IoT deployment, namely the LOG-a-TEC testbed, and they significantly affected the expected operations of the testbed. Motivated by this, we develop detection models for each type of anomaly by considering five different data representations and six different ML techniques. We performed an extensive relative evaluation of the models from data representations and ML models perspective, and the limitations of our models are discussed. The resulting tool-set for anomaly injection, feature generation and model development are made publicly available for reproducibility.

Our study reveals that *with respect to the data representations*; i) none of the four manually generated features clearly dominates the remaining ones in terms of anomaly detection performance, and ii) in most cases, automatically generated encoded data representations improve anomaly detection performance by up to 40% compared to their non-encoded counterparts.

With respect to the selected ML approach, our results demonstrate that; i) there is no major difference among the selected supervised ML approaches, where all are capable of detecting anomalies with F1 scores of above 0.98, and ii) the unsupervised approaches are also able to detect anomalies with F1 scores of, on average, about 0.90 and OC-SVM outperforms all the other unsupervised ones reaching at F1 scores of 0.99 for SuddenD, 0.95 for SuddenR, 0.93 for InstaD and 0.95 for SlowD.

ACKNOWLEDGMENT

The authors would like to recognize Tomaž Šolc, one of the core developers of the LOG-a-TEC testbed for his contribution to the motivation of this work.

REFERENCES

[1] T. Qiu, N. Chen, K. Li, M. Atiquzzaman, and W. Zhao, "How can heterogeneous Internet of Things build our future: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2011–2027, 3rd Quart., 2018.

[2] J. Davies and C. Fortuna, *The Internet of Things: From Data to Insight*. Hoboken, NJ, USA: Wiley, 2020.

[3] R. Díaz-Díaz, L. Muñoz, and D. Pérez-González, "Business model analysis of public services operating in the smart city ecosystem: The case of SmartSantander," *Future Gener. Comput. Syst.*, vol. 76, pp. 198–214, Nov. 2017.

[4] U. Wetzker, I. Splitt, M. Zimmerling, C. A. Boano, and K. Römer, "Troubleshooting wireless coexistence problems in the industrial Internet of Things," in *Proc. IEEE Int. Conf. Comput. Sci. Eng. (CSE)*, Paris, France, Aug. 2016, p. 98.

[5] J. D. C. Silva, J. J. P. Rodrigues, K. Saleem, S. A. Kozlov, and R. A. Rabêlo, "M4DN. IoT—A networks and devices management platform for Internet of Things," *IEEE Access*, vol. 7, pp. 53305–53313, Apr. 2019.

[6] A. Sheth, C. Doerr, D. Grunwald, R. Han, and D. Sicker, "MOJO: A distributed physical layer anomaly detection system for 802.11 WLANs," in *Proc. 4th Int. Conf. Mobile Syst., Appl. Services (MobiSys)*, 2006, pp. 191–204.

[7] S. Gupta, R. Zheng, and A. M. K. Cheng, "ANDES: An anomaly detection system for wireless sensor networks," in *Proc. IEEE Int. Conf. Mobile Adhoc Sensor Syst.*, Oct. 2007, pp. 1–9.

[8] H. Alipour, Y. B. Al-Nashif, P. Satam, and S. Hariri, "Wireless anomaly detection based on IEEE 802.11 behavior analysis," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 10, pp. 2158–2170, Oct. 2015.

[9] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, p. 58, Jul. 2009.

[10] M. Vucnik, T. Solc, U. Gregorc, A. Hrovat, K. Bregar, M. Smolnikar, M. Mohorcic, and C. Fortuna, "Continuous integration in wireless technology development," *IEEE Commun. Mag.*, vol. 56, no. 12, pp. 74–81, Dec. 2018.

[11] P. Gogoi, D. K. Bhattacharyya, B. Borah, and J. K. Kalita, "A survey of outlier detection methods in network anomaly identification," *Comput. J.*, vol. 54, no. 4, pp. 570–588, Apr. 2011.

[12] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Stat. Anal. Data Mining*, vol. 5, no. 5, pp. 363–387, Oct. 2012.

[13] C. C. Aggarwal, "Outlier ensembles: Position paper," *ACM SIGKDD Explor. Newslett.*, vol. 14, no. 2, pp. 49–58, Apr. 2013.

