PARs: Predicate-based Association Rules for Efficient and Accurate Anomaly Explanation

Cheng Feng
cheng.feng@siemens.com
Siemens Technology
Beijing, China

Abstract

While new and effective methods for anomaly detection are frequently introduced, many studies prioritize the detection task without considering the need for explainability. Yet, in real-world applications, anomaly explanation, which aims to provide explanation of why specific data instances are identified as anomalies, is an equally important task. In this work, we present a novel approach for efficient and accurate model-agnostic anomaly explanation for tabular data using Predicate-based Association Rules (PARs). PARs can provide intuitive explanations not only about which features of the anomaly instance are abnormal, but also the reasons behind their abnormality. Our user study indicates that the anomaly explanation form of PARs is better comprehended and preferred by regular users of anomaly detection systems as compared to existing model-agnostic explanation options. Furthermore, we conduct extensive experiments on various benchmark datasets, demonstrating that PARs compare favorably to state-of-the-art model-agnostic methods in terms of computing efficiency and explanation accuracy on anomaly explanation tasks. The code for our experiments is available at https://github.com/cfeng783/PARs.

CCS Concepts

• **Information systems** → **Association rules**; • **Computing methodologies** → **Rule learning**; • **Security and privacy** → **Intrusion/anomaly detection and malware mitigation**.

Keywords

Model-agnostic anomaly explanation; Predicate-based association rules

ACM Reference Format:
Cheng Feng. 2024. PARs: Predicate-based Association Rules for Efficient and Accurate Anomaly Explanation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3627673.3679625

1 Introduction

Anomaly detection, which aims to identify data instances that do not conform to the expected behavior, is a classic machine learning task with numerous applications in various domains including fraud detection, intrusion detection, predictive maintenance, etc. Over the past decades, numerous methods have been proposed to tackle this challenging problem. Examples include one-class classification-based [18, 29], nearest neighbor-based [2], clustering-based [10], isolation-based [9, 13], density-based [5, 12] and deep anomaly detection models based on autoencoders [39, 40], generative adversarial networks [8, 38], to name a few. However, comparing to the vast body of literature on the detection task, anomaly explanation techniques have received relatively little attention so far [28]. In fact, providing accurate explanations of why specific data instances are detected as anomalies is equally critical for many real-world applications. For instance, when an anomaly is reported by a fault detection application for a critical device in a factory, human operators need straightforward clues regarding the reported anomaly, and then can decide what next steps - such as fault diagnosis, predictive maintenance and system shutdown – should be taken. The required clues include which feature(s) is abnormal, why that feature(s) is abnormal.

To fill this gap, we propose a novel approach for efficient and accurate model-agnostic anomaly explanation for tabular data. Specifically, we leverage decision tree learning and association rule mining [1] to learn Predicate-based Association Rules (PARs) which capture normal behaviors exhibited by training data. During inference, we efficiently find the precise PARs for explaining anomalies identified by arbitrary anomaly detection models. PARs provide intuitive explanations not only about which features of the anomaly instance are abnormal, but also why those features are abnormal. Our user study shows that the anomaly explanation form of PARs is better understood and favoured by regular anomaly detection system users compared with existing model-agnostic anomaly explanation options. In our experiments, we demonstrate that it is significantly more efficient to find PARs than anchors [27], another rule-based explanation, for identified anomaly instances. Moreover, PARs are also far more accurate than anchors for anomaly explanation, meaning that they have considerably higher precision and recall when applied as anomaly detection rules on unseen data other than the anomaly instance on which they were originally derived for explanation. Additionally, we show that PARs can also achieve higher accuracy on abnormal feature identification compared with many state-of-the-art model-agnostic explanation methods including LIME [26], SHAP [17], COIN [14] and ATON [35]. We summarize the main contributions of this paper as follows:

- We introduce PARs, a novel and informative explanation form for anomalies, and the form of PARs is more appealing to regular anomaly detection system users according to our user study.

This work is licensed under a Creative Commons Attribution International 4.0 License.

CIKM '24, October 21–25, 2024, Boise, ID, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679625

- We provide a purely data-driven approach to efficiently constructing and finding the precise PARs for anomaly explanation.
- We reduce the expected time cost of deriving explanation rules for a single anomaly instance in the inference stage from *tens of seconds* to *less than one second* compared to anchors, which is critical for online anomaly detection and diagnosis applications.
- We conduct extensive experiments on public benchmark datasets to show that PARs can achieve more accurate anomaly explanation than existing model-agnostic explanation methods.

2 PARs as Anomaly Explanations

Given a black box anomaly detection model $f : \mathcal{X} \longrightarrow \mathcal{Y}$ where $\mathcal{Y} \in \{0, 1\}$ with 0 indicating normality and 1 indicating abnormality, and an anomaly data instance $x \in \mathcal{X}$ with $f(x) = 1$, let $\mathcal{F}$ be all the features of $\mathcal{X}$, our model-agnostic anomaly explanation aims to achieve two goals: 1) identify the abnormal feature subspace $\mathcal{F}_{sub} \in \mathcal{F}$ of x; 2) intuitively explain why the feature subspace $\mathcal{F}_{sub}$ of x is abnormal. Both goals can be well achieved by PARs.

Specifically, a PAR is an association rule in the following form: $P \longrightarrow p$ where P represents a set of antecedent predicates and p represents a single consequent predicate such that $p \notin P$. PARs describe patterns of behavior that normal data instances should follow. An anomaly instance x can be explained by a PAR if it violates the PAR, i.e., all the antecedent predicates of the PAR are satisfied but the consequent predicate is **not** satisfied by x. For example, in Table 1, the PAR Level>10, Pump=ON $\longrightarrow$ Valve=Open gives following explanation for the anomaly instance x=[11.1,ON,Close,25] with features $\mathcal{F}$=[Level, Pump, Valve, Temperature]: the Valve feature is abnormal because according to the PAR, when Level>10 and Pump=ON, then Valve should be Open, however, Valve=Close in x. This PAR actually describes a physical law that governs the behavior of a water tank system, where the outlet valve must be open to prevent the tank from exploding when the water level in the tank exceeds a certain threshold and the inlet pump remains on. Utilizing such rules with potential physical meanings for anomaly explanation is highly helpful for assisting users in understanding and diagnosing detected anomalies.

