Interpretable Outlier and Anomaly Detection for Mobile Networks from Small Tabular Data

Juan Marcos Ramirez
IMDEA Networks Institute
Leganés, Madrid, Spain
juan.ramirez@imdea.org

Pablo Rojo
Nokia MN
Madrid, Spain
pablo.rojo@nokia.com

Vincenzo Mancuso
University of Palermo
IMDEA Networks Institute
vincenzo.mancuso@imdea.org

Antonio Fernández Anta
IMDEA Software Institute
IMDEA Networks Institute
antonio.fernandez@imdea.org

***Abstract*—Outliers and anomalies in mobile networks refer to significant deviations of the Key Performance Indicator (KPI) from expected values, often degrading user experience. Therefore, detecting and understanding these atypical events is crucial for troubleshooting. To monitor network performance, operators continuously collect data using different testing strategies. One such strategy involves drive-tests that capture datasets with many parameters but limited sample sizes, rendering them unsuitable for deep learning approaches, which require large datasets for effective learning. This paper proposes ROAD (Interpretable Outlier and Anomaly Detection), an unsupervised machine learning methodology designed to detect and understand atypical operational scenarios in mobile networks from drive-tests. This methodology comprises an unsupervised detection stage and introduces an interpretability module that is applied separately to outliers and anomalies. The interpretability module identifies variables and samples associated with atypical events, quantifies the degree of similarity between each variable and the anomaly pattern, and builds a decision tree to reveal the ranges of variables describing atypical scenarios. We implemented the methodology in software and evaluated its performance using real drive-test data. Our method provides high accuracy in detecting outliers and anomalies separately, while reducing the identification of false positives (recall) between 39% and 63% compared to an existing explainable detection method.**

***Index Terms*—anomaly detection, drive-test data, interpretable machine learning, mobile networks, outlier detection**

I. Introduction

Mobile networks offer high-speed connectivity and unprecedented data transfer rates, particularly in 5G systems [2]. However, the complexity of network infrastructures poses challenges in identifying events associated with performance degradation and service disruptions. Mitigating these shortcomings is crucial to ensure service reliability and improve quality of experience (QoE). To address these challenges, operators continuously collect data on various network operational aspects. However, datasets typically involve many variables and limited sample sizes, complicating the application of deep learning-based anomaly detection methods.

Extracting information from such data is challenging in many senses. Firstly, high-dimensional data can lead to overfitting in models, reducing their generalizability to new datasets [11]. Secondly, identifying relevant features in anomaly detection becomes difficult as the input features increase, making detection models less interpretable. In addition, redundant features can cause inaccuracies in detection tasks, requiring feature selection techniques to improve detection performance [18]. Lastly, limited sample sizes can result in class imbalance, where the number of anomalous examples is extremely low compared to regular instances [33].

This paper has been funded by project PID2022-140560OB-I00 (DRONAC) funded by MICIU/AEI /10.13039/501100011033 and ERDF, EU.

A. Performance Outliers and Anomalies

This work focuses on detecting and understanding two types of *atypical events*: outliers and anomalies. We assume that data characterizing normal network behavior can be extracted through data analysis and machine learning techniques.

- **Outliers:** refer to events whose KPI values fall outside the boundary established by the normal data. Such outliers may correspond to scenarios characterized by atypical low throughput (often several orders of magnitude below the expected performance), excessively long session durations, and unusually high packet loss rates. These deviations typically indicate severe performance degradation or malfunctioning of network components.
- **Anomalies:** we define *anomalies* as scenarios that, although their KPI values are within the limits defined by the normal data, exhibit lower performance than that predicted by a machine learning model. This model, referred to as the *normality model*, captures the expected behavior of the network. We use tree-based regressors to build normality models. In mobile networks, anomalies are often linked to underlying suboptimal operational conditions, such as unexpected delays or inadequate resource allocation, which cause the actual performance to deviate from the expected behavior, even when KPI values appear to be within normal ranges.

B. Contributions

This paper introduces the ROAD (Interpretable Outlier and Anomaly Detection) methodology that identifies and characterizes outliers and anomalies in mobile networks from drive-test data. This approach consists of three key stages. First, the *data preprocessing* phase enhances data quality by handling inconsistencies and missing values. Second, *unsupervised outlier and anomaly detection* identifies atypical events without requiring prior knowledge of the proportions of atypical scenarios. Third, *interpretability modules* that improve the understanding by identifying patterns in input variables strongly associated with anomalous events. These modules also compute indices that quantify the relationship between each variable and anomalous patterns and build decision

trees to extract variable ranges closely related to outliers and anomalies. Our contributions are listed as follows:

- In the context of mobile networks, we define and differentiate outliers and anomalies while introducing an interpretable methodology for their detection and characterization.
- The proposed methodology integrates three key stages: data preprocessing, outlier and anomaly detection, and interpretability.
- The detection stage identifies atypical scenarios without requiring prior information on outlier and anomaly rates, thereby providing adaptability across different network conditions.
- The interpretability module identifies features and samples related to atypical events. This module computes the ROAD index, quantifying the relationship between atypical samples and patterns extracted from input variables. It also builds a decision tree to describe variable intervals related to atypical events, thereby providing valuable understanding into network behavior.
- We evaluate the proposed methodology on real datasets, exhibiting a competitive performance compared to state-of-the-art approaches. Our approach provides ranking and visualization tools for interpretable insights into network performance, facilitating the diagnosis for troubleshooting.

The methodology has been implemented in software and its performance has been evaluated using real drive-test data. The datasets consist of small-scale tabular data with a few hundred features and approximately the same number of samples.

II. Related Work

Various methods have been developed to detect atypical events in mobile networks from data. Sangaiah et al. [29] introduced a method to detect and diagnose faults in cellular networks using performance support system data, drive-test data, and customer service data. In their approach, performance support system data is the primary source for detecting and diagnosing network failures. In contrast, drive test data is utilized as a secondary source to identify failures across three call scenarios: short, long, and idle. Notice that their approach does not specifically address atypical performance scenarios crucial for maintaining optimal network operations. In addition, their methodology underutilizes the extensive information available in drive tests, which could offer more detailed insights into network performance anomalies. In contrast, our method leverages the full potential of drive-test data, which contain rich information on various network aspects, such as TCP performance and radio parameter measurements, allowing for a more comprehensive analysis of outliers and anomalies.

Shayea et al. reported a performance analysis of mobile broadband (MBB) networks in urban areas in Malaysia using drive-test data [31]. Their research compared 3G and 4G networks through various video-streaming tests, highlighting differences between these two generations. However, their method did not detect anomalies and relied on manually extracting a small set of variables. Conversely, our method is designed to identify atypical scenarios and to explain the combination of features leading to performance degradation by leveraging the rich information in drive-test data.

TABLE I
Summary of the characteristics of each dataset used to evaluate the performance of the proposed methodology.

Dataset	Subset	Test Type	Rows	Cols
Dataset 1	Subset 1A	`Capacity DL`	1001	189
	Subset 1B	`Capacity UL`	971	161
	Subset 1C	`HTTP Transfer DL`	709	188
	Subset 1D	`HTTP Transfer UL`	688	166
Dataset 2	Subset 2A	`HTTP File DL`	2095	119
	Subset 2B	`HTTP File UL`	1985	102

Kim et al. proposed anomaly detection in 4G-LTE networks by analyzing network log data [13]. This method extracts the normal data applying principal component analysis (PCA). However, PCA severely decreases its performance as the number of features increases, preventing the identification of problematic variables. In contrast, our method builds a normality model and an interpretability stage using input features in the original domain to understand the relevant variables that describe anomalous scenarios.