[14] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2250–2267, Sep. 2014.
[15] X. Xu, H. Liu, and M. Yao, "Recent progress of anomaly detection," *Complexity*, vol. 2019, pp. 1–11, Jan. 2019.
[16] A. A. Cook, G. Misirli, and Z. Fan, "Anomaly detection for IoT time-series data: A survey," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6481–6494, Jul. 2020.
[17] A. Lavin and S. Ahmad, "Evaluating real-time anomaly detection algorithms–the Numenta anomaly benchmark," in *Proc. IEEE 14th Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2015, pp. 38–44.
[18] R. Jurdak, X. R. Wang, O. Obst, and P. Valencia, "Wireless sensor network anomalies: Diagnosis and detection strategies," in *Intelligence-Based Systems Engineering*. Berlin, Germany: Springer, 2011, pp. 309–325, doi: 10.1007/978-3-642-17931-0_12.
[19] T. Kieu, B. Yang, C. Guo, and C. S. Jensen, "Outlier detection for time series with recurrent autoencoder ensembles," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Macao, Republic of China, Aug. 2019, pp. 2725–2732.
[20] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Unsupervised representation learning of structured radio communication signals," in *Proc. 1st Int. Workshop Sens., Process. Learn. Intell. Mach. (SPLINE)*, Jul. 2016, pp. 1–5.
[21] T. J. O'Shea, T. Erpek, and T. Charles Clancy, "Deep learning based MIMO communications," 2017, *arXiv:1707.07980*. [Online]. Available: http://arxiv.org/abs/1707.07980
[22] H. Zhang, K. Liu, Q. Shang, L. Feng, C. Chen, Z. Wu, and S. Guo, "Dual-band Wi-Fi based indoor localization via stacked denosing autoencoder," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.
[23] B. Wang, F. Hu, Y. Zhao, and T. N. Guo, "Anomaly detection and array diagnosis in wireless networks with multiple antennas: Framework, challenges and tools," *IEEE Netw.*, vol. 32, no. 1, pp. 152–159, Jan. 2018.
[24] M. R. Shahid, G. Blanc, Z. Zhang, and H. Debar, "Anomalous communications detection in IoT networks using sparse autoencoders," in *Proc. IEEE 18th Int. Symp. Netw. Comput. Appl. (NCA)*, Cambridge, MA, USA, Sep. 2019, pp. 1–5.
[25] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection," in *Proc. Wireless Telecommun. Symp. (WTS)*, Phoenix, AZ, USA, Apr. 2018, pp. 1–5.
[26] C. Yin, S. Zhang, J. Wang, and N. N. Xiong, "Anomaly detection based on convolutional recurrent autoencoder for IoT time series," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Feb. 7, 2020, doi: 10.1109/TSMC.2020.2968516.
[27] V. L. L. Thing, "IEEE 802.11 network anomaly detection and attack classification: A deep learning approach," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, San Francisco, CA, USA, Mar. 2017, pp. 1–6.
[28] J. Ran, Y. Ji, and B. Tang, "A semi-supervised learning approach to IEEE 802.11 network anomaly detection," in *Proc. IEEE 89th Veh. Technol. Conf. (VTC-Spring)*, Apr. 2019, pp. 1–5.
[29] O. Salem, A. Guerassimov, A. Mehaoua, A. Marcus, and B. Furht, "Anomaly detection in medical wireless sensor networks using SVM and linear regression models," *Int. J. E-Health Med. Commun.*, vol. 5, no. 1, pp. 20–45, Jan. 2014.
[30] O. Salem, A. Guerassimov, A. Mehaoua, A. Marcus, and B. Furht, "Sensor fault and patient anomaly detection and classification in medical wireless sensor networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Budapest, Hungary, Jun. 2013, pp. 4373–4378.
[31] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1996–2018, 4th Quart., 2014.
[32] T. Šolc, C. Fortuna, and M. Mohorčič, "Low-cost testbed development and its applications in cognitive radio prototyping," in *Cognitive Radio and Networking for Heterogeneous Wireless Networks*. New York, NY, USA: Springer, 2015, pp. 361–405.
[33] T. Šolc and Z. Padrah, "Network design for the LOG-a-TEC outdoor testbed," in *Proc. 2nd Int. Workshop Meas.-Based Experim. Res., Methodol. Tools*, 2013.
[34] J. K. Mann, S. Perinpanayagam, and I. Jennions, "Aging detection capability for switch-mode power converters," *IEEE Trans. Ind. Electron.*, vol. 63, no. 5, pp. 3216–3227, May 2016.
[35] J. Lin, S. Williamson, K. Borne, and D. DeBarr, "Pattern recognition in time series," *Adv. Mach. Learn. Data Mining Astron.*, vol. 1, nos. 617–645, p. 3, 2012.
[36] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE J.*, vol. 37, no. 2, pp. 233–243, Feb. 1991.
[37] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[38] I. Alawe, A. Ksentini, Y. Hadjadj-Aoul, and P. Bertin, "Improving traffic forecasting for 5G core network scalability: A machine learning approach," *IEEE Netw.*, vol. 32, no. 6, pp. 42–49, Nov. 2018.
[39] R. B. D'Agostino, "An omnibus test of normality for moderate and large size samples," *Biometrika*, vol. 58, no. 2, pp. 341–348, 1971.
[40] R. D'Agostino and E. S. Pearson, "Tests for departure from normality. empirical results for the distributions of b^2 and $\sqrt{b^1}$," *Biometrika*, vol. 60, no. 3, pp. 613–622, 1973.
[41] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, Oct. 2004.
[42] R. Malouf, "A comparison of algorithms for maximum entropy parameter estimation," in *Proc. 6th Conf. Natural Lang. Learn.*, vol. 20. Stroudsburg, PA, USA: Association Computational Linguistics, 2002, pp. 1–7, doi: 10.3115/1118853.1118871.
[43] C. C. Aggarwal, "Outlier analysis," in *Data Mining*. New York, NY, USA: Springer, 2015, pp. 237–263.
[44] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[45] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer, 2013.
[46] L. Lin and Z. Xiaolong, "Optimization of SVM with RBF Kernel," *Comput. Eng. Appl.*, vol. 42, no. 29, pp. 190–192 and 204, 2006. [Online]. Available: https://jglobal.jst.go.jp/en/detail?JGLOBAL_ID=200902220640336864
[47] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," *SIGMOD Rec.*, vol. 29, no. 2, p. 93–104, May 2000, doi: 10.1145/335191.335388.
[48] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 413–422.
[49] S. K. Kaul, I. Seskar, and M. Gruteser. (Apr. 2007). *CRAWDAD Dataset Rutgers/Noise (v. 2007-04-20)*. [Online]. Available: https://crawdad.org/rutgers/noise/20070420/RSSI
[50] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, 2013, vol. 30, no. 1, p. 3.