3 Related Work

3.1 Model-agnostic Anomaly Explanation

The mainstream of prior art on model-agnostic anomaly explanation is to find the most anomalous/outlying feature subspace of anomaly instances in the score-and-search manner [3, 31, 32, 34]. Specifically, these methods identify abnormal features by searching for possible feature subspaces and compute the anomalous/outlying score of the anomaly instance in each subspace. However, since the number of possible subspaces increases exponentially with the growth of feature dimension, these methods are oftentimes too costly and potentially ineffective for high-dimensional data. Consequently, some methods are proposed to improve the efficiency and accuracy of abnormal feature identification. For example, SOAM [16] is an outlying feature mining method which fits an Sum-Product Network (SPN) to model high-dimensional feature distributions, and leverages the tractability of marginal inference in SPNs to efficiently compute outlier scores in feature subsets. COIN [14] transforms the anomaly explanation task into a classification problem which involves training a series of $l1-$norm classifiers that separate augmented anomalies from clusters of normal data in close proximity, and uses the classifiers' weights as anomaly contribution weights of features. ATON [35] leverages the attention mechanism of neural networks to assign anomaly contribution weights to feature dimensions. Specifically, it utilizes a triplet deviation-based loss which estimates the separability of the anomaly instance and some heuristically sampled informative normal data within the triplets, and then a self-attention module is optimized to compute the contribution of each feature dimension to the anomaly instance.

It is noteworthy to mention that most model-agnostic anomaly explanation methods like above focus on providing explanations for why the data instance is abnormal rather than why the anomaly detection algorithm has deemed it to be so. This is because the primary objective of anomaly explanation is to assist users for subsequent decision-making and diagnosis regarding the reported anomalies. PARs also focus on explaining the abnormality of data instead of the decision logic of the anomaly detection algorithm.

3.2 General Local Model-agnostic Explanation

Although not specifically designed for anomaly explanation tasks, some general local model-agnostic explanation methods that employ a perturbation-based strategy to generate local explanations for predictions of black box machine learning models, such as LIME [26], SHAP [17] and Anchor [27], are frequently used for anomaly explanation by considering anomaly detection as a binary classification problem. However, there are certain drawbacks of utilizing these general methods for anomaly explanation: 1) the perturbation-based strategy to generate local explanations for the algorithms is rather time-consuming, this makes such methods less applicable in typical online anomaly detection applications. 2) Determining the precise form to explain the decision logic of a complex anomaly detection algorithm is far more difficult than identifying the correct cause for the abnormality of data, which makes the derived explanations less robust than those methods directly explaining the abnormality of data.

3.3 Anomaly Explanation Forms

It is important to emphasize that both informativeness and intuitiveness are crucial for anomaly explanation methods. To show the differences between the anomaly explanation forms provided by various model-agnostic methods, we illustrate the explanations of SOAM, COIN, ATON, SHAP, LIME, Anchor and PAR to an anomaly instance in a water tank condition monitoring dataset in Table 1. As can be seen, most existing anomaly explanation methods only handle the abnormal feature identification task but do not provide intuitive explanations about why the corresponding features are abnormal. The only exception is Anchor, which also provides rule-based explanation about why a data instance is reported as anomaly. However, PARs are more informative than anchors. Referring to the example presented in Table 1, when the predicate violation on the right-hand side is known, specifically Valve=Open, the user is able to identify the suspected abnormal feature. However, the anchor

Table 1: Illustration of explanations of various model-agnostic methods to an anomaly instance (x): Level=11.1, Pump=ON, Valve=Close, Temperature=25 in a water tank condition monitoring dataset.

Method	Explanation
SOAM	{Level, Pump, Valve} Level, Pump and Valve are abnormal features.
COIN ATON	Level: 0.3, Pump: 0.2, Valve: 0.4, Temperature: 0.1 The abnormality weights for above features are [0.3, 0.2, 0.4, 0.1].
SHAP	Level: 0.3↑, Pump: 0.2↑, Valve: 0.4↑, Temperature: 0.1↓ Level, Pump and Valve features push the probability of prediction to anomaly higher and the Temperature feature pushes the probability lower with different degrees.
LIME	Level>10: 0.3↑, Pump=ON: 0.2↑, Valve=Close: 0.4↑, Temperature<30: 0.1↓ Level>10, Pump=ON and Valve=Close push the probability of prediction to anomaly higher and Temperature<30 pushes the probability lower with different degrees.
Anchor	Level>10, Pump=ON, Valve=Close $\longrightarrow f(x) = 1$ If Level>10, Pump=ON and Valve=Close, then x is classified as an anomaly.
PAR	Level>10, Pump=ON $\longrightarrow$ Valve=Open If Level>10 and Pump=ON, then Valve should be OPEN. If Level>10, Pump=ON but Valve≠Open, then x is classified as an anomaly.

rule fails to accurately pinpoint the specific abnormal feature. Additionally, understanding the predicate violation on the right-hand side also informs the user about the correct behavior expected from the abnormal feature. In this instance, the valve should be in an open state rather than closed. This crucial information cannot be derived from the anchor rule when the variable can assume more than two potential values.

3.4 Relation to Explainable Anomaly Detection

There are existing methods which leverage association rules [4, 22, 36] and other dependency-based methods [15, 24] for explainable anomaly detection [11]. However, we emphasize that our work is fundamentally different from those explainable anomaly detection methods. Specifically, those methods did not separate the anomaly detection task from the anomaly explanation task. Consequently, their application in practice is significantly limited due to their reliance on a relatively weak or less-general model for anomaly detection. In contrast, our approach designs a novel systematic framework for efficiently constructing and leveraging association rules exclusively for anomaly explanation. The decoupling of the anomaly explanation task from the anomaly detection task makes our method highly general and useful in practical applications.

4 The Method

In this section, we present how to learn PARs from data, how to find precise PARs for anomaly explanation, and guidelines for setting hyperparameter values in our method.

4.1 Learning PARs from Data

In contrast to general local model-agnostic explanation methods where explanations are generated by a costly perturbation-based process during inference, we learn and store all PARs in the training stage. During inference, we simply need to efficiently find the precise PARs to explain the anomaly instance.