In the context of explainable anomaly detection in mobile networks, Moulay et al. proposed in [24] a supervised method to detect faults in cellular networks from drive-test data. They also reported unsupervised versions in [22], [23]. Unsupervised and explainable methodologies to detect and classify performance outliers in mobile networks from drive-test data were introduced in [27], [28]. These methods used classification and discretization to describe normal network behavior, reducing the detection accuracy. In contrast, we build the normality model using regression models to predict the KPI values and accurately estimate the errors between actual KPI values and predictions. Our approach also includes a method to detect problematic variables and samples that considers the closeness between anomaly patterns and feature clustering.

III. Datasets

To evaluate the performance of the proposed methodology in practical scenarios, we use two datasets (Dataset 1 and Dataset 2) captured by Nokia during drive-test campaigns across European mobile networks. Table I summarizes the main characteristics of the collected datasets. In this work, we consider six types of drive-test: capacity download (`Capacity DL`), capacity upload (`Capacity UL`), HTTP transfer download (`HTTP Transfer DL`), HTTP transfer upload (`HTTP Transfer UL`), HTTP file download (`HTTP File DL`), and HTTP file upload (`HTTP File UL`). For example, `Capacity DL` records information on the network response when a testing mobile device downloads a file from a remote server. Each row in the datasets corresponds to an operational scenario capturing different network aspects, including spatiotemporal metadata (e.g., geolocation),

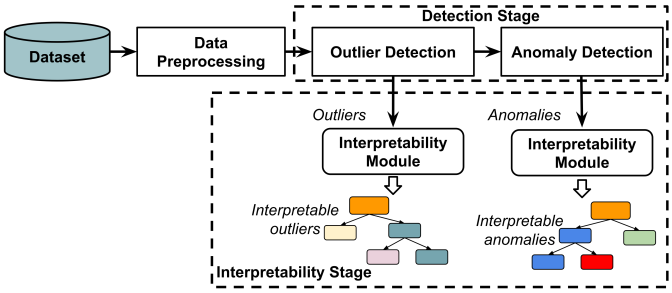


Fig. 1. Flowchart of the interpretable outlier and anomaly detection (ROAD) methodology.

network configuration (e.g., server settings), quality metrics (e.g., packet loss), and the key performance indicator (KPI). Note that information on mobile network operators and other sensitive data has been excluded from the datasets. Section XI addresses ethical issues associated with the collected datasets.

1) Dataset 1: This dataset was collected in December 2022. As shown in Table I, Dataset 1 has been segmented into four subsets (1A-1D), with each corresponding to a specific test type. Unless stated otherwise, we use Subset 1A, which includes 1001 experiments and 189 features, to illustrate the effectiveness of the proposed methodology.

2) Dataset 2: This dataset, captured in 2019, has been divided into Subset 2A and Subset 2B.

IV. Methodology Overview

Fig. 1 illustrates the flowchart of the Interpretable Outlier and Anomaly Detection (ROAD). The core idea of the proposed approach is to provide a tool that identifies instances where KPI values significantly deviate from normal behavior (outliers) and scenarios that, while having KPI values within the normal range, show significant deviations from those predicted by a learning model (anomalies). Our approach incorporates interpretability modules that recognize features and instances associated with detected outliers or anomalies.

The proposed methodology first imports drive-test data as input, typically organized as a small tabular dataset containing many performance indicators (features, columns, or variables) and a few hundred operational scenarios (rows, samples, or experiments). The limited sample size in these datasets renders them unsuitable for deep learning methods, which generally require large-scale training data [22], [23]. Also, note that input data typically lacks ground truth labels, highlighting the need for developing an unsupervised strategy. Our framework comprises a three-stage pipeline: (1) data preprocessing, which filters uninformative variables and ensures data quality; (2) outlier and anomaly detection, which identifies deviations from expected behavior; and (3) interpretability, which provides understandable insights into the detected outliers and anomalies. Each phase is detailed as follows.

V. Data Preprocessing

This stage aims to identify and remove uninformative and redundant variables. It comprises three subphases: data cleaning, detection and removal of multicollinear columns, and detection and removal of features highly correlated with the KPI. Each step is outlined below.

1) Data Cleaning: This subphase focuses on detecting and removing features with null values, often present in drive-test data due to sampling errors or failures in collecting variables during the test. To this end, our approach identifies and removes any variables exceeding a predefined null-value threshold (1% of samples). For the retained features, the process replaces the remaining null values with the median of the respective column. Finally, the procedure eliminates non-informative features by removing variables with zero variance, i.e., constant values across all samples.

2) Multicollinarity Detection and Removal: Regression models are susceptible to multicollinearity because of high correlations between variables. Multicollinearity introduces redundant information into the model, leading to performance degradation and reduced interpretability [35]. In our case, multicollinearity may occur due to the small sample size and the large number of features in the collected data. To address this issue, we use the variance inflation factor (VIF) to identify features that exhibit strong multicollinearity. VIF is computed by regressing each variable against the others and computing $\text{VIF} = 1/(1 - R^2)$, where R^2 is the coefficient of determination of the regression model. In addition, VIF thresholds exceeding ten typically indicate significant multicollinearity [11]. This work employs an iterative procedure to remove multicollinear variables. The pseudocode is outlined in Algorithm 1.

Algorithm 1: Multicollinearity Detection and Removal

Data: Input dataset $\mathbf{X}_0$, VIF_threshold
Result: Output dataset $\mathbf{X}_1$
$k \leftarrow 0$;
$\mathbf{X}_{0,k} \leftarrow \mathbf{X}_0$;
maximum_VIF $\leftarrow \max(\text{VIF}(\mathbf{X}_{0,k}))$;
while maximum_VIF $\geq$ VIF_threshold **do**
  feature_to_remove $\leftarrow \arg\max_{\text{feature}} \text{VIF}(\mathbf{X}_{0,k})$;
  $\mathbf{X}_{0,k+1} \leftarrow$ remove_feature$(\mathbf{X}_{0,k}$, feature_to_remove$)$;
  maximum_VIF $\leftarrow \max(\text{VIF}(\mathbf{X}_{0,k}))$;
  $k \leftarrow k + 1$;
end
$\mathbf{X}_1 \leftarrow \mathbf{X}_{0,k}$

Initially, the algorithm takes the dataset produced by the data preprocessing stage, denoted as $\mathbf{X}_0$. During each iteration, the algorithm identifies the feature exhibiting the highest VIF and removes the respective column from the dataset. The procedure recalculates VIF values of the updated dataset. The process continues until the maximum VIF is below a predefined threshold. We set this threshold at 1000, targeting features with high multicollinearity. Once the stopping criterion is satisfied, the method retrieves a new dataset, $\mathbf{X}_1$.

