Achieving Counterfactual Explanation for Sequence Anomaly Detection

He Cheng[1], Depeng Xu[2], Shuhan Yuan[1(✉)], and Xintao Wu[3]

[1] Utah State University, Logan, UT, USA
{he.cheng,shuhan.yuan}@usu.edu
[2] University of North Carolina at Charlotte, Charlotte, NC, USA
depeng.xu@uncc.edu
[3] University of Arkansas, Fayetteville, AR, USA
xintaowu@uark.edu

Abstract. Anomaly detection on discrete sequential data has been investigated for a long time because of its potential in various applications, such as detecting novel attacks or abnormal system behaviors from log messages. Although many approaches can achieve good performance on anomalous sequence detection, how to explain the detection results is still challenging due to the discrete nature of sequential data. Specifically, given a sequence that is detected as anomalous, the explanation is to highlight those anomalous entries in the sequence leading to the anomalous outcome. To this end, we propose a novel framework, called CFDet, that can explain the detection results of one-class sequence anomaly detection models by highlighting the anomalous entries in the sequences based on the idea of counterfactual explanation. Experimental results on three datasets show that CFDet can provide explanations by correctly detecting anomalous entries.

Keywords: anomaly detection · counterfactual explanations · sequential data

1 Introduction

Anomaly detection (AD) on discrete sequential data has received a lot of attention recently because of wide applications, such as detecting anomalous log sequences or user activity sequences [8,20,23,26]. For example, log messages generated by computing systems are critical resources for debugging the abnormal patterns of systems or detecting novel attacks. Identifying the anomalous log sequences generated by computing systems promptly is important to building stable systems [8,26].

However, the current approaches mainly focus on improving the performance of sequence anomaly detection while not much work targets explaining the detection results. In practice, given detected anomalous sequences, the domain users must understand why the detection model made such predictions. Providing

A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14948, pp. 19–35, 2024.
https://doi.org/10.1007/978-3-031-70371-3_2

explanations on the detected anomalous sequence can help the domain users locate the exact issues. For example, anomalous sequence detection is commonly used to identify potential attacks against computer systems. In the security operation center, the security analytics would receive thousands of alerts per day, and manually investigating all the alerts requires a large amount of effort, often leading to the issue of "alert fatigue" and missing real security alerts [1]. On the other hand, highlighting the anomalous entries in the sequences can help the security analysts understand the security alerts by only checking the highlighted anomalous entries instead of inspecting the whole sequence, which could massively reduce the workload for analysts [3].

In this work, we propose a counterfactual explanation framework, called CFDet, which can provide explanations for the commonly used one-class anomaly detection models, deep support vector data description (Deep SVDD) [18] and its variance, OC4Seq [23], by identifying the anomalous entries from a detected anomalous sequence. The idea of counterfactual explanation is to identify the "minimum change" to anomalous sequences that could change the prediction to normal. Here, the changes indicate anomalous entries, because by removing the anomalous entries, we can "change" a sequence from anomalous to normal. Then, we can highlight the anomalous entries in a sequence as explanations.

The main contributions of this paper are as follows. First, we propose CFDet, a novel counterfactual explanation framework for one-class sequence anomaly detection approaches. Second, CFDet can highlight the anomalous entries in the detected anomalous sequences. Third, the experimental results on three datasets show that CFDet can provide high-fidelity explanations by accurately detecting anomalous entries in anomalous sequences.

2 Related Work

Anomaly Detection in Sequential Data. Sequential anomaly detection plays an important role in a wide spectrum of application scenarios. Due to a limited number of anomalies, many unsupervised or one-class deep learning approaches are proposed to detect anomalous sequences by identifying the differences between normal and anomalous patterns [8,9,26]. A typical idea is to make use of recurrent neural networks (RNNs) or Transformer to capture the normal patterns from normal sequences. Then, an anomalous sequence can be detected with deviating patterns [8,26]. For example, DeepLog [8] is trained to predict the log entry by an RNN model based on a large number of normal sequences. The anomalous sequence can then be detected when the RNN cannot correctly predict the log entries, meaning the sequence does not follow the normal patterns. However, the majority of approaches only focus on detecting anomalous sequences and cannot point out fine-grained subsequences or entries in the sequences that lead to anomalous outcomes.

Explainable Anomaly Detection. Explainability in machine learning is crucial for high-stakes decisions and troubleshooting. Explainable machine learning

techniques can be categorized into two types, intrinsic explainability and post-hoc explainability [5]. Intrinsic explainability indicates self-explanatory models that achieve explainability directly based on their structures, while post-hoc explainability means that explainability is achieved by applying another model to provide explanations.

Only a few studies target the task of explainable anomaly detection, especially the sequence anomaly detection [4,6,7,12,14]. To achieve intrinsic explanations, the explainable deep one-class classification model [14] provides intrinsic explainability for anomaly detection on image data but cannot identify the discrete anomalous entries in sequences. Meanwhile, the attention mechanism, which provides intrinsic interpretation based on the attention weights, is adopted for detecting anomalous events from sequential data [4]. However, the attention scores derived in the proposed approach indicate the contributions to predicting the next event in the sequence and are not strictly related to the anomalous outcome. Some studies achieve post-hoc explanations based on the perturbation-based or gradient-based interpretation approaches. Research in [2] develops explainable autoencoder models to identify features leading to high reconstruction errors using Shapley values. Similarly, research in [16] adopts a variational autoencoder as the anomaly detection model and identifies important features based on the gradient values. The closest to our work is DeepAID [11], which also identifies anomalous entries as explanations. DeepAID first leverages DeepLog to detect the anomalous sequences and propose a gradient-based approach to identify the abnormal entries in the sequences. In this work, we leverage the idea of counterfactual interpretation to explain the Deep SVDD-based sequence anomaly detection approaches, which can provide human-understandable post-hoc explanations for anomalous sequence detection results. Our experimental results show that CFDet significantly outperforms DeepAID.