Learning PARs from a given dataset $\mathcal{D}$ mainly consists of two steps: *predicate generation* and *PAR mining*. Firstly, we derive a global predicate set $\mathcal{P}$ in the predicate generation step. We then transform each data instance $x \in \mathcal{D}$ as a set of satisfied predicates P such that $P \subseteq \mathcal{P}$. Let $P_1, \ldots, P_{|\mathcal{D}|}$ represent the records of satisfied predicates for all data instances in $\mathcal{D}$, we then mine all PARs that satisfy a minimum support condition and a minimum confidence condition from the records. Specifically, let $P \longrightarrow p$ be a PAR, θ and γ be the minimum support and minimum confidence thresholds respectively, we require $sup(P \longrightarrow p) > \theta$ and $conf(P \longrightarrow p) > \gamma$ where $sup(P \longrightarrow p) = \frac{\#(P \cup p)}{|\mathcal{D}|}$ measures the frequency of apparition of $P \cup p$ within the dataset, $conf(P \longrightarrow p) = \frac{sup(P \longrightarrow p)}{sup(P)}$ measures the percentage of data satisfying all the predicates in P that also satisfy the consequent predicate p. Intuitively, a higher support for a PAR indicates a higher coverage of data whereas a higher confidence indicates a higher precision for explaining anomalies.

4.1.1 Predicate Generation. The quality of the global predicate set $\mathcal{P}$ is critical in our method. Specifically, the generated predicates should have high likelihoods leading to PARs. Furthermore, the form of generated predicates should be as simple as possible to maximize the interpretability of PARs. With these points in mind, we propose two algorithms for generating predicates, one specifically for categorical features and the other for numeric features. Before presenting the algorithms, it is important to highlight that in order for any generated predicate p to have a nonzero probability of contributing to a PAR, $sup(p) > \theta$ is required. It can be seen that if the support of a predicate p is less than θ, then p can never contribute to any PAR due to the anti-monotone constraint [21].

The algorithm of generating predicates for categorical features is straightforward. Let $\{1, \ldots, U\}$ be the set of observed values for a categorical feature F^c in $\mathcal{D}$, we generate candidate predicates $p : F^c = u$ for all $u \in \{1, \ldots, U\}$. If $sup(p) > \theta$, we add p to $\mathcal{P}$. Otherwise, we add p to a list $\mathcal{L}$ which stores candidate predicates whose supports are less than the threshold. Then, we traverse all predicates in $\mathcal{L}$ and use the "or"("|") operator to generate combined predicates until their supports are larger than the threshold. For example, let $sup(p_1) < \theta$ and $sup(p_2) < \theta$, we generate a combined predicate $p : p_1|p_2$ if $sup(p_1|p_2) > \theta$. It is noteworthy to mention that we also prioritize the combination of predicates belonging to the same feature to optimize interpretability. Implementation details are given in the Algorithm 1.

To promote interpretability, we generate predicates for each numeric feature by a set of proposed cut-off values. For example, assuming there are three cut-off values τ_1, τ_2, τ_3 for a numeric feature F^n where $\tau_1 < \tau_2 < \tau_3$, we generate four predicates which are $p_1 : F^n < \tau_1$, $p_2 : \tau_1 \leq F^n < \tau_2$, $p_3 : \tau_2 \leq F^n < \tau_3$ and $p_4 : F^n \geq \tau_3$ and add them to $\mathcal{P}$ if the support for each predicate is larger than θ. Moreover, we prefer cut-off values for a numeric feature which maximize the reduction of uncertainty in other features. In this way, predicates generated by such cut-off values could contribute to PARs with high likelihoods. Concretely, we employ a dependency-based approach which consists of following two steps to generate predicates for numeric features: 1) propose a set of candidate cut-off values by learning decision tree (DT) models on $\mathcal{D}$, 2) select cut-off values with higher impurity decrease to generate predicates.

Propose candidate cut-off values: In this step, we learn two sets of DT models on $\mathcal{D}$ to propose candidate cut-off values. Specifically, Let $\mathcal{F}^c$ and $\mathcal{F}^n$ be the set of all categorical features and the set of all numeric features of the dataset, we first learn a DT classification model $DT(\mathcal{F}^n) \longrightarrow F^c$ for each categorical feature $F^c \in \mathcal{F}^c$ where all the numeric features $\mathcal{F}^n$ are used as input variables and F^c is the prediction target. As for the second set, we learn a DT regression model $DT(\mathcal{F}^n_{-i}) \longrightarrow F^n_i$ for each numeric feature $F^n_i \in \mathcal{F}^n$ where all the remaining numeric features $\mathcal{F}^n_{-i}$ are used as input variables. Information gain and variance reduction [1] are used to measure the quality of a split for the classification models and the regression models, respectively. We set the minimum number of samples required at a leaf node to be greater than $|\mathcal{D}| \times \theta$ as the stop criterion for the training of DT models. This avoids creating cut-off values which would definitely lead to predicates with support lower than θ. Since each internal node in a learned DT model defines a split rule $\mathbf{1}_{(\tau,\infty)}(x_{F^n})$ where τ is a cut-off value on feature F^n that directs an instance x to the left/right child node, we can extract a tuple (F^n, τ, q_τ) from an internal node where q_τ is the impurity decrease of the node. Specifically,

$$q_\tau = \frac{N}{|\mathcal{D}|}(H - \frac{N_{left}}{N}H_{left} - \frac{N_{right}}{N}H_{right})$$

where N, N_{left} and N_{right} are the number of data instances reaching the node, its left child and its right child; H, H_{left} and H_{right} are the impurity of the target feature for data reaching the node, its left child and right child, respectively. More specifically, impurity at a node is calculated as the entropy of the target feature for the data reaching the node for DT classification models and variance of the target feature for the data reaching the node for DT regression models.

Select cut-off values for predicate generation: It is beneficial to select cut-off values with higher q_τ values to generate predicates because such cut-off values reduce more uncertainty for the corresponding target feature and thus are more likely to contribute to PARs containing that feature. Therefore, after extracting all the cut-off values for a numeric feature F^n from all the trained DT models, we arrange $F^n : (\tau_1, \ldots, \tau_J)$ for the numeric feature such that $q_{\tau_1} \geq \ldots \geq q_{\tau_J}$, i.e., the q_τ value for its cut-off values are sorted in a descending order. Then, we sequentially traverse the list of

[1]For DT regression models, the target feature is standardized before training such that variance reduction for different features are comparable.