3) KPI-correlated Detection and Removal: Features highly correlated with the KPI often lead to overfitting in machine learning models. These features yield biased and uninformative outcomes, such as variable rankings that solely include highly correlated features [5]. Therefore, this subphase recognizes and removes features highly correlated with the

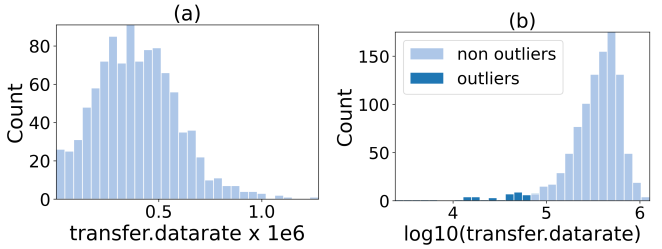


Fig. 2. Subset 1A. (a) Histogram of the target KPI. (b) Histogram of the logarithmic version of the target KPI.

KPI. To this end, we compute a vector of correlation factors between each feature and the KPI. In this regard, let $\mathbf{X}_1 = [\mathbf{x}_1^\top, \ldots, \mathbf{x}_{n_1}^\top]^\top$ be the matrix containing the input features with dimensions $m_1 \times n_1$, where $\mathbf{x}_i = [x_{(1,i)}, \ldots, x_{(m_1,i)}]^\top$ for $i = 1, \ldots, n_1$ are the m_1-dimensional vectors describing input features, and $\mathbf{y}_1 = [y_1, \ldots, y_{m_1}]^\top$ the m_1-dimensional vector of KPI samples. The vector of correlation factors is defined as $\mathbf{c}_1 = [c_1, \ldots, c_{n_1}]^\top$, where each component is computed as

$$c_i = 1 - \left| \frac{\mathbf{cov}(\mathbf{x}_i, \mathbf{y}_1)}{\sigma_{\mathbf{x}_i} \sigma_{\mathbf{y}_1}} \right| \tag{1}$$

for $i = 1, \ldots, n_1$, where $\mathbf{cov}(\mathbf{x}_i, \mathbf{y}_1)$ denotes the covariance between the i-th input feature and the KPI, $\sigma_{\mathbf{x}_i}$ and $\sigma_{\mathbf{y}_1}$ represent the standard deviation of the i-th feature and the KPI vector, respectively. Each correlation factor c_i can be defined as one minus the magnitude of the Pearson correlation coefficient between the i-th feature and the KPI.

To identify features highly correlated with the KPI, we apply a model-based clustering to the vector of correlation factors [21]. However, this clustering method has disadvantages, particularly in selecting the optimal number of clusters and managing random initializations. To determine the optimal number of clusters, we perform a Monte Carlo simulation based on the Bayesian Information Criterion (BIC). Our approach generates multiple BIC curves with varying numbers of clusters, ranging from one to the length of the vector of correlation factors, n_1, where each point of the curves is obtained using a different random initialization. We then calculate the average BIC curve and determine the optimal number of clusters as the point that minimizes this average curve. Next, we carry out model-based clustering using the optimal number of clusters and identify the cluster with the lowest centroid value, i.e., the cluster closest to zero. Finally, we remove the columns of the identified cluster.

VI. Outlier Detection in the Target KPI

Performance outliers refer to operational scenarios where the target KPI significantly deviates from its typical values. Our approach directly analyzes deviations in the KPI to detect outliers. In this study, we select throughput (measured in kbps) as the KPI. Detecting samples with significant performance loss is challenging due to the presence of atypical values embedded within the main lobe of the KPI distribution, making it difficult to distinguish normal from atypical. Fig. 2(a) shows the histogram of the KPI for Subset 1A, where the dominant distribution masks low-performance samples. We apply a logarithmic transformation to the KPI to enhance the separation between normal and atypical instances.

Subsequently, we employ an outlier detection method based on the median absolute deviation (MAD), a robust statistical approach that provides a reliable criterion for identifying outliers [15]. Let $\mathbf{l_y} = [l_{y_1}, \ldots, l_{y_{m_1}}]^\top$ represent the logarithmic version of the KPI, where each element is defined as $l_{y_i} = \log_{10}(y_i)$. An entry is classified as atypical if it falls outside the range:

$$\text{median}(\mathbf{l_y}) - 3 \cdot \text{MAD} < l_{y_i} < \text{median}(\mathbf{l_y}) + 3 \cdot \text{MAD} \tag{2}$$

for $i = 1, \ldots, m_1$, where MAD is given by

$$\text{MAD} = \frac{1}{Q_3} \text{median}\left(|\mathbf{l_y} - \text{median}(\mathbf{l_y})| \right), \tag{3}$$

and Q_3 is the third quartile of the vector $\mathbf{l_y}$ [10]. Fig 2(b) depicts the histogram of the logarithmic version of the KPI, highlighting the atypical samples. As seen, low-performance samples are clearly located at the tail of the distribution, allowing for more reliable detection.

VII. Anomaly Detection

Performance anomalies arise when KPI values fall within the expected interval but actual performance deviates from machine-learning predictions. This stage focuses on building a regression model to describe normal network behavior. Our approach then includes an anomaly detection step based on the differences between actual and predicted KPI values.

1) Outlier Detection in the Feature Space: To characterize normal behavior, we apply multivariate outlier detection in the space defined by input data. Given the complex interdependencies among these features, we utilize a one-class support vector machine (OCSVM), an unsupervised method for detecting anomalies in high-dimensional data [30]. The OCSVM is particularly helpful because it models the distribution of normal observations in a multivariate space, identifying samples that deviate from the learned boundary. To capture non-linear relationships between features, we set the OCSVM with a polynomial kernel of degree $d = 3$. This approach assumes that non-detected outliers represent the normal network behavior.

2) Normality Model Building: Previous anomaly detection methods in mobile networks rely on decision trees to build normality models. These approaches typically involve discretizing the KPI to generate class labels [22], [27], [28]. However, decision trees are prone to performance degradation when handling high-dimensional data, and the discretization process introduces quantization errors, leading to discrepancies between actual KPI values and their predictions.

We propose a regression-based normality model that eliminates the need for discretization and enables continuous KPI predictions, offering a more accurate representation of normal network behavior. Specifically, we employ gradient-boosting regression models, providing high predictive accuracy and interpretability. We implement two state-of-the-art regressors: extreme gradient boosting (XGBoost) [6] and light gradient

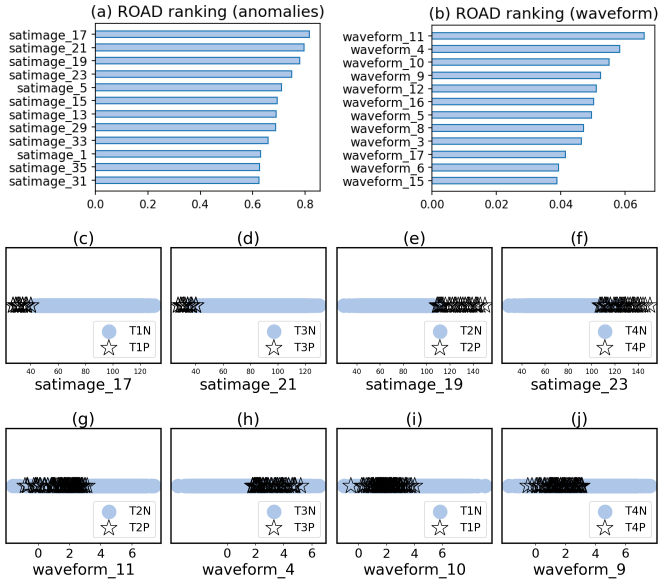


Fig. 3. Satimage-2: (a) ROAD-based ranking, (c)-(f) anomalous (black stars) and non-anomalous (blue circle) scenarios across entire feature interval for the top four features. Waveform: (b) ROAD-based ranking, (g)-(j) anomalous and non-anomalous scenarios across entire feature interval for the top four features.

boosting machine (LGBM) [12]. Our approach selects only non-outlier samples (identified in outlier detection in the feature space) to ensure a reliable training set. In this setup, the KPI is the regression response, while features act as predictors. Once the model is optimized, we apply interpretability techniques, such as feature importance or Shapley values, to identify the most influential variables driving normal behavior. This framework enables engineers to gain valuable monitoring and optimization insights. Finally, the methodology applies the regression model to the entire dataset and computes the vector of differences between the predicted and actual KPI values.