3 Preliminary: Deep SVDD-Based Approaches for Anomalous Sequence Detection

Given a set of normal sequences $\mathcal{P} = \{S_n^+\}_{n=1}^N$, where S indicates a sequence, we derive an anomaly detection model based on Deep SVDD [18]. First, a recurrent neural network, such as long short-term memory (LSTM) neural network, is adopted to derive the representation of sequence S_n^+ as $\mathbf{r}_n^+ = f(S_n^+)$, where $f(\cdot)$ indicates the LSTM model and $\mathbf{r}_n^+$ is the last hidden state of LSTM. Deep SVDD aims at making the normal data close to the center of the hypersphere consisting of normal samples. We derive the center $\mathbf{c}$ of normal sequences by a mean operation, i.e., $\mathbf{c} = Mean(\mathbf{r}_n^+)$. The objective function of Deep SVDD is:

$$\mathcal{L}_{SVDD} = \frac{1}{N}\sum_{n=1}^{N}||f(S_n^+;\Theta) - \mathbf{c}||_2^2 + \lambda_{SVDD}||\Theta||_F^2, \tag{1}$$

where Θ denotes the parameters in the LSTM model. Deep SVDD employs a quadratic loss for penalizing distances of normal data representations to the cen-

ter $\mathbf{c}$. Therefore, it can jointly learn the LSTM model together with minimizing the volume of a data-enclosing hypersphere in the latent space.

After training, we can deploy the LSTM model $f(\cdot)$ to detect the anomalous sequences in the unlabeled set $\mathcal{U}$. Given a sequence $S^* \in \mathcal{U}$ and its representation $\mathbf{r}^* = f(S^*)$, the anomaly score is the distance of a sequence to the center:

$$s(\mathbf{r}^*) = ||\mathbf{r}^* - \mathbf{c}||_2^2. \tag{2}$$

If the representation of the sequence $\mathbf{r}^*$ falls outside the hypersphere, i.e., $s(\mathbf{r}^*) > \epsilon$, where ϵ indicates the threshold, we will label the sequence S^* as anomalous. We then obtain a set of sequences $\tilde{\mathcal{U}}^-$ that are detected as anomalous from $\mathcal{U}$.

OC4Seq [23] is an extension of Deep SVDD. Besides learning the representations for the entire sequence, OC4Seq further ensures the subsequences are close to a local center, which can improve anomaly detection. The objective of OC4Seq is defined as a combination of the original SVDD loss and an additional term that emphasizes the normality of local subsequence representation, denoted as $\mathcal{L}_{OC4Seq} = \mathcal{L}_{SVDD} + \lambda\mathcal{L}_{local}$, where λ is a hyperparameter, and $\mathcal{L}_{local}$ is the loss to train the representation of subsequences, $\mathcal{L}_{local} = \frac{1}{N}\sum_{n=1}^{N}\sum_{w=1}^{W}||f_{local}(S_n^w;\Theta_{local}) - \mathbf{c}_l||_2^2 + \lambda_{OC4Seq}||\Theta_{local}||_F^2$, where S_n^w indicates the w-th subsequence derived from S_n based on a small sliding window leading to W subsequences in total; $\mathbf{c}_l$ is the center of the hypersphere corresponding to subsequences in the latent space, and $f_{local}(\cdot)$ is another LSTM model parameterized by Θ_{local}. The anomaly score is derived as the sum of distances between the sequence and corresponding subsequences to the global and local centers.

4 Framework

4.1 Overview

In this work, we consider the following real-world scenario. An IT team trains a Deep SVDD-based sequence anomaly detection model to identify the abnormal behavior in its computer system. After deploying, it is expected that several sequences will be labeled as anomalous. However, it is hard to diagnose the problem by only providing a simple alarm without any evidence. Security analysts expect more information to validate the security alerts efficiently. To address this issue, we propose a novel framework (shown in Fig. 1), CFDet, that can provide explanations by identifying anomalous entries for each detected anomalous sequence $Z \in \tilde{\mathcal{U}}^-$. In particular, CFDet trains an anomalous entry detector $g(\cdot)$ to identify the anomalous entries based on the idea of counterfactual explanations.

Key Idea of Counterfactual Explanation. Counterfactual explanations describe a situation in the form: "If X had not occurred, Y would not have occurred" [15]. In the anomaly detection scenario, we can rephrase the above

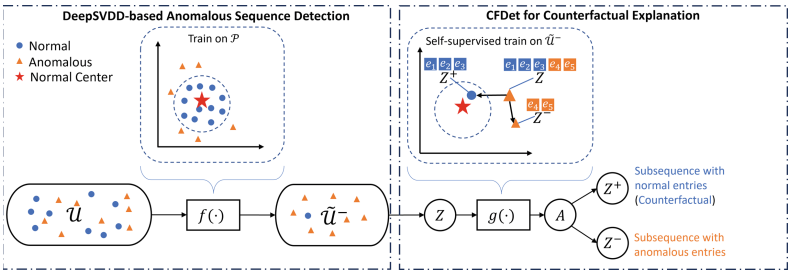


Fig. 1. Illustration of counterfactual explanation achieved by CFDet

statement as if there had been no anomalous entries in a sequence, the sequence would not be anomalous.

We first denote a detected anomalous sequence with length L as $Z = \{e_l\}_{l=1}^{L}$, where e_l indicates the l-th entry. We use e^+ and e^- to denote normal and anomalous entries respectively. We then denote the subsequence consisting of only anomalous entries in Z as Z^-. Formally, we have

$$Z^- = A \odot Z, \tag{3}$$

where $A = \{a_l\}_{l=1}^{L}$ is an indicator sequence with each $a_l \in \{0, 1\}$ being a binary indicator, and $\odot$ indicates the element-wise product. If the entry is an anomalous one e_l^-, the corresponding indicator $a_l = 1$; otherwise, $a_l = 0$. Similarly, we can also derive its subsequence with normal entries as Z^+, i.e., $Z^+ = (1 - A) \odot Z$.