Algorithm 1 Predicate generation for categorical features

Require: The dataset $\mathcal{D}$, the minimum support threshold θ, the categorical feature set $\mathcal{F}^c$

$\mathcal{P} \leftarrow \emptyset$, $\mathcal{L} \leftarrow \emptyset$
for $i = 1, \ldots, |\mathcal{F}^c|$ **do**
  $\mathcal{T} \leftarrow \emptyset$
  Let $\{1, \ldots, U\}$ be the set of possible values for a categorical feature F^c_i
  for $u = 1, \ldots, U$ **do**
    Generate predicate p: $F^c_i = u$
    if $sup(p) > \theta$ **then**
      Add p to $\mathcal{P}$
    else
      Add p to $\mathcal{T}$
    end if
  end for
  if $|\mathcal{T}| > 1$ **then**
    Generate predicate p: $p_1|\ldots|p_{|\mathcal{T}|}$ # combine predicates for a single feature
    if $sup(p) > \theta$ **then**
      Add p to $\mathcal{P}$
    else
      Add p to $\mathcal{L}$
    end if
  else if $|\mathcal{T}| = 1$ **then**
    Add p in $\mathcal{T}$ to $\mathcal{L}$
  end if
end for
$k \leftarrow 1$
for $j = 2, \ldots, |\mathcal{L}|$ **do**
  if $sup(p_k|\ldots|p_j) > \theta$ **then**
    if $sup(p_{j+1}|\ldots|p_{|\mathcal{L}|}) > \theta$ **then**
      Generate predicate p: $p_k|\ldots|p_j$
      Add p to $\mathcal{P}$
      $k \leftarrow j + 1$
    else
      Generate predicate p: $p_k|\ldots|p_{|\mathcal{L}|}$
      Add p to $\mathcal{P}$
      break
    end if
  end if
end for
return $\mathcal{P}$

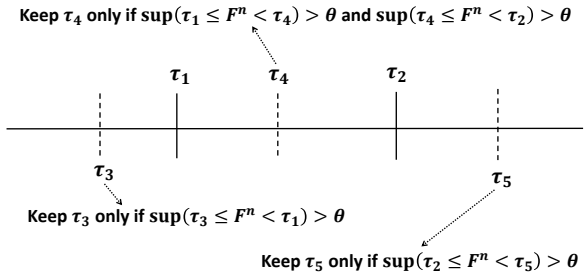


Figure 1: Illustration of the logic for whether keeping a cut-off value with a lower q_τ value for predicate generation. In the example, we assume $q_{\tau_1} \ge q_{\tau_2} \ge q_{\tau_3} \ge q_{\tau_4} \ge q_{\tau_5}$.

cut-off values and only keep a cut-off value for predicate generation for F^n if its inclusion will not cancel the predicate of a cut-off value with a higher q_τ value. The logic of whether to keep a cut-off value for predicate generation is illustrated in Figure 1. Implementation details of the whole algorithm is given in Algorithm 2.

4.1.2 PAR Mining. After obtaining the global predicate set $\mathcal{P}$, we transform each data instance $x \in \mathcal{D}$ as a set of satisfied predicates P such that $P \subseteq \mathcal{P}$. Given the records of satisfied predicates for the data as $P_1, \ldots, P_{|\mathcal{D}|}$, mining PARs becomes an association rule mining problem. Concretely, we first find all frequent itemsets (predicate sets) with the minimum support threshold θ from the records using the FPGrowth algorithm [6]. Then, for an arbitrary frequent predicate set P, regarding each $p \in P$, we partition P into two parts p and $P - p$, a PAR $P - p \longrightarrow p$ is generated if its confidence is larger than γ.

Besides, we also generate a set of special PARs called univariate PARs for explaining anomalies with simply out-of-range feature values. Concretely, for each categorical feature F^c, we generate an univariate PAR: $\emptyset \longrightarrow F^c \in \{1, \ldots, U\}$, where $\{1, \ldots, U\}$ is the set of seen values for F^c in $\mathcal{D}$. For each numeric feature F^n, we generate an univariate PAR: $\emptyset \longrightarrow \mu - 3\sigma \le F^n \le \mu + 3\sigma$, where μ and σ are the mean and standard deviation of the values for F^n in $\mathcal{D}$.

4.2 Finding Precise PARs for Explanation

Let $\mathcal{A}$ be the set of all PARs learned from the training data $\mathcal{D}$. In the inference stage, when an anomaly instance x is identified by a black box anomaly detection model f, we find the top-k PARs with the highest supports and confidences that are violated by x to precisely explain the anomalous behavior of x. Concretely, the top-k PARs are selected as follows:

$$\{A_1, ..., A_k\} = \underset{A \in \mathcal{A} \text{ and } A(x)=1}{\arg \text{Topk}} \quad \frac{sup(A) - \theta}{1 - \theta} + \lambda \frac{conf(A) - \gamma}{1 - \gamma}$$

where we call $(\frac{sup(A)-\theta}{1-\theta} + \lambda \frac{conf(A)-\gamma}{1-\gamma})$ the *accuracy score* of PAR A for anomaly explanation, in which λ denotes the relative importance weight for the confidence with respect to the support; $A(x) = 1$ if and only if A is violated by x. Intuitively, this means we prefer to search for violated PARs with the highest coverage of data (support) and highest precision (confidence) to accurately explain anomalies.

Algorithm 2 Predicate generation for numeric features

Require: The dataset $\mathcal{D}$, the global predicate set $\mathcal{P}$, the minimum support threshold θ
Require: The categorical feature set $\mathcal{F}^c$, the numeric feature set $\mathcal{F}^n$
$\mathcal{T} \leftarrow \emptyset$
for $i = 1, \ldots, |\mathcal{F}^c|$ **do**
  Train a DT classification model $DT(\mathcal{F}^n) \longrightarrow F_i^c$ on $\mathcal{D}$
  For each internal node j in the trained DT model, add $(F_j^n, \tau_j, q_{\tau_j})$ to $\mathcal{T}$ based on its split rule
end for
for $i = 1, \ldots, |\mathcal{F}^n|$ **do**
  Train a DT regression model $DT(\mathcal{F}_{-i}^n) \longrightarrow F_i^n$ on $\mathcal{D}$
  For each internal node j in the trained DT model, add $(F_j^n, \tau_j, q_{\tau_j})$ to $\mathcal{T}$ based on its split rule
end for
for $i = 1, \ldots, |\mathcal{F}^n|$ **do**
  Get $F_i^n : (\tau_1, \ldots, \tau_J)$ from $\mathcal{T}$ where $q_{\tau_1} \ge ... \ge q_{\tau_J}$
  $\mathcal{L} \leftarrow [\tau_1]$ # list of cut-off values to keep
  for $j = 2, \ldots, J$ **do**
    $k \leftarrow 1$ # insert position of τ_j
    while $k \le |\mathcal{L}|$ and $\tau_j < \mathcal{L}[k]$ **do**
      $k \leftarrow k + 1$
    end while
    if $k = 1$ **then**
      if $sup(\tau_j \le F_i^n < \mathcal{L}[1]) > \theta$ **then**
        Insert τ_j to $\mathcal{L}$ at position k
      end if
    else if $k = |\mathcal{L}| + 1$ **then**
      if $sup(\mathcal{L}[k-1] \le F_i^n < \tau_j) > \theta$ **then**
        Insert τ_j to $\mathcal{L}$ at position k
      end if
    else
      if $sup(\mathcal{L}[k-1] \le F_i^n < \tau_j) > \theta$ and $sup(\tau_j \le F_i^n < \mathcal{L}[k]) > \theta$ **then**
        Insert τ_j to $\mathcal{L}$ at position k
      end if
    end if
  end for
  for $j = 1, \ldots, |\mathcal{L}|$ **do**
    if $j = 1$ **then**
      Generate predicate $p : F_i^n < \mathcal{L}[j]$ and add p to $\mathcal{P}$
    end if
    if $1 < j \le |\mathcal{L}|$ **then**
      Generate predicate $p : \mathcal{L}[j-1] \le F_i^n < \mathcal{L}[j]$ and add p to $\mathcal{P}$
    end if
    if $j = |\mathcal{L}|$ **then**
      Generate predicate $p : F_i^n \ge \mathcal{L}[j]$ and add p to $\mathcal{P}$
    end if
  end for
end for
return $\mathcal{P}$