GREGOR CERAR (Graduate Student Member, IEEE) received the master's degree in telecommunications from the Faculty of Electrical Engineering, University of Ljubljana, in 2016. He is currently pursuing the Ph.D. degree with the Jožef Stefan International Postgraduate School. He is also a Research Assistant with the Department of Communication Systems, Jožef Stefan Institute. His main research interests include the IoT, wireless networking of constrained devices, and machine learning applications in IoT.

HALIL YETGIN (Member, IEEE) received the B.Eng. degree in computer engineering from Selcuk University, Turkey, in 2008, the M.Sc. degree in wireless communications from the University of Southampton, U.K., in 2010, and the Ph.D. degree in wireless communications from the Next Generation Wireless Research Group, University of Southampton, in 2015. He is currently an Assistant Professor with the Department of Electrical and Electronics Engineering, Bitlis Eren University, Turkey, and a Research Fellow with the Department of Communication Systems, Jožef Stefan Institute, Ljubljana, Slovenia. His research interests include the development of intelligent communication systems, energy efficient cross-layer design, and resource allocation of the future wireless communication networks. He was a recipient of the full scholarship granted by the Republic of Turkey, Ministry of National Education.

BLAZ BERTALANIC (Member, IEEE) received the master's degree in electrical engineering from the Faculty of Electrical Engineering, University of Ljubljana, in 2020, where he is currently pursuing the Ph.D. degree. He is also a Junior Researcher with the Department of Communication Systems, Jožef Stefan Institute. His main research interests include solving classification problems with the help of machine learning, wireless networking, electronics, and signal processing.

CAROLINA FORTUNA received the B.Sc. degree, in 2006, and the Ph.D. degree, in 2013. She was a Postdoctoral Research Associate with IBCN, Ghent University, from 2014 to 2015. She is currently a Research Fellow with the Department of Communication Systems, Jožef Stefan Institute, and an Assistant with the Jožef Stefan International Postgraduate School. Her research interests include interdisciplinary, focusing on data and knowledge driven modeling of communication and sensor systems. She has participated in H2020, FP7, and FP6 projects. In H2020 WiSHFUL, she was the Technical Leader of the project on behalf of UGhent/iMinds while in FP7 CREW she was the Technical Leader of the JSI Team. She has coauthored more than 50 peer-reviewed publications, was a TPC Member at IEEE ICC 2011, 2012, 2013, 2014, 2016, ESWC 2012, IEEE GLOBECOM 2011, 2016, VTC 2010, 2016, and IEEE WCNC 2009.