Following the notion of counterfactual explanation, we aim at identifying and removing all the anomalous entries from the sequence Z. Once the anomalous entries are removed from the sequence, the corresponding counterfactual sequence should be normal and is the normal subsequence Z^+. Therefore, our goal for counterfactual explanation is to learn an anomalous entry detector $g(\cdot) : Z \rightarrow A$ that can properly generate the indicator sequence A.

Properties of Counterfactual Sequences. An ideal counterfactual sequence of an anomalous sequence should satisfy the following properties.

- **Normality:** The counterfactual sequence Z^+ should be normal. As illustrated in [19], by assuming that the latent representations derived by $f(\cdot)$ follow an isotropic Gaussian distribution, Deep SVDD is to minimize the upper bound on the entropy of the Gaussian. Therefore, the hidden representations of counterfactual sequences should have minimal entropy:

$$\mathcal{H}(f(Z^+)) \leq \tau, \tag{4}$$

  where $\mathcal{H}(\cdot)$ indicates the entropy and τ is a constant.
- **Comprehensiveness:** After removing Z^+ from Z, Z^- should be abnormal:

$$\mathcal{H}(f(Z^-)) \geq \mathcal{H}(f(Z^+)) + t, \tag{5}$$

where t is a constant margin. It requires Z^- as anomalous entries in a sequence Z should have a higher entropy with a margin compared with the counterfactual sequence. Therefore, we can ensure that Z^+ includes all the normal entries while Z^- has all the anomalous entries.

- **Conciseness:** The counterfactual should have the minimum changes on the original one, so the anomalous entries Z^- should be consecutive and sparse,

$$\sum_l \|a_l - a_{l-1}\| \leq c, \quad \sum_l a_l \leq s, \tag{6}$$

where c and s are both constants.

Both Z^+ and Z^- can be derived based on the indicator sequence A that is generated by the anomalous entry detector $g(\cdot)$. Therefore, given an anomalous sequence Z, the objective of training $g(\cdot)$ is to meet the above three properties of counterfactual sequences.

4.2 Achieving Counterfactual Explanation for Deep SVDD

We first use the Deep SVDD model as the sequence anomaly detection model to illustrate our counterfactual explanation approach and then extend to explain the detection results from OC4Seq.

Given an anomalous sequence $Z \in \tilde{\mathcal{U}}^-$, to generate the counterfactual with normal entries Z^+ as well as the subsequence with anomalous entries Z^-, we propose to train the anomalous entry detector $g(\cdot)$ to generate the indicator sequence A, i.e., $A = g(Z)$. We use another LSTM model as a part of the implementation of $g(\cdot)$. Based on the LSTM model, we can derive the hidden state $\mathbf{h}_l$ for each entry $e_l \in Z$. Then, we apply a logistic regression model $q(\cdot)$ on $\mathbf{h}_l$ to predict the probability p_l of the entry e_l as anomalous:

$$\mathbf{h}_l = LSTM(\mathbf{e}_l, \mathbf{h}_{l-1}) \quad p_l = q(\mathbf{h}_l), \tag{7}$$

where $\mathbf{e}_l$ denotes the representation of entry e_l. After rounding the probability p_l to the 0-1 binary value, we get the indicator a_l for the entry e_l:

$$a_l = \begin{cases} 1, & \text{if } p_l \geq 0.5 \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Equations 7 and 8 are the implementation of the anomalous entry detector $g(\cdot)$.

In order to train $g(\cdot)$ to accurately generate A so that the counterfactual sequences meet the properties of normality, comprehensiveness, and conciseness, we train $g(\cdot)$ on the detected anomalous sequence set $\tilde{\mathcal{U}}^-$ by the following objective function:

$$\mathcal{L} = \mathcal{L}_n + \alpha\mathcal{L}_t + \beta\mathcal{L}_c + \gamma\mathcal{L}_s, \tag{9}$$

where $\mathcal{L}_n$ is a Deep SVDD-based loss to ensure the normality of generated counterfactual sequences; $\mathcal{L}_t$ indicates the triplet loss that is to ensure the comprehensiveness; $\mathcal{L}_c$ indicates the continuity loss that is to ensure the continuity; $\mathcal{L}_s$

indicates the sparsity loss that is to ensure the generated counterfactual sample with the minimum change; α, β, and γ are hyperparameters to balance the weight of each loss term.

Normality Loss $\mathcal{L}_n$. To meet the property of normality about the counterfactual sequence defined in Eq. 4, given a detected anomalous sequence, the objective is to make the representation of counterfactual sequence Z^+ close to the center $\mathbf{c}$ (the center of the hypersphere with normal sequences). Therefore, we first derive the representations of counterfactual sequence Z^+ based on the LSTM model $f(\cdot)$ as $\mathbf{r}_{z^+} = f(Z^+)$. Then, the loss is defined as

$$\mathcal{L}_n = ||\mathbf{r}_{z^+} - \mathbf{c}||_2^2. \tag{10}$$

Triplet Loss $\mathcal{L}_t$. To meet comprehensiveness defined in Eq. 5, we also need to make sure Z^+ includes all normal entries so that Z^- only consists of anomalous entries. The idea is to make the representation of the subsequence with anomalous entries Z^- far from the center $\mathbf{c}$ while the representation of the counterfactual sequence Z^+ close to the center. To this end, we also derive the representation of anomalous entries Z^- by $f(\cdot)$, i.e., $\mathbf{r}_{z^-} = f(Z^-)$. Then, we consider the center $\mathbf{c}$ as an anchor, the representation of counterfactual $\mathbf{r}_{z^+}$ as a positive sample and the representation of the anomalous subsequence $\mathbf{r}_{z^-}$ as a negative sample. The triplet loss is adopted to ensure the comprehensiveness:

$$\mathcal{L}_t = \max\left\{||\mathbf{c} - \mathbf{r}_{z^+}||_2^2 - ||\mathbf{c} - \mathbf{r}_{z^-}||_2^2 + \lambda_t, 0\right\}, \tag{11}$$

where λ_t is a margin between positive and negative pairs. Intuitively, if the distance between $\mathbf{c}$ and $\mathbf{r}_{z^-}$ is larger than the distance between $\mathbf{c}$ and $\mathbf{r}_{z^+}$ with a margin, $\mathbf{r}_{z^-}$ should only have anomalous entries.