3) Anomaly Detection: To detect performance anomalies, we apply the MAD-based outlier detection method (Section VI) to the vector of differences.

VIII. Interpretability Module

The interpretability module aims to identify the features and samples most strongly associated with atypical events (outliers or anomalies). As illustrated in Fig. 1, this module operates separately for outliers and anomalies. The module begins by applying a series of one-dimensional (1D) model-based clusterings to each input feature, where a separate clustering is performed for each specified number of clusters. For a given number of clusters, n_c, the method generates n_c binary patterns. For each binary pattern, the samples in the cluster i, for $i = 1, \ldots, n_c$, are labeled as 1 (atypical samples) and the samples belonging to the remaining clusters are labeled as 0 (non-atypical samples). The number of clusters ranges from two to a maximum value defined as $\texttt{max_cluster} = \lceil \log_2(n_x) \rceil$, where n_x represents the number of samples in the dataset, and $\lceil x \rceil$ denotes the ceiling function, which returns the smallest integer greater than or equal to x.

TABLE II
Subset 1A. Dimensions of datasets extracted by different data preprocessing subphases.

Data preprocessing step	Rows	Columns
Original dataset	1001	189
Null cell column detection and removal	1001	180
Constant column detection and removal	1001	180
Multicollinearity detection and removal	1001	133
KPI-correlated detection and removal	1001	120

The methodology then evaluates the Jaccard index between binary patterns and detected atypical samples. The Jaccard index quantifies the similarity between two binary arrays, where higher values indicate greater similarity. We introduce the ROAD index as the maximum Jaccard index computed across binary patterns for a given feature. We can identify features closely associated with anomalous behavior by ranking them in descending order based on their ROAD index. The ROAD index ranges from 0 to 1, where values close to 1 suggest that the feature is strongly related to the detected atypical pattern. Existing approaches identify problematic features and samples by calculating the average of the differences between discretized KPI labels and classifier predictions [22], [27], [28]. However, this procedure typically masks localized patterns closely associated with anomalies. The ROAD index uses the Jaccard measure instead of the average. Note that the Jaccard index captures structural similarities between binary patterns. By focusing on the similarity between anomalies and feature patterns, our approach provides a targeted method for accurately identifying the features that significantly contribute to network anomalies.

For illustrative purposes, Fig. 3 shows the results for two multivariate anomaly detection datasets: satimage-2 and waveform [8]. In both cases, the ground truth labels serve as reference atypical events. Fig. 3(a) illustrates the ranking of the satimage-2 data set, where the four top features show indices that exceed 0.7, indicating a strong association with anomalies. Figs 3(c)-(f) show in 1D the locations of anomalies (black stars) and non-anomalies (blue circles) across these top features. In this case, we can identify the feature intervals closely associated with anomalies.

Fig.3(b) shows the ROAD-based feature ranking for the waveform dataset, where the top four features exhibit indices below 0.1. Figs. 3(g)-(j) illustrate in 1D the distribution of anomalies (black stars) and non-anomalous samples (blue circles) across these features. Despite these relatively low indices, we can observe feature intervals strongly associated with anomalous samples. This observation highlights an advantage of the ROAD-based ranking: even for small ROAD indices, this approach can identify feature ranges that differentiate anomalous from non-anomalous scenarios.

Finally, we build an explainable classifier to determine the features and intervals associated with atypical scenarios. The methodology selects a set of the top features according to the ROAD-based ranking. The number of selected features is defined $n_f = \lceil \log_2(n_2)/2 \rceil$. Moreover, we consider that

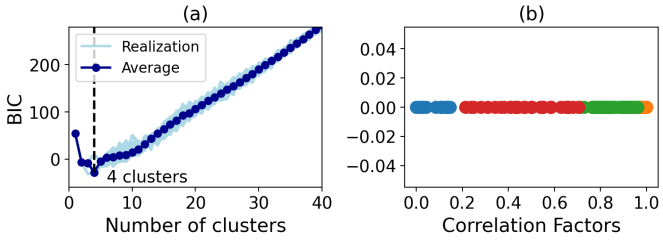


Fig. 4. Subset 1A. (a) BIC curves for 50 random realizations and the BIC average curve. The number of clusters minimizing the average curve. (b) Clustering of the correlation factors in the $[0, 1]$ range.

a one-valued sample in a binary pattern corresponds to a problematic scenario with label **T$P**, where **$** is the assigned feature ID. A zero-valued sample in a binary pattern is labeled non-problematic with **T$N**. The output classes are built concatenating labels of problematic and non-problematic samples across samples [27]. We use a decision tree classifier to train the anomaly classification model. We selected a decision tree because it is intuitive and easy to understand. The procedure uses the classification and regression tree (CART) algorithm to train the classifier and use the Gini impurity index as the cost function.

IX. Methodology in Action

This section presents the results across key stages of the methodology. ROAD is implemented in software using Python, equipped with the NumPy and Pandas libraries. We utilize Scikit-Learn for machine-learning tasks and XGBoost for regression models. The source code and datasets can be downloaded from the link: https://github.com/JuanMarcosRamirez/road. All experiments were conducted on a standard hardware: a Dell Inspiron 14 7000 laptop, 11th-gen Intel Core i7 (2.80 GHz), 16 GB DDR4 RAM, and Ubuntu 24.04 LTS OS.

A. Data Preprocessing

Table II shows the dimensions of the datasets for Subset 1A obtained through data preprocessing subphases. This table includes the dimensions of the original dataset. This stage removes 81 columns (36.5% of the original variables), keeping variables statistically relevant to the downstream outlier and anomaly detection task.

Significant attention is focused on evaluating the subphase that eliminates features highly correlated with the KPI. Fig. 4 (a) displays the BIC curves generated by 50 realizations of the Monte Carlo simulation to determine the optimal number of clusters for Subset 1A. This figure also shows the average BIC curve and the identified optimal number of clusters. In addition, Fig. 4 (b) depicts the clustering of the correlation factors within the interval $[0, 1]$. Fig. 5 depicts scatter plots of the KPI against the removed features. Each scatter plot indicates the respective correlation factor. The removed features strongly correlate with the KPI.