Importantly, finding such top-k PARs for an arbitrary anomaly instance x can be done rather efficiently. We sort all the PARs in $\mathcal{A}$ by their accuracy scores in a descending order in advance. Then, for an arbitrary identified anomaly instance x, we only need to sequentially traverse the sorted PARs in $\mathcal{A}$ until k violated PARs are found.

4.3 Hyperparameter Settings

As noticed, we allow only one predicate in the right-hand side of a PAR. We outline the reason for doing this as follows: suppose there are three rules, A_1: $P \longrightarrow p_1$, A_2: $P \longrightarrow p_2$ and A_3: $P \longrightarrow \{p_1, p_2\}$. Then, due to the anti-monotone constraint, we have $sup(A_1) = \frac{\#(P \cup p_1)}{|\mathcal{D}|} \geq \frac{\#(P \cup \{p_1, p_2\})}{|\mathcal{D}|} = sup(A_3)$ and $conf(A_1) = \frac{sup(A_1)}{sup(P)} \geq \frac{sup(A_3)}{sup(P)} = conf(A_3)$, likewise $sup(A_2) \geq sup(A_3)$ and $conf(A_2) \geq conf(A_3)$. As a result, according to the accuracy score defined in the previous section, if the data satisfy P and violates p_1 and/or p_2, then we prefer to pick A_1 and/or A_2 for anomaly explanation as they must have higher accuracy scores than A_3. Therefore, *it is redundant to generating PARs with more than one predicate in the right-hand.* Regarding the maximum number of predicates in the left-hand side of a PAR, we limit it to four to make derived PARs not over-complicated for users.

Regarding the minimum confidence threshold γ and the minimum support threshold θ in the PAR mining step, they define the lower limit for the confidence and support required for a PAR to be eligible for selection during the anomaly explanation process. Our algorithm is capable of identifying PARs with highest support and confidence using the accuracy score as the criterion for PAR selection. As a result, we recommend to set γ and θ to values less than which the found PARs are deemed to be entirely useless. Regarding λ, the importance weight for confidence, we recommend to set it to a value larger than one. This is because we generally prioritize high precision over high coverage of data for explaining anomalies, thus the confidence of PARs is much more important than their support in terms of facilitating accurate anomaly explanation. Throughout our experimentation, we fixed these hyperparameters to reasonable values $\theta = \max(10/|\mathcal{D}|, 0.01)$, $\gamma = 0.9$, $\lambda = 5$, and achieved robust results. Note that by setting λ to 5, we assign nearly identical accuracy score to a PAR with 100% confidence and minimum support and another PAR with 98% confidence and 100% support. This weighting reflects our preference for confidence over support as the primary factor in selecting PARs.

5 Showcase and User Study

To demonstrate the informativeness and intuitivenenss of PARs for anomaly explanation, we apply PARs to explain detected anomalies in SWAT, a dataset collected from a real-world water treatment testbed [19]. The derived PARs for explaining anomalies can be well understood by the system operators thus can significantly improve anomaly diagnosis efficiency. We present selected PARs for explaining three example anomaly instances with labeled ground-truth abnormal features as below:

- Anomaly instance 1:
  - Ground-truth abnormal features: MV101
  - Number of found PARs for explaining the anomaly instance: 1
  - Top-1 selected PAR for anomaly explanation:
    $$LIT101 \geq 811.18 \longrightarrow MV101 = 1$$
  - Translation: If the reading of level indicator LIT101 is larger than 811.18, then the state of valve MV101 should be 1, however, MV101 is not in state 1. This means that MV101 is suspected to be abnormal.
- Anomaly instance 2:
  - Ground-truth abnormal features: AIT202, P203
  - Number of found PARs for explaining the anomaly instance: 2
  - Top-1 selected PAR for anomaly explanation:
    $$\emptyset \longrightarrow 8.21 \leq AIT202 \leq 8.84$$
  - Translation: The reading of AIT202 should be in the range between 8.21 and 8.84, the current reading of AIT202 is not within the correct range. This means AIT202 is suspected to be abnormal.
  - Top-2 selected PAR for anomaly explanation:
    $$P205 = 1 \longrightarrow P203 = 1$$
  - Translation: If pump P205 is in state 1, then pump P203 should also be in state 1. However, pump P203 is not in state 1. This means P203 is suspected to be abnormal.
- Anomaly instance 3:
  - Ground-truth abnormal features: LIT101
  - Number of found PARs for explaining the anomaly instance: 1
  - Top-1 selected PAR for anomaly explanation:
    $$\begin{aligned} &AIT202 < 8.50, \\ &MV101 = 2, \\ &188.16 \leq PIT503 \leq 189.43, \\ &824.64 \leq LIT301 \leq 940.62 \\ &\longrightarrow LIT101 < 537.59 \end{aligned}$$
  - Translation: If sensor reading AIT202 is less than 8.5, valve MV101 is in state 2, sensor reading PIT503 is between 188.16 and 189.43, sensor reading LIT301 is between 824.64 and 940.62, then sensor reading for LIT101 should be less than 537.59. However, LIT101 is larger than 537.59. This means LIT101 is suspected to be abnormal.