Continuity Loss $\mathcal{L}_c$. Meanwhile, the abnormal entries in a sequence are usually coherent. For example, if a system is under attack, the abnormal log entries are often consecutive. Hence, to ensure the generated indicator sequence A with consecutive selection on the abnormal entries, inspired by [25], we also incorporate the continuity loss:

$$\mathcal{L}_c = \max\{\sum_l \|a_l - a_{l-1}\| - \lambda_c, 0\}, \tag{12}$$

where λ_c is a hyperparameter that controls the continuity of the indicator sequence. Minimizing the continuity loss defined in Eq. 12 ensures the indicator sequence A with a minimum number of small pieces controlled by λ_c.

Sparsity Loss $\mathcal{L}_s$. The counterfactual explanation usually expects the "minimum" change on the original sample. In our scenario, we expect that removing the detected anomalous entries is just enough to change the anomalous sequence to a normal one. We do not want to remove the normal entries which could lead to false positive detection. Moreover, in most scenarios, the anomalous entries

should be sparse compared with the normal entries in a sequence. Hence, we also incorporate the sparsity loss in the loss function:

$$\mathcal{L}_s = \max\{\sum_l a_l - \lambda_s, 0\} \tag{13}$$

where λ_s is a hyperparameter that indicates the expectation of anomalous entries in the sequence. Minimizing the sparsity loss is to make the number of detected anomalous entries close to a pre-set value.

Training. Because the indicator sequence A is a sequence with binary values, the regular gradient descent algorithm cannot be used to optimize the anomalous entry detector $g(\cdot)$. Here, we use the policy gradient algorithm used in reinforcement learning [22] to train CFDet. To this end, we can consider the negative loss of the objective function in Eq. 9 as the reward function of a reinforcement learning model, and $g(\cdot)$ as an agent that takes an action $\{0, 1\}$ on an entry e_l based on the current state (i.e., the hidden representation of an entry $\mathbf{h}_l$). The anomalous entry detector is then trained to maximize the reward function.

4.3 Achieving Counterfactual Explanation for OC4Seq

We now discuss how to extend the above training objectives to explain the detection results from OC4Seq. The main difference between Deep SVDD and OC4Seq is that OC4Seq detects sequential anomalies from both global (whole sequence) and local (subsequence) levels. More specifically, OC4Seq constructs a global hypersphere and a local hypersphere to detect abnormal behaviors. We adopt the same structure defined in Eqs. 7 and 8 as the implementation of the anomalous entry detector $g(\cdot)$ for OC4Seq. Because OC4Seq also checks the normality in the subsequence level, after deriving the Z^+ and Z^- in the sequence level, we further derive the normal entries Z_w^+ and abnormal entries Z_w^- from w-th subsequence Z_w. To achieve a fidelity explanation, we mainly revise the normality loss $\mathcal{L}_n$ and triplet loss $\mathcal{L}_t$ to fit the training objective of OC4Seq.

First, we denote the center of subsequences as $\mathbf{c}_l$, and the representation of normal entries in the subsequence is defined as $\mathbf{r}_{z^+}^w = f_{local}(Z_w^+)$. Similar to $\mathcal{L}_n$ in Eq. 10, the normality loss can be revised as the combination of ensuring the normality in both whole sequence and subsequence levels:

$$\mathcal{L}'_n = ||\mathbf{r}_{z^+} - \mathbf{c}||_2^2 + \sum_{w=1}^{W} ||\mathbf{r}_{z^+}^w - \mathbf{c}_l||_2^2, \tag{14}$$

Besides, the triplet loss $\mathcal{L}_t$ in Eq. 11 is adapted to $\mathcal{L}'_t$ formulated as:

$$\mathcal{L}'_t = \max\left\{||\mathbf{c}-\mathbf{r}_{z^+}||_2^2 - ||\mathbf{c}-\mathbf{r}_{z^-}||_2^2 + \lambda_t, 0\right\} + \sum_{w=1}^{W} \max\left\{||\mathbf{c}_l-\mathbf{r}_{z^+}^w||_2^2 - ||\mathbf{c}_l-\mathbf{r}_{z^-}^w||_2^2 + \lambda_t, 0\right\}, \tag{15}$$

where $\mathbf{r}_{z^-}^w = f_{local}(Z_w^-)$ indicates the representation of abnormal entries in the w-th subsequence. Then, the overall objective function to achieve the counterfactual explanation for OC4Seq can be revised as $\mathcal{L}_{OC4Seq} = \mathcal{L}'_n + \alpha\mathcal{L}'_t + \beta\mathcal{L}_c + \gamma\mathcal{L}_s$,

where $\mathcal{L}_c$ and $\mathcal{L}_s$ are defined in Eqs. 12 and 13. We adopt a similar training strategy described previously to train the entry detection $g(\cdot)$ for explanation.

5 Experiments

5.1 Experimental Setting

Datasets. We evaluate our model on the following three datasets, which all provide entry-level labels:

- **BlueGene/L (BGL)** [17] contains alert and not-alert log messages collected from a BlueGeme/L supercomputer system.
- **Thunderbird** [17] is another log dataset collected from a Thunderbird supercomputer system.
- **CERT Insider Threat Dataset (CERT)** [10] is a synthetic dataset consisting of log files that record the computer-based activities for all employees in an institution. The CERT dataset contains 3995 benign employees and 5 insiders. On average, the number of activities for each employee is around 40000. We use version 4.2 of the CERT dataset.