B. Outlier Detection

To evaluate the performance of the outlier detection stage, Fig. 6 presents the histograms of the logarithmic version of the KPI for the datasets described in Section III. As observed, the outlier detection subphase recognizes instances with significantly low performance. In mobile networks, this stage identifies scenarios where throughput falls below acceptable levels. For example, in Subsets 1A and 1C, the methodology detects cases with throughput below 80 kbps, while in Subsets 2A and 2B, it detects instances with throughput below 1 kbps. Such low-throughput conditions typically indicate significant service degradation, which can manifest as stalled downloads, prolonged web page loading times, or interruptions in real-time applications, such as video streaming or VoIP. We also compare the MAD-based outlier detection method with various state-of-the-art unsupervised approaches [8]. We implement isolation forest (IForest) [19], k-nearest neighbor (KNN) [26], local outlier factor (LOF) [4], principal component analysis (PCA) [32], Gaussian mixture model (GMM) [1], kernel density estimation (KDE) [14], one-class support vector machines (OCSVM) [30], clustering-based LOF (CBLOF) [9], connectivity-based outlier factor (COF) [34], histogram-based outlier detection (HBOS) [7], copula-based outlier detection (COPOD) [16], empirical cumulative distribution functions (ECOD) [17], and lightweight online detector of anomalies (LODA) [25].

Fig. 7 presents radar charts illustrating the F1-score and the area under the receiver operating characteristic curve (AUCROC) obtained across different datasets for each method. The MAD-based technique is the reference baseline, enabling a comparative evaluation across the various approaches. The radar plots illustrate that our proposed approach exhibits competitive performance to PCA, GMM, KDE, OCSVM, CBLOF, and COPOD. Notice that most of these methods require prior knowledge of the outlier proportion to distinguish outliers from non-outliers. In contrast, the MAD-based method operates independently of outlier proportion assumptions, relying on standard statistical parameters for detection. This characteristic provides adaptability to diverse network conditions.

Fig. 9 shows the SHAP beeswarm plots for normality models using (a) XGBoost and (b) LGBM, for the Subset 1A. Each point represents a dataset instance, with colors indicating feature values, blue for lower and red for higher values [20]. We can see in the figures the impact of the first four features, that are listed below:

1) `ack.pkts.sent.a2b`
2) `abs.idletime.avg`
3) `abs.segmentsizes.50`
4) `segs.cum.acked.b2a`

The feature `ack.pkts.sent.a2b` significantly influences the normality model. High values (red points) increase predictions, while low values (blue points) decrease them. This behavior indicates that throughput predictions depend on ACK packets sent from the mobile device to the server, reflecting TCP performance and data transmission efficiency. Conversely, the feature `abs.idletime.avg` behaves oppositely: high values result in lower predictions, while low values lead to higher predictions. This feature measures inactivity during the test, indicating potential conges-

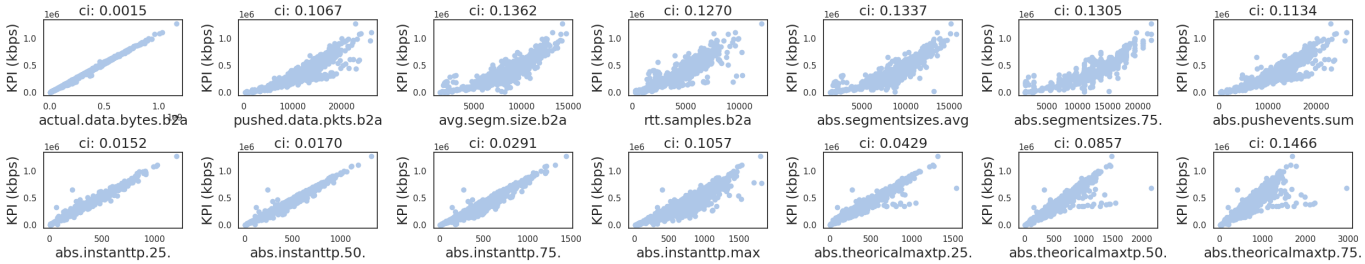


Fig. 5. Subset 1A. Scatter plots of the target KPI versus the features removed by the KPI-correlated detection and removal stage.

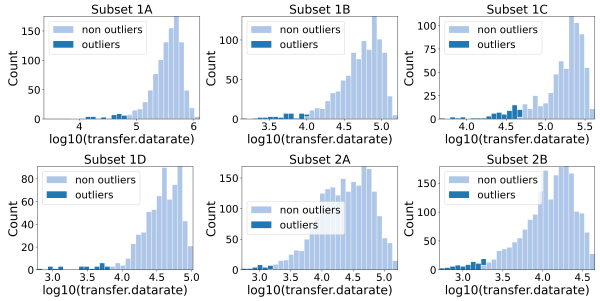


Fig. 6. Histograms of the logarithmic version of the target KPI for the six datasets under test, highlighting the detected outliers.

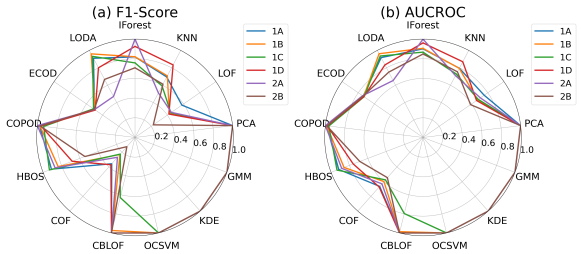


Fig. 7. (a) F1-Score yielded by different outlier detection methods for the six drive-test datasets, using our outlier detection approach as the ground truth. (b) AUCROC obtained by the various methods for the six datasets.

tion or routing problems. Thus, high `ack.pkts.sent.a2b` signifies efficient data transmission, while high values of `abs.idletime.avg` suggest signal delays.

C. Anomaly Detection

To observe the anomaly detection performance, Fig. 8 displays scatter plots of the vector of differences against the target KPI for the Subset 1A. Most alternative methods rely on discretization and classification to estimate the vector of differences [27], [28]. However, these approaches introduce quantization errors that, depending on the granularity of the discretization, imply a few kbps to tens or thousands of kbps. Our approach eliminates the discretization by employing a regression-based normality model. By leveraging a continuous regression output, ROAD produces an accurate vector of differences without introducing artificial approximations. This refinement preserves the fine-grained structure of performance variations, ensuring a precise characterization of deviations from normal behavior.

Fig. 8(a) depicts the location of anomalies (black stars) and non-anomalies (blue circles) generated by the MAD-based outlier detection approach. As seen, although the detected anomalies exhibit throughput values within the acceptable range (i.e., above 80 kbps), the performance significantly deviates from regression model predictions. In mobile networks, such deviations may reflect inefficient operational conditions, such as unexpected delays or suboptimal resource allocation, that are not captured by the outlier detection stage. Identifying these anomalies is crucial for proactive performance management, allowing operators to detect early signs of service degradation and address potential issues before they escalate into user-perceived failures. Fig. 8(b)-(n) depicts the location of anomalies and non-anomalies produced by other state-of-the-art techniques. These figures include the F1-score (F1) and AUCROC (AUC) obtained by detection methods using the MAD-based response as reference. The MAD-based approach performs similarly to GMM, KDE, CBLOF, and COPOD.