We also conducted a user study with 30 participants who use anomaly detection systems regularly in their work for different applications including equipment predictive maintenance, process condition monitoring and network intrusion detection. The sole objective of the user study is to investigate which anomaly explanation form is more useful to users under the assumption that all the explanations are accurate. Thus, we presented the users with various forms of anomaly explanations as shown in Table 1, and asked them two questions: 1) "Please rank the usefulness of different explanation forms in the table", and 2) "Please provide reasons why you rank the particular explanation form as the best one". In the end, PAR was selected as the most useful form by 24 users for mainly two reasons: 1) rule-based format of PARs is more intuitive and understandable than other options, 2) PARs provide concrete

Table 2: The average rankings of the usefulness of explanation forms provided by different methods in the user study.

Method	SOAM	COIN	SHAP	LIME	Anchor	PAR
Rank	5.3	4.1	4.1	3.8	2.3	**1.4**

Table 3: Details of benchmark datasets

Dataset	# Features	# Samples
breastw	9	683
cardio	21	1831
Cardiotocography	21	2114
fault	27	1941
Ionosphere	32	351
Lymphography	18	148
magic	10	19020
Pima	8	768
satellite	36	6435
satimage-2	36	5803
shuttle	9	49097
skin	3	245057
Stamps	9	340
thyroid	6	3772
WBC	9	223
WDBC	30	367
wine	13	129

information about the suspected abnormal feature. The average rankings for all algorithms are presented in Table 2, which shows that users generally prefer rule-based anomaly explanations, such as PAR and Anchor, over other alternatives.

6 Experiments

In our experiments, two popular algorithms, Isolation Forecast (IF) [13] and Autoencoder (AE) [30], are selected as our representative anomaly detection models. The ADBench [7] is used as our benchmark datasets. Concretely, each tabular dataset in ADBench is split into a training set and a test set at a 4:1 ratio. We then train IF and AE models on the training set and tune anomaly thresholds for both models to achieve the best F_1 score on the testing set. We select all datasets on which both models can achieve at least 0.5 F_1 score on the testing set as our benchmark datasets for further experiments. This leads to 17 datasets being selected. The details of the select benchmark datasets are given in Table 3.

6.1 Efficiency of Finding Explanation Rules

We first investigate how efficient to find PARs for anomaly explanation. The time cost is consisting of two stages: the training stage and the inference stage. *The training stage is done in advance and offline thus will not affect user experience for real-time anomaly explanation.* For the offline training stage, the time cost of predicate generation increases linearly with the number of features in the dataset. The efficiency of the FP-growth mining algorithm is impacted by the quantity of predicates and the frequency distribution of these predicates. Unfortunately, a formal big O-style formulation of the efficiency of FP-growth is unavailable. Our experimental results indicate that the training stage time cost ranges from approximately 1 second to 5 minutes across different datasets. In practice, parallel mining may be employed for extremely high-dimensional datasets by limiting the number of predicates in each mining process to a tolerable quantity, and combining the results of parallel processes during the reduction step. For the inference stage, to find the top-k violated PARs for an anomaly, we only need to sequentially traverse the sorted PARs until k violated PARs are found. As a result, the time complexity for the best case is $O(k)$, for the worst case is $O(N)$ where N is the total number of generated PARs in the training stage.

Table 4: The average time cost to compute the anchors and top 5 PARs at each anomaly instance in the benchmark datasets identified by IF and AE models, and the average precision, recall and F_1 score of using anchors and top 5 PARs computed at each anomaly instance to do anomaly detection in unseen data.

	Anchor	PAR	Improvement
Avg. time cost (secs)	25.71	0.22	-99%
Avg. precision	0.55	0.67	+22%
Avg. recall	0.21	0.39	+86%
Avg. F_1 score	0.23	0.42	+83%

To illustrate the the efficiency of PARs for anomaly explanation, anchor, which also provides rule-based anomaly explanation, is used as our baseline for comparison. We report the average time cost for computing the top 5 PARs and the anchor at each anomaly instance identified by IF and AE on the benchmark datasets in Table 4. Note that due to lack of space, we only report the average metric across all the benchmark datasets in the table, and this also applies for all experiment results hereafter. As can be seen from the table, the average time cost of finding PARs is less than one second, whereas the average time cost for finding anchors is more than 25 seconds. This high computing efficiency makes PARs far more suitable for online anomaly explanation compared with Anchors.

6.2 Accuracy of Explanation Rules

We then compare the accuracy of PARs to anchors. According to [20], the accuracy of an explanation is defined as "How well does an explanation predict unseen data?". Thus for each dataset, to evaluate the accuracy of PARs and anchors, we use the top 5 PARs and the anchor computed for explaining a single anomaly instance as a anomaly detection model to detect anomalies in the remaining part of test set. We report the average performance, in terms of precision, recall and F_1 score, of using anchors and top 5 PARs computed at each identified anomaly instance to detect anomalies in unseen data in Table 4. As shown in the table, the selected PARs are much more accurate than anchors for anomaly explanation since on average PARs achieve 22% improvement on precision, 86% improvement on recall and 83% improvement on F_1 score compared with anchors.

Table 5: Accuracy of abnormal feature identification in terms of HitRate@100% and HitRate@150%.

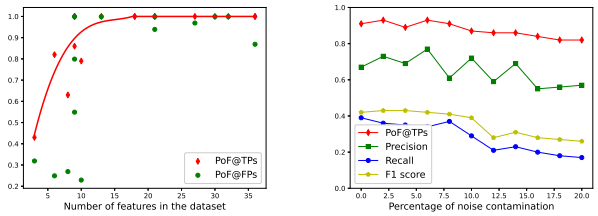
	HitRate@100%		HitRate@150%	
	Avg.	Avg. Rank	Avg.	Avg. Rank
LIME	0.39	4.47	0.48	4.35
SHAP	0.59	2.24	0.68	**2.12**
COIN	0.54	3.53	0.65	3.18
ATON	0.62	2.53	**0.71**	2.47
PAR	**0.66**	**2.00**	0.70	2.53

6.3 Accuracy of Abnormal Feature Identification

In this subsection, we study PARs' accuracy of abnormal feature identification compared with other state-of-the-art model-agnostic abnormal feature identification methods including LIME, SHAP, COIN and ATON. Specifically, for each identified anomaly, we output the features in the consequent predicates of the top 5 PARs as the suspected abnormal feature list. Since LIME, SHAP, COIN and ATON can output a feature weight vector per identified anomaly indicating the suspected anomaly contribution, we output a list of suspected abnormal features sorted by their anomaly contribution weights in a descending order for these methods.