Table 1. Statistics of Three Datasets.

Dataset	Normal Dataset $\mathcal{P}$ (seq)	Unlabeled Dataset $\mathcal{U}$	
		Normal (seq)	Anomalous (seq/entry)
BGL	344576	77548	36470/627373
Thunderbird	280064	63126	134365/408202
CERT	1391104	572560	52033/121751

For BGL and Thunderbird, we apply the log parser, Drain [13], to transfer the raw unstructured log messages to log templates and represent the log sequences as the template sequences. For CERT, we use user activities to compose the sequences. For all three datasets, we adopt a sliding window with size 20 to split the log files into sequences and set the step size as 10.

Table 1 shows the statistics of the normal dataset $\mathcal{P}$ and unlabeled datasets $\mathcal{U}$, where the last column indicates the numbers of anomalous sequences as well as the anomalous entries in the unlabeled datasets $\mathcal{U}$. For BGL, Thunderbird, and CERT, the ratios of anomalous entries in anomalous sequences are 0.86, 0.15, and 0.12, respectively. It is worth noting that the ground truth about the anomalous sequences and entries in the unlabeled dataset $\mathcal{U}$ is not available during the training phase. We also build a small validation set for each dataset to tune the hyper-parameters in CFDet as well as baselines for the anomalous sequence and entry detection, where the normal/anomalous sequences on BGL, Thunderbird, and CERT are 8617/4053, 7015/14930, 63618/5782, respectively.

We deploy the models with the best performance on the validation set to detect anomalous sequences on $\mathcal{U}$.

Baselines. We compare our CFDet with the following explanation approaches.

- **Attention.** The attention mechanism is often used to provide explanations based on the attention weights [24]. We revise the LSTM model $f(\cdot)$ by adding the attention mechanism and train the model based on the Deep SVDD and OC4Seq losses. The attention weights show the contributions of entries to the predictions. A threshold is set to identify anomalous entries.
- **Shapley.** The Shapley value is a classical approach that attributes the prediction of a machine-learning model on input to its base features [15,21]. We derive the Shapley values of entries in sequences as explanations of the anomalous sequence detector. Especially, given an anomalous sequence, an entry with a high Shapley value indicates that replacing the entry with an entry in a normal sequence could significantly reduce the distance from the sequence to the normal center $\mathbf{c}$.
- **DeepAID** [11]. DeepAID can interpret the sequence anomaly detection models by identifying the abnormal entries in the sequence. In particular, DeepAID adopts DeepLog [8] for sequence anomaly detection. To achieve explanations for DeepLog, DeepAID proposes to identify the abnormal entries by a search-based saliency test on each entry in an anomalous sequence.

Implementation Details. We represent the log templates in BGL and user activities in CERT using embedding vectors of size 50, and in Thunderbird using vectors of size 500. For both BGL and CERT, we use a single-layer LSTM with a hidden size of 128 as the anomalous sequence detector $f(\cdot)$. For Thunderbird, we use a single-layer LSTM with a hidden size of 512. We train $f(\cdot)$ in 50 epochs and update the center $\mathbf{c}$ in the first 20 epochs. For the LSTM model used in the anomalous entry detector $g(\cdot)$, for all three datasets, we use a single-layer LSTM with a hidden size of 128, which is trained in 100 epochs. The **code and datasets** are available online[1].

For the attention-based baseline, we set the threshold as 0.05 because the sequence length is 20, which means if one entry makes a contribution higher than an average ratio, we will label it as anomalous. For the Shapley value-based baseline, once a sequence is detected as an anomaly, we consider the entries with positive Shapley values as anomalous entries. To ensure a fair comparison, all hyperparameters in baselines are also tuned based on the validation set.

Evaluation Metrics. As the explanation to the anomalous sequence detection is to highlight the abnormal entries in the sequences, we adopt the precision (Prec.), recall (Rec.), F-1 score, and Area Under the Receiver Operating Characteristic Curve (AUC) to evaluate the performance of anomalous sequence and entry detection and report the mean and standard deviation after 5 times of running. Precision, recall, and F-1 score indicate the performance focusing on the anomaly class, while AUC indicates the true positives against false positives across various anomalous score thresholds.

[1] https://github.com/Serendipity618/CFDet.

5.2 Experimental Results

Sequence Anomaly Detection. As shown in Table 2, we first demonstrate that both Deep SVDD and OC4Seq can achieve good performance on anomalous sequence detection. Therefore, there is an urgent need to develop explanation approaches for this type of sequence anomaly detection approaches.

Table 2. Anomalous sequence detection (mean ± std.).

Dataset		BGL	Thunderbird	CERT
Deep SVDD	Pre.	$98.81_{\pm 0.39}$	$94.59_{\pm 3.62}$	$100.00_{\pm 0.00}$
	Rec.	$98.02_{\pm 1.14}$	$100.00_{\pm 0.00}$	$63.77_{\pm 0.00}$
	F-1	$98.41_{\pm 0.56}$	$97.19_{\pm 1.96}$	$77.88_{\pm 0.00}$
	AUC	$98.73_{\pm 0.54}$	$93.75_{\pm 4.55}$	$81.88_{\pm 0.00}$
OC4Seq	Pre.	$94.74_{\pm 2.25}$	$95.31_{\pm 0.49}$	$99.47_{\pm 0.50}$
	Rec.	$99.37_{\pm 0.35}$	$96.17_{\pm 3.04}$	$63.29_{\pm 1.86}$
	F-1	$96.98_{\pm 1.02}$	$95.71_{\pm 1.33}$	$77.34_{\pm 1.25}$
	AUC	$98.38_{\pm 0.43}$	$97.77_{\pm 1.48}$	$81.63_{\pm 0.92}$

The Performance of Counterfactual Explanations. We verify the effectiveness of counterfactual explanations by examining the identified anomalous entries. Attention, Shapley, and CFDet are to explain the results from Deep SVDD and OC4Seq, while DeepAID is to explain the results from DeepLog.