TABLE III
ACCURACY METRICS OBTAINED BY DETECTION METHODS OF ANOMALOUS SCENARIOS USING THE REAL DATASETS

Dataset	Metric	Method		
		XMLAD	Our approach	
		Outliers	Outliers	Anomalies
Subset 1A	Sensitivity	0.92	1.00	1.00
	Recall	0.36	0.99	1.00
	F1-score	0.10	0.94	0.95
Subset 2A	Sensitivity	0.69	1.00	0.57
	Recall	0.40	1.00	0.98
	F1-score	0.17	1.00	0.64
Subset 1B	Sensitivity	0.94	0.73	0.48
	Recall	0.61	1.00	1.00
	F1-score	0.54	0.81	0.65

We also compare the performance of our outlier and detection stages to other methods developed to identify anomalies in mobile networks [3]. Table III shows the accuracy metrics (sensitivity, recall, and F1-score) yielded by our methodology for three datasets for space limitations. Further, we include the results of XMLAD, an explainable state-of-the-art approach developed to detect performance losses in mobile networks [27]. XMLAD detects outliers only. Both methodologies exhibit competitive performance in sensitivity, i.e., remarkable results in detecting anomalies. Our approach performs better in identifying normal or non-outlier cases with a significant gain in the recall metric (63%, 60%, and 39%). This behavior is reflected in the F1-score. The results also show that the proposed approach provides outstanding outlier and anomaly detection performance.

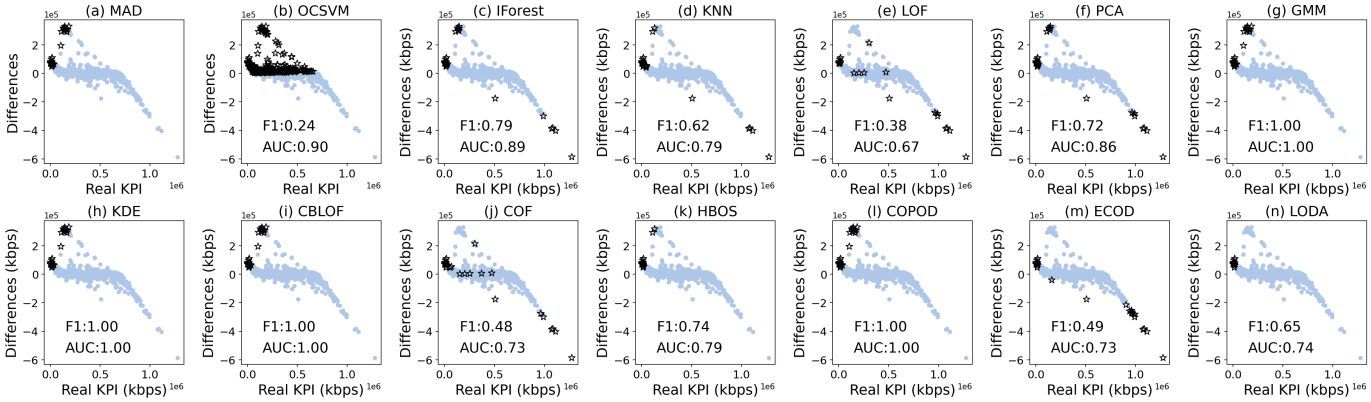


Fig. 8. Subset 1A. Scatter plots of the vector of differences versus the target KPI with the detected (black stars) and non-detected (blue circles) anomalies yielded by various unsupervised approaches. Each scatter plot displays the F1-score (F1) and the AUCROC (AUC) values generated by the detection methods. The anomaly pattern produced by the MAD-based detection method is used as reference.

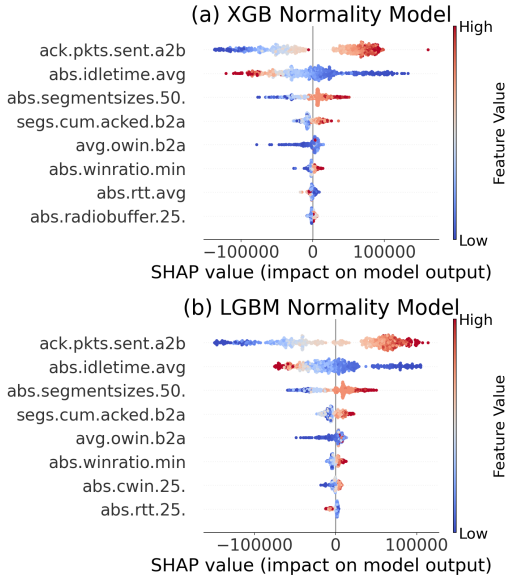


Fig. 9. Subset 1A. SHAP values beeswarm plots for normality models using (a) XGBoost, and (b) LGBM.

D. Interpretability Module

Fig. 10(a) shows the ROAD-based feature ranking for detected outliers in Subset 1A. The top four features associated with outliers are:

1) `abs.idletime.avg`
2) `abs.idletime.75`
3) `max.segm.size.b2a`
4) `abs.downlinkdelay.max`

Figs 10(c)-(f) display the vector of differences versus feature values for the top four features. These figures include the position of outliers and non-outliers. Outliers are closely related to significant idle times and low segment sizes. High idle times indicate excessive inactivity in certain network elements, possibly leading to poor network efficiency and user experience. Fig. 11 illustrates the decision tree structure for performance outliers. At the root node, the decision tree examines whether `abs.rwin.25` (percentile 25 of receive window) has values less than 12 584 448 bytes. If the comparison is true, the sample moves to the left part of the tree, detecting most of the performance outliers. The receive window (RWIN) measures the amount of data, in bytes, that the receiver can accept before needing an acknowledgment packet. If RWIN is small, the remote server may require acknowledgment packets more frequently, which could reduce the network throughput. It is important to note that the drive-test datasets used in this study include measurements related to various aspects of mobile network operation, including radio, traffic, and parameters configured by the operator. For example, these datasets provide variables related to radio performance, including RSRP (Reference Signal Received Power), RSRQ (Reference Signal Received Quality), and SINR (Signal-to-Interference-plus-Noise Ratio). Notably, in the dataset under study, the interpretability module primarily identified features linked to the TCP/IP layers, such as window size and packet delays, as being the most correlated with the detected outliers. This result indicates that, while the radio environment was adequately measured and stable in most scenarios, outliers were more closely linked to transport and network-layer behaviors.

Fig. 10(b) illustrates the ROAD-based ranking for detected performance anomalies. The top features are:

1) `abs.segmentsizes.50`
2) `max.segm.size.b2a`
3) `abs.rwin.75`
4) `max.win.adv.a2b`

Figs 10(g)-(j) present the vector of differences versus the feature values for the top attributes. These figures display locations of anomalies and non-anomalies. The performance outliers are related to low segment sizes. Small segment sizes suggest the network operates with excessive retransmissions or flow control restrictions. Fig. 12 displays the decision tree structure for performance anomalies. In this case, the root node separate anomalies and non-anomalies by comparing `abs.segmentsizes.50` with the threshold of approximately 1 500 bytes. The Gini index in the final leaves is zero, indicating that the decision tree identifies the intervals in features associated with detected anomalies. In particular, segment sizes refer to the size of data segments transmitted in