Datasets preparation. Evaluating the accuracy of abnormal feature identification requires benchmark datasets with ground-truth annotations of abnormal feature subspace. To the best of our knowledge, there is no publicly available real-world tabular dataset with such annotations. As a result, we propose to use the 17 selected benchmark datasets for experiments. Specifically, we create ground-truth annotations of abnormal features for the benchmark datasets by randomly perturbing 1 to 3 features for the normal instances in the testing set. The perturbed features at each instance are labeled as abnormal features. We only apply different methods to identify abnormal features for perturbed data which are actually identified as anomalies by IF and AE models in our experiments.

Evaluation metrics and results. Similar to [33], we borrow the idea of HitRate@K for recommender systems [37] as the metrics to evaluate the accuracy of abnormal feature identification. Specifically, we define a metric HitRate@P%= $\frac{Hit@\lfloor P\% \times |GF| \rfloor}{|GF|}$ where $|GF|$ is the size of ground-truth abnormal features and P can be 100 or 150. HitRate@P% measures the number of overlapping features between ground-truth abnormal features and the top $\lfloor P\% \times |GF| \rfloor$ suspected abnormal features suggested by the anomaly explanation methods. For example, if ground-truth abnormal features are $\{2, 6\}$ and the suspected abnormal feature list is [2, 3, 6, 1, 5, 4], the result is 0.5 for HitRate@100% and 1.0 for HitRate@150%. In Table 5, we report the metrics HitRate@100% and HitRate@150% for LIME, SHAP, COIN, ATON and PAR on the benchmark datasets. As can be seen from the table, PAR achieves the highest average HitRate@100% and the second highest HitRate@150% which demonstrate its high accuracy on abnormal feature identification.



(a) (b)

Figure 2: (a): PoF@TPs and PoF@FPs for datasets with different number of features. (b): PoF@TPs and average accuracy of PARs with different percentage of noise contamination in the training data.

6.4 Probability of Finding PARs for Anomalies

In this subsection, we study the probability of finding at least one PAR (PoF for abbreviation) for explaining identified anomalies using our method. Concretely, we report PoF@TPs (PoF for True Positives) and PoF@FPs (PoF for False Positives) for predicated anomalies by IF and AE models on the benchmark datasets in Figure 2a. From the figure, we make two important observations: 1) PoF is discernibly higher when the predicted anomalies are TPs than when the predicted anomalies are FPs. This is desirable since we prefer to find PARs for explaining TPs. With respect to FPs, failure to finding a PAR for them actually gives the chance to mitigate their negative impact. 2) PoF is closely related to the feature dimension of datasets. This is as expected because we rely on the dependency between different features to construct PARs, thus the difficulty in finding PARs grows with fewer feature dimension. However, the good news is that we find PoF@TPs in 12 out of 17 datasets achieves 100%. Moreover, there is a clear trend that when the feature dimension of dataset is larger than 10, PoF@TPs quickly converges to 100% as shown in Figure 2a.

6.5 Impact of Noise Contamination

To evaluate the robustness of our approach with respect to noise (anomaly) contamination in the training data, we measure PoF@TPs and the accuracy of selected PARs when setting noise proportion within training data at levels ranging from 0% to 20%. Specifically, we report PoF@TPs, the average precision, recall and F_1 score of applying top-5 PARs for explaining a single detected anomaly instance to anomaly detection on unseen data in Figure 2b. As can be seen, the impact of noise contamination on both PoF@TPs and the accuracy of selected PARs for anomaly explanation is limited when the noise contamination is less than 10%, implying that PARs are relatively robust to noise contamination in real-world anomaly detection scenarios. However, when the proportion of noise contamination exceeds 10%, which is an uncommon occurrence in real-world applications, the recall of selected PARs is primarily affected by the noise. This is likely due to the fact that the noise level has surpassed a certain threshold, causing a number of PARs to be unable to meet the minimum confidence threshold. This hypothesis is consistent with the decreasing trend of PoF@TPs as

Table 6: The PoF for explaining predicted TPs and the average accuracy score of the top1 PAR, when using different methods to generate predicates for numeric features.

	PoF@TPs	Accuracy Score
Uniform Bins	0.83	4.99
KMeans Bins	0.84	5.12
Dependency-based	0.91	5.37

the proportion of noise contamination increases beyond 10%. Nevertheless, it is important to note that the precision of PARs still remains at a relatively high level, indicating that selected PARs remain rather reliable as they do not generate an excessive amount of false positives compared to PARs learned from noise-free data.

6.6 Ablation Study

We also demonstrate the importance of our dependency-based predicate generation method for numeric features in this subsection. Concretely, we replace our dependency-based predicate generation method for numeric features with an uniform interval-based and a KMeans-based discretization method which simply discretize the value of numeric features to 10 bins for predicate generation. Then, we generate PARs for the benchmark datasets based on three different predicate generation method for numeric features, namely Uniform Bins, KMeans Bins and dependency-based. Furthermore, we compare two important metrics on the performance of anomaly explanation: PoF for TPs, the average accuracy score of the top1 PAR. The result is given in Table 6. As can be seen, there are discernible improvements on both metrics when using our dependency-based method for predicate generation of numeric features compared with Uniform Bins and KMeans Bins methods.

7 Conclusion

We introduced Predicate-based Association Rules (PARs), a novel and intuitive form of model-agnostic anomaly explanation. PARs not only highlight the suspected abnormal features, but also the reasons behind their abnormality. Our user study shows that PARs are better understood and preferred by regular anomaly detection system users compared with existing model-agnostic explanation options. We demonstrated the efficiency and the accuracy of PARs for anomaly explanation on various benchmark datasets. As people have increasingly highlighted the importance and imperativeness on providing tangible explanations in the anomaly detection field [23, 28], PARs can make a highly valuable practical contribution to this domain.

A Implementation Details for Experiments

Regarding baseline anomaly explanation methods, we implement Anchor based on the Github repository in github.com/marcotcr/anchor. We set its hyperparameters $B = 10$, $\epsilon = 0.1$, $\delta = 0.05$ as suggested in the original paper [27]. We implement LIME based on the Github repository in github.com/marcotcr/lime. We implement SHAP based on the Github repository in github.com/slundberg/shap. Specifically, the KernelExplainer is used with 20 samples generated by KMeans as background dataset for integrating out features. COIN and ATON are implementated based on the Github repository in github.com/xuhongzuo/outlier-interpretation. For all methods, the default hyperparameter values are used unless specifically mentioned here.