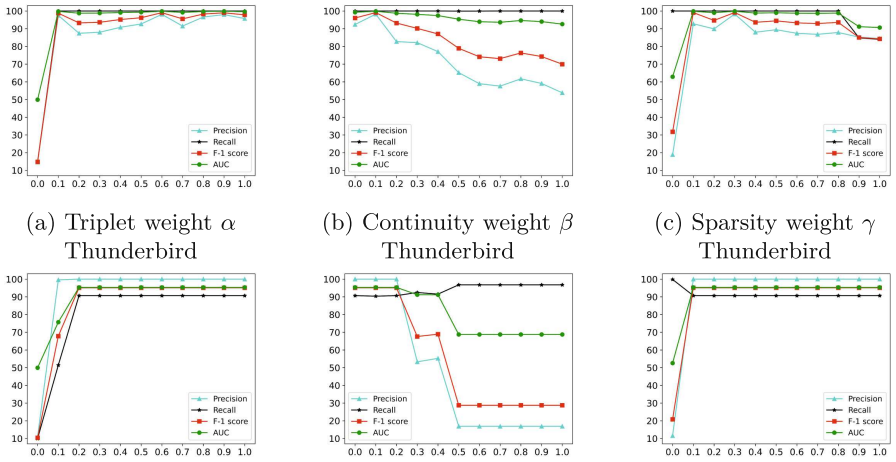
Table 3 shows the performance of detecting anomalous entries on the detected anomalous dataset $\tilde{\mathcal{U}}^{-}$. First, Shapley can achieve good performance on BGL. This could be because the ratio of anomalous entries in anomalous sequences is high, thus making the anomalous entries easy to detect. On the other hand, both Thunderbird and CERT have smaller numbers of anomalous entries, which makes the detection more challenging. Similarly, DeepAID can achieve good performance on BGL but its performance on Thunderbird and CERT is poor due to low recall. CFDet achieves near-perfect F-1 scores and AUC on all three datasets. It means CFDet can provide the post-hoc explanation for anomalous sequence detection with high fidelity. In particular, CFDet detects all the anomalous entries on Thunderbird as well as a perfect precision on CERT. Meanwhile, the standard deviations of CFDet are smaller compared with other baselines, which shows that CFDet is stable in highlighting the rare events in the sequences.

Table 3. Counterfactual Explanations in terms of Anomalous Entry Detection (mean ± std.).

Method		DeepAID	Attention		Shapley		CFDet	
Base AD Model		DeepLog	Deep SVDD	OC4Seq	Deep SVDD	OC4Seq	Deep SVDD	OC4Seq
BGL	Pre.	**99.48**$_{\pm 0.41}$	70.19$_{\pm 18.57}$	82.79$_{\pm 2.71}$	98.05$_{\pm 0.98}$	94.00$_{\pm 0.40}$	98.91$_{\pm 0.91}$	96.77$_{\pm 2.06}$
	Rec.	89.53$_{\pm 0.29}$	77.17$_{\pm 17.62}$	53.23$_{\pm 14.29}$	98.37$_{\pm 1.84}$	94.97$_{\pm 2.11}$	**99.38**$_{\pm 1.51}$	99.08$_{\pm 0.35}$
	F-1	94.25$_{\pm 0.34}$	72.80$_{\pm 16.39}$	63.92$_{\pm 12.27}$	98.19$_{\pm 0.77}$	94.47$_{\pm 0.93}$	**99.13**$_{\pm 0.83}$	97.90$_{\pm 1.02}$
	AUC	90.81$_{\pm 2.92}$	55.99$_{\pm 17.45}$	54.09$_{\pm 6.25}$	93.31$_{\pm 2.51}$	80.69$_{\pm 2.29}$	**96.33**$_{\pm 2.81}$	95.75$_{\pm 2.44}$
Thunderbird	Pre.	80.06$_{\pm 16.71}$	33.40$_{\pm 28.66}$	16.03$_{\pm 0.44}$	67.22$_{\pm 14.59}$	81.09$_{\pm 11.91}$	**90.99**$_{\pm 10.33}$	87.81$_{\pm 6.60}$
	Rec.	48.03$_{\pm 9.64}$	51.79$_{\pm 17.08}$	56.39$_{\pm 1.70}$	94.17$_{\pm 5.33}$	96.89$_{\pm 1.36}$	**100.00**$_{\pm 0.00}$	99.56$_{\pm 0.93}$
	F-1	60.02$_{\pm 12.19}$	34.61$_{\pm 15.20}$	24.95$_{\pm 0.68}$	77.96$_{\pm 11.45}$	87.97$_{\pm 7.45}$	**94.97**$_{\pm 6.41}$	93.19$_{\pm 3.63}$
	AUC	72.09$_{\pm 6.43}$	58.87$_{\pm 11.48}$	53.98$_{\pm 2.23}$	92.82$_{\pm 4.88}$	96.75$_{\pm 1.72}$	99.04$_{\pm 1.38}$	**99.05**$_{\pm 0.31}$
CERT	Pre.	66.22$_{\pm 2.70}$	71.35$_{\pm 20.02}$	24.51$_{\pm 12.06}$	60.72$_{\pm 9.05}$	22.97$_{\pm 1.28}$	**99.31**$_{\pm 2.19}$	98.67$_{\pm 2.82}$
	Rec.	38.83$_{\pm 0.52}$	58.94$_{\pm 24.37}$	74.01$_{\pm 18.67}$	**91.60**$_{\pm 0.33}$	85.07$_{\pm 17.27}$	90.73$_{\pm 0.16}$	89.16$_{\pm 4.03}$
	F-1	48.94$_{\pm 0.98}$	63.26$_{\pm 23.02}$	36.22$_{\pm 16.49}$	72.67$_{\pm 6.39}$	35.89$_{\pm 2.53}$	**94.81**$_{\pm 0.94}$	93.61$_{\pm 2.42}$
	AUC	66.95$_{\pm 0.44}$	78.20$_{\pm 12.25}$	74.07$_{\pm 15.55}$	91.92$_{\pm 1.26}$	77.98$_{\pm 2.73}$	**95.32**$_{\pm 0.05}$	94.51$_{\pm 1.99}$

Sensitivity Analysis. We adopt Deep SVDD as the base AD model and use Thunderbird and CERT datasets to analyze how triplet loss (Eq. 11), continuity loss (Eq. 12), and sparsity loss (Eq. 13) affect the counterfactual explanation in terms of anomalous entry detection.