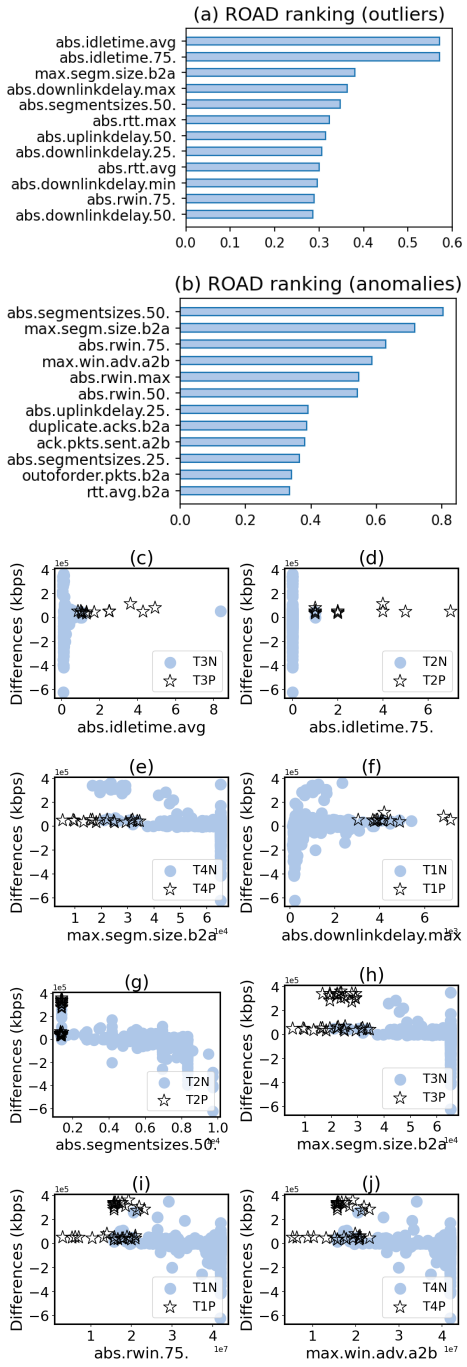


Fig. 10. Subset 1A. Outliers: (a) ROAD-based ranking, (c)-(f) vector of differences versus feature values with locations of outliers (black stars) and non-outliers (blue circle) for the top four features. Anomalies: (b) ROAD-based ranking, (g)-(j) vector of differences versus feature values with locations of anomalies (black stars) and non-anomalies (blue circles).

the network, typically in bytes. Lower segment sizes could indicate congestion or inefficiencies in data transmission. When combined with other parameters, variations in segment size can provide additional information. For example, low segment sizes and high RTT values may indicate network congestion or queuing delays forcing TCP to reduce segment sizes.

X. Conclusions and Future Work

The expansion of network infrastructures requires interpretable anomaly detection methods to identify performance issues and support corrective actions. This work introduced the ROAD (Interpretable Outlier and Anomaly Detection) methodology to identify and understand atypical scenarios from small tabular data. We defined and differentiated outliers and anomalies in the context of mobile network performance. Then, we described a data preprocessing approach that ensures the extraction of high-quality samples. To enhance detection accuracy, we incorporated a regression-based normality model, which refines the estimation of differences between KPI values and their predictions. Beyond detection, this study presented a novel interpretability module that identifies patterns closely related to atypical samples. By leveraging interpretable tools, our approach provided deep insights into features closely linked to anomalies. We validated the proposed framework using real datasets, demonstrating its effectiveness in detection and interpretability. For future work, we aim to build a technological solution for real-time monitoring.

XI. Ethics

The sensitive information on network operators was deleted upon aggregation. The level of aggregation ensures that no operator or customer is re-identified in the acceptance of the General Data Protection Regulation.

References

[1] C. C. Aggarwal. *An Introduction to Outlier Analysis*, pages 1–34. Springer International Publishing, Cham, 2017.

[2] N. Al-Falahy and O. Y. Alani. Technologies for 5G Networks: Challenges and Opportunities. *IT Professional*, 19(1):12–20, 2017.

[3] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[4] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, page 93104, New York, NY, USA, 2000. Association for Computing Machinery.

[5] M. Carletti, C. Masiero, A. Beghi, and G. A. Susto. Explainable machine learning in industry 4.0: Evaluating feature importance in anomaly detection to enable root cause analysis. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 21–26, 2019.

[6] T. Chen and C. Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785794, New York, NY, USA, 2016. Association for Computing Machinery.

[7] M. Goldstein and A. Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: poster and demo track*, 1:59–63, 2012.

[8] S. Han, X. Hu, H. Huang, M. Jiang, and Y. Zhao. Adbench: Anomaly detection benchmark. *Advances in Neural Information Processing Systems*, 35:32142–32159, 2022.

[9] Z. He, X. Xu, and S. Deng. Discovering cluster-based local outliers. *Pattern recognition letters*, 24(9-10):1641–1650, 2003.

[10] P. Huber, J. Wiley, and W. InterScience. *Robust statistics*. Wiley New York, 1981.

[11] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013.

[12] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[13] C. Kim, V. B. Mendiratta, and M. Thottan. Unsupervised anomaly detection and root cause analysis in mobile networks. In *2020 International Conference on COMmunication Systems and NETworkS (COMSNETS)*, pages 176–183, 2020.

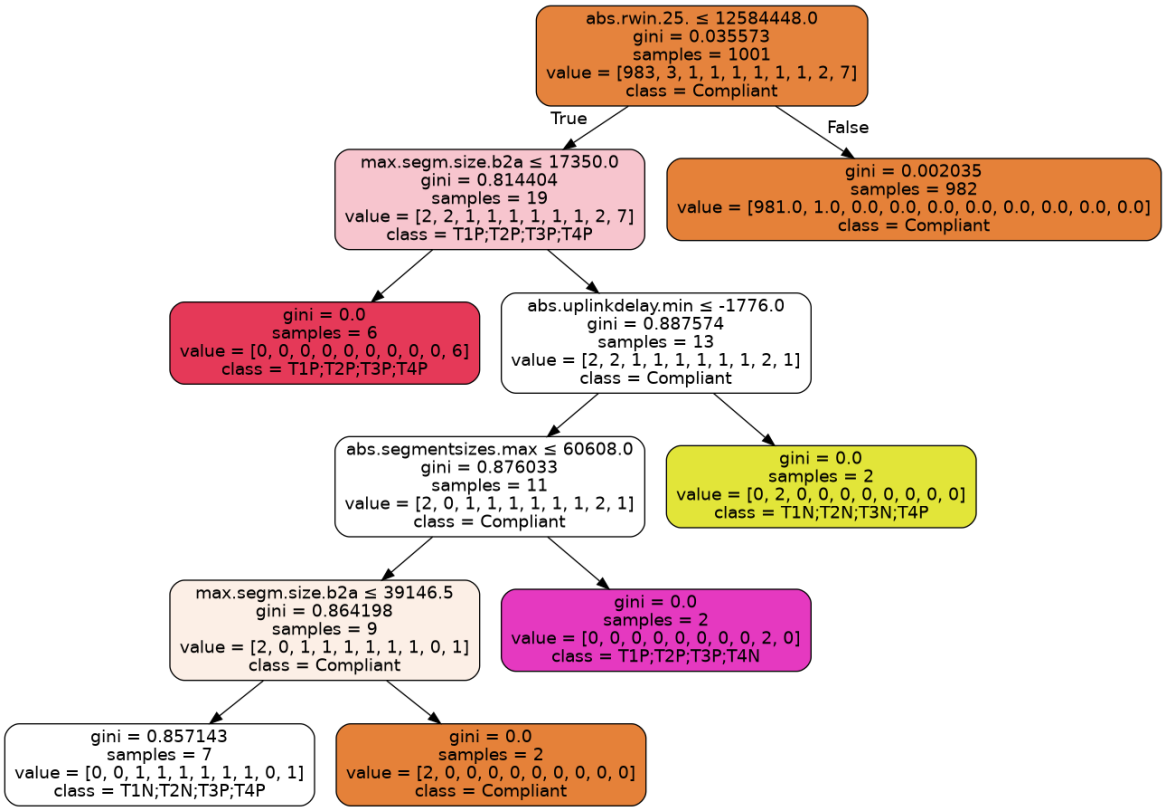


Fig. 11. Subset 1A. Decision tree structure of the interpretability module for performance outlier detection.