Regarding Isolation Forest (IF) and Autoencoder (AE), we implement IF using Scikit-Learn [25] library of version 1.1.2. The default hyperparameter values are used. We implement a vanilla version of AE with three hidden layers, the number of neurons in the bottleneck layer is set to $\frac{1}{3}$ of the input dimension. Each model is trained with 10 epochs to minimize the reconstruction error on the training set. All of our experiments were run on a Linux machine with 64 GiB memory, 8 4.2GHz Intel Cores and a GTX 1080 GPU.

References

[1] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. 1993. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*. 207–216.

[2] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 93–104.

[3] Lei Duan, Guanting Tang, Jian Pei, James Bailey, Akiko Campbell, and Changjie Tang. 2015. Mining outlying aspects on numeric data. *Data Mining and Knowledge Discovery* 29, 5 (2015), 1116–1151.

[4] Cheng Feng, Venkata Reddy Palleti, Aditya Mathur, and Deeph Chana. 2019. A Systematic Framework to Generate Invariants for Anomaly Detection in Industrial Control Systems. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society.

[5] Cheng Feng and Pengwei Tian. 2021. Time series anomaly detection for cyber-physical systems via neural system identification and bayesian filtering. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2858–2867.

[6] Jiawei Han, Jian Pei, and Yiwen Yin. 2000. Mining frequent patterns without candidate generation. *ACM sigmod record* 29, 2 (2000), 1–12.

[7] Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. 2022. Adbench: Anomaly detection benchmark. *Advances in Neural Information Processing Systems* 35 (2022), 32142–32159.

[8] Xu Han, Xiaohui Chen, and Li-Ping Liu. 2021. Gan ensemble for anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4090–4097.

[9] Sahand Hariri, Matias Carrasco Kind, and Robert J Brunner. 2019. Extended isolation forest. *IEEE Transactions on Knowledge and Data Engineering* 33, 4 (2019), 1479–1489.

[10] Sheng-yi Jiang and Qing-bo An. 2008. Clustering-based outlier detection method. In *2008 Fifth international conference on fuzzy systems and knowledge discovery*, Vol. 2. IEEE, 429–433.

[11] Zhong Li, Yuxuan Zhu, and Matthijs Van Leeuwen. 2023. A survey on explainable anomaly detection. *ACM Transactions on Knowledge Discovery from Data* 18, 1 (2023), 1–54.

[12] Boyang Liu, Pang-Ning Tan, and Jiayu Zhou. 2022. Unsupervised Anomaly Detection by Robust Density Estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[13] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2012. Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 6, 1 (2012), 1–39.

[14] Ninghao Liu, Donghwa Shin, and Xia Hu. 2018. Contextual outlier interpretation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 2461–2467.

[15] Sha Lu, Lin Liu, Jiuyong Li, Thuc Duy Le, and Jixue Liu. 2020. Lopad: A local prediction approach to anomaly detection. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 660–673.

[16] Stefan Lüdtke, Christian Bartelt, and Heiner Stuckenschmidt. 2023. Outlying Aspect Mining via Sum-Product Networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 27–38.

[17] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).

[18] Larry M Manevitz and Malik Yousef. 2001. One-class SVMs for document classification. *Journal of machine Learning research* 2, Dec (2001), 139–154.

[19] Aditya P Mathur and Nils Ole Tippenhauer. 2016. SWaT: A water treatment testbed for research and training on ICS security. In *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*. IEEE, 31–36.

[20] Christoph Molnar. 2020. *Interpretable machine learning*. Lulu. com.

[21] Raymond T Ng, Laks VS Lakshmanan, Jiawei Han, and Alex Pang. 1998. Exploratory mining and pruning optimizations of constrained associations rules. *ACM Sigmod Record* 27, 2 (1998), 13–24.

[22] Koyena Pal, Sridhar Adepu, and Jonathan Goh. 2017. Effectiveness of association rules mining for invariants generation in cyber-physical systems. In *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*. IEEE, 124–127.

[23] Guansong Pang and Charu Aggarwal. 2021. Toward explainable deep anomaly detection. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 4056–4057.

[24] Heiko Paulheim and Robert Meusel. 2015. A decomposition of the outlier detection problem into a set of supervised learning problems. *Machine Learning* 100, 2 (2015), 509–531.

[25] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.

[26] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

[27] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[28] Lukas Ruff, Jacob R. Kauffmann, Robert A. Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G. Dietterich, and Klaus-Robert Müller. 2021. A Unifying Review of Deep and Shallow Anomaly Detection. *Proc. IEEE* 109, 5 (2021), 756–795.

[29] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In *International conference on machine learning*. PMLR, 4393–4402.

[30] Mayu Sakurada and Takehisa Yairi. 2014. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*. 4–11.

[31] Durgesh Samariya, Sunil Aryal, Kai Ming Ting, and Jiangang Ma. 2020. A new effective and efficient measure for outlying aspect mining. In *Web Information Systems Engineering–WISE 2020: 21st International Conference, Amsterdam, The Netherlands, October 20–24, 2020, Proceedings, Part II 21*. Springer, 463–474.

[32] Durgesh Samariya, Jiangang Ma, and Sunil Aryal. 2020. A comprehensive survey on outlying aspect mining methods. *arXiv preprint arXiv:2005.02637* (2020).

[33] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. 2019. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2828–2837.

[34] Nguyen Xuan Vinh, Jeffrey Chan, Simone Romano, James Bailey, Christopher Leckie, Kotagiri Ramamohanarao, and Jian Pei. 2016. Discovering outlying aspects in large datasets. *Data Mining and Knowledge Discovery* 30, 6 (2016), 1520–1555.

[35] Hongzuo Xu, Yijie Wang, Songlei Jian, Zhenyu Huang, Yongjun Wang, Ning Liu, and Fei Li. 2021. Beyond outlier detection: Outlier interpretation by attention-guided triplet deviation network. In *Proceedings of the Web Conference 2021*. 1328–1339.

[36] Takehisa Yairi, Yoshikiyo Kato, and Koichi Hori. 2001. Fault detection by mining association rules from house-keeping data. In *proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Vol. 18. Citeseer, 21.

[37] Xiwang Yang, Harald Steck, Yang Guo, and Yong Liu. 2012. On top-k recommendation using social networks. In *Proceedings of the sixth ACM conference on Recommender systems*. 67–74.

[38] Houssam Zenati, Manon Romain, Chuan-Sheng Foo, Bruno Lecouat, and Vijay Chandrasekhar. 2018. Adversarially learned anomaly detection. In *2018 IEEE International conference on data mining (ICDM)*. IEEE, 727–736.

[39] Chong Zhou and Randy C Paffenroth. 2017. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 665–674.

[40] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. 2018. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*.