Sensitivity Analysis on the Weight of Each Loss Term in Equation 9. We first analyze the effects on performance by tuning the weights of triplet loss



(a) Triplet weight α Thunderbird (b) Continuity weight β Thunderbird (c) Sparsity weight γ Thunderbird

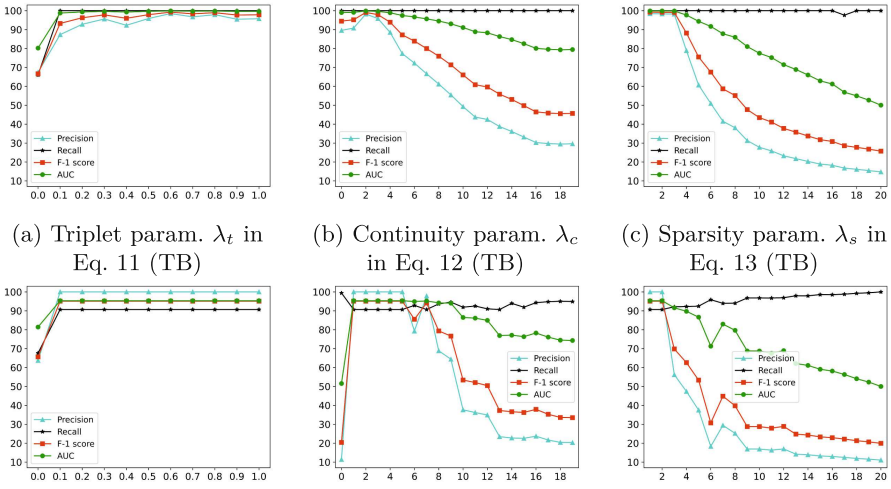
(d) Triplet weight α CERT (e) Continuity weight β CERT (f) Sparsity weight γ CERT

Fig. 2. Hyperparameter sensitivity on the weight of each loss term in Eq. 9.

α, the continuity loss β, and sparsity loss γ defined in Eq. 9. First, Fig. 2 shows on both datasets, triplet loss, continuity loss, and sparsity loss are critical for anomalous entry detection. Especially, for the triplet weight, when $\alpha = 0$, the AUC value is just around 50%, and the F-1 score is around 10% (shown in Figs. 2a and 2d), meaning the critical of triplet loss for accurately detecting anomalous entries. We can have a similar observation for the sparsity loss shown in Figs. 2c and 2f, i.e., no sparsity loss ($\gamma = 0$) leading to low F-1 and AUC. Meanwhile, we can also notice that both the triplet weight α and sparsity weight γ do not have significant impacts on the performance as long as the values are not zero. On the other hand, the performance is sensitive to the continuity weight β. As shown in Figs. 2b and 2e, when the β value keeps increasing, the performance becomes worse.

Sensitivity Analysis on the Constant Terms in Loss Terms Defined in Equations 11–13. We then analyze the sensitivity of the triplet parameter λ_t (Eq. 11), continuity parameter λ_c (Eq. 12) and sparsity parameter λ_s (Eq. 13). λ_t indicates the margin between the distance from $\mathbf{r}_{z^+}$ to $\mathbf{c}$ and distance from $\mathbf{r}_{z^-}$ to $\mathbf{c}$. λ_c indicates the expected number of anomalous subsequences. For example, in our experiments, the sequence length is 20. $\lambda_c = 0$ means all entries in the anomalous sequence are anomalous, while $\lambda_c = 19$ means no consecutive anomalous entries in the sequence. λ_s indicates the expected number of anomalous entries. $\lambda_s = 1$ indicates we expect only one anomalous entry in the sequence.

Figures 3a and 3d show that when the margin is 0 ($\lambda_t = 0$), CFDet cannot get a reasonable result, which meets the expectation that we need to ensure the



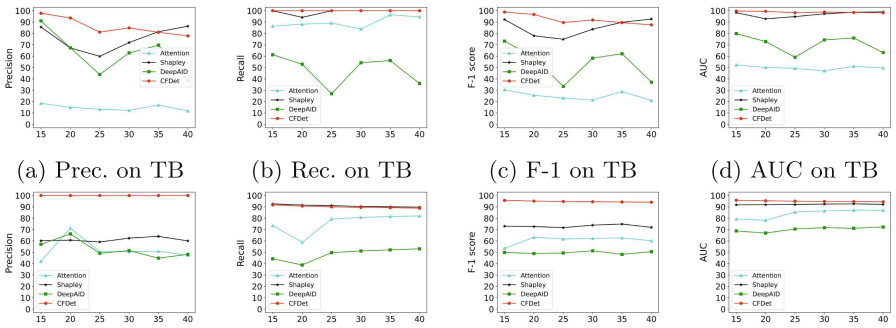
(a) Triplet param. λ_t in Eq. 11 (TB)
(b) Continuity param. λ_c in Eq. 12 (TB)
(c) Sparsity param. λ_s in Eq. 13 (TB)
(d) Triplet param. λ_t in Eq. 11 (CERT)
(e) Continuity param. λ_c in Eq. 12 (CERT)
(f) Sparsity param. λ_s in Eq. 13 (CERT)