[14] L. J. Latecki, A. Lazarevic, and D. Pokrajac. Outlier detection with kernel density functions. In *International workshop on machine learning and data mining in pattern recognition*, pages 61–75. Springer, 2007.

[15] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 49(4):764–766, 2013.

[16] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu. Copod: Copula-based outlier detection. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1118–1123, 2020.

[17] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, and G. H. Chen. Ecod: Unsupervised outlier detection using empirical cumulative distribution functions. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12181–12193, 2023.

[18] Z. Li, Y. Zhu, and M. Van Leeuwen. A survey on explainable anomaly detection. *ACM Trans. Knowl. Discov. Data*, 18(1), sep 2023.

[19] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation-based anomaly detection. *ACM Trans. Knowl. Discov. Data*, 6(1), Mar. 2012.

[20] S. M. Lundberg and S.-I. Lee. A Unified Approach to Interpreting Model Predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[21] G. J. McLachlan and K. E. Basford. *Mixture models: Inference and applications to clustering.* Marcel Dekker, New York, 1988.

[22] M. Moulay, R. G. Leiva, V. Mancuso, P. J. Rojo Maroni, and A. F. Anta. Ttrees: Automated classification of causes of network anomalies with little data. In *2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 199–208, 2021.

[23] M. Moulay, R. G. Leiva, P. J. R. Maroni, F. Diez, V. Mancuso, and A. F. Anta. Automated identification of network anomalies and their causes with interpretable machine learning: The cian methodology and ttrees implementation. *Computer Communications*, 191:327–348, 2022.

[24] M. Moulay, R. G. Leiva, P. J. R. Maroni, J. Lazaro, V. Mancuso, and A. F. Anta. A novel methodology for the automated detection and classification of networking anomalies. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 780–786, 2020.

[25] T. Pevnỳ. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102:275–304, 2016.

[26] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. SIGMOD '00, page 427438, New York, NY, USA, 2000. Association for Computing Machinery.

[27] J. M. Ramírez, F. Díez, P. Rojo, V. Mancuso, and A. Fernández-Anta. Explainable machine learning for performance anomaly detection and classification in mobile networks. *Computer Communications*, 200:113–131, 2023.

[28] J. M. Ramrez, P. Rojo, F. Dez, V. Mancuso, and A. F. Anta. Cleaning matters! preprocessing-enhanced anomaly detection and classification in mobile networks. In *2022 20th Mediterranean Communication and Computer Networking Conference (MedComNet)*, pages 103–112, 2022.

[29] A. K. Sangaiah, S. Rezaei, A. Javadpour, F. Miri, W. Zhang, and D. Wang. Automatic fault detection and diagnosis in cellular networks and beyond 5g: Intelligent network management. *Algorithms*, 15(11), 2022.

[30] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. Support Vector Method for Novelty Detection. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999.

[31] I. Shayea, M. H. Azmi, M. Ergen, A. A. El-Saleh, C. T. Han, A. Arsad, T. A. Rahman, A. Alhammadi, Y. I. Daradkeh, and D. Nandi. Performance analysis of mobile broadband networks with 5g trends and beyond: Urban areas scope in malaysia. *IEEE Access*, 9:90767–90794, 2021.

[32] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang. A novel anomaly detection scheme based on principal component classifier. In *Proceedings of the IEEE foundations and new directions of data mining workshop*, pages 172–179. IEEE Press Piscataway, NJ, USA, 2003.

[33] P. D. Talagala, R. J. Hyndman, and K. Smith-Miles. Anomaly detection in high-dimensional data. *Journal of Computational and Graphical Statistics*, 30(2):360–374, 2021.

[34] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung. Enhancing effectiveness of outlier detections for low density patterns. In *Advances in Knowledge Discovery and Data Mining: 6th Pacific-Asia Conference,*

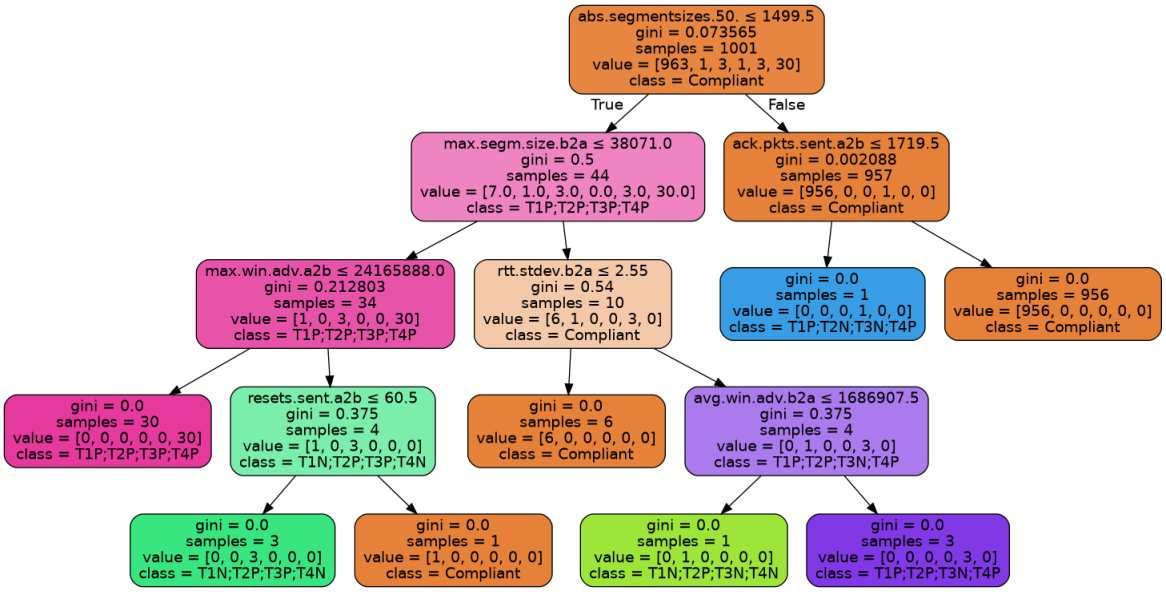


Fig. 12. Subset 1A. Decision tree structure of the interpretability module for anomaly detection.

PAKDD 2002 Taipei, Taiwan, May 6–8, 2002 Proceedings 6, pages 535–548. Springer, 2002.

[35] A. Zaki, A. Métwalli, M. H. Aly, and W. K. Badawi. 5g and beyond: Channel classification enhancement using vif-driven preprocessing and machine learning. *Electronics*, 12(16):3496, 2023.