Fig. 3. Hyperparameter sensitivity on the constant terms in losses defined in Eqs. 11–13

detected anomalous subsequence have a large distance to the center. Once the margin is greater than 0 ($\lambda_t > 0$), CFDet achieves much better performance. Figures 3b and 3c show the performance of anomalous entry detection on Thunderbird with λ_c ranging from 0 to 19 and λ_s from 1 to 19, respectively. We observe that when λ_c and λ_s are small, our approach achieves very good performance. On the other hand, with the increase of λ_c and λ_s, the performance becomes worse. This is because, for Thunderbird, the ratio of anomalous entries is small (0.15). Therefore, once we set λ_c and λ_s in a reasonable range, our approach can achieve good performance. On the other hand, if we keep increasing the values λ_c and λ_s, the performance will get worse. This is because a large λ_c or λ_s would force the model to select more entries as anomalous, leading to high false alerts. Similar to the results on Thunderbird, from Figs. 3e and 3f, we can observe that when λ_c and λ_s are small, our approach achieves good performance. On the other hand, with the increase of λ_c and λ_s, the performance becomes worse due to the low ratio of anomalous entries.

Anomalous Entry Detection with Various Sequence Lengths. In our experiments, the default sequence length is 20. We further analyze the performance of CFDet against baselines on anomalous entry detection with various sequence lengths (from 15 to 40 entries) on the Thunderbird and CERT datasets.

Figure 4 shows the experimental results of counterfactual explanations with various sequence lengths on Thunderbird and CERT, respectively. We can observe that CFDet outperforms Attention and DeepAID on both datasets by a large margin in terms of F-1 score and AUC for different sequence lengths. Shapley also achieves good performance in terms of F-1 score on Thunderbird when the sequence length reaches 40 and a very close AUC score to CFDet on both Thunderbird and CERT, but CFDet still outperforms Shapley in most cases. Meanwhile, we can observe that for CFDet, the recall and AUC values are high when the sequence lengths increase from 15 to 40, but the precision and

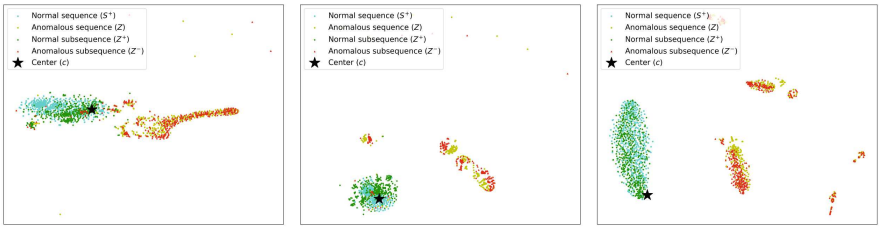


(a) Prec. on TB (b) Rec. on TB (c) F-1 on TB (d) AUC on TB

(e) Prec. on CERT (f) Rec. on CERT (g) F-1 on CERT (h) AUC on CERT

Fig. 4. Performance of counterfactual explanations in terms of anomalous entry detection on Thunderbird (Figs. 4a–4d) and CERT (Figs. 4e–4h)

F-1 scores keep reducing when the length is greater than 15 on Thunderbird. It means on the Thunderbird dataset, CFDet is able to identify all the abnormal entries but predicts some normal entries as abnormal (false positive) when the sequence length is large. On the other hand, the performance of CFDet on the CERT dataset is relatively stable with various sequence lengths.



(a) BGL (b) Thunderbird (c) CERT

Fig. 5. Visualization of the center (black), detected normal sequences S^+ (cyan), detected anomalous sequences Z (yellow) and the corresponding normal Z^+ (green) and anomalous Z^- (red) subsequences (Color figure online)

Visualization. We further visualize representations of normal (S^+) and anomalous (Z) sequences on BGL, Thunderbird, and CERT datasets by adopting Deep SVDD as the base AD model. We randomly select 500 detected normal and anomalous sequences, respectively. For the detected anomalous sequence Z from each dataset, we then get the anomalous subsequence Z^- and the normal subsequence Z^+. After deriving the representations of S^+, Z, Z^-, and Z^+ based on $f(\cdot)$, we adopt the t-SNE to map the representations into a two-dimensional space. Figure 5 shows the visualization plots. Overall, as expected, S^+ and Z^+ as normal sequences and subsequences are grouped together and located around the normal center $\mathbf{c}$, while Z and Z^- as anomalous sequences and subsequences are close but far from the normal center. It also demonstrates the detected Z^+ only contains the normal entries, while the corresponding Z^- has the anomalous entries. Meanwhile, on both Thunderbird and CERT, the anomalous sequences and the corresponding subsequences are much more diverse compared with normal sequences.

6 Conclusion

In this paper, we have developed CFDet that can provide counterfactual explanations to anomalous sequences identified by Deep SVDD-based sequence anomaly detection models. As in the anomaly detection scenario, the counterfactual example and the subsequence with anomalous entries are complementary to each other, by detecting anomalous entries, we can generate the counterfactual of an

anomalous sequence. The core idea to identify anomalous entries for explanation is that the anomalous entries in the anomalous sequence should be far from the center of normal samples while the complementary normal subsequence, i.e., the counterfactual sequence of the original anomalous sequence, should be close to the center. Experiments on three datasets show that CFDet can provide explanations by identifying anomalous entries with high accuracy.

Acknowledgment. This work was supported in part by NSF 1910284, 2103829, and 2348391.

References

1. Alahmadi, B.A., Axon, L., Martinovic, I.: 99% False Positives: A Qualitative Study of {SOC} Analysts' Perspectives on Security Alarms, pp. 2783–2800 (2022)
2. Antwarg, L., Miller, R.M., Shapira, B., Rokach, L.: Explaining anomalies detected by autoencoders using SHAP (2020). arXiv:1903.02407 [cs, stat]
3. Bhusal, D., et al.: Sok: modeling explainability in security analytics for interpretability, trustworthiness, and usability. In: Proceedings of the 18th International Conference on Availability, Reliability and Security, pp. 1–12 (2023)
4. Brown, A., Tuor, A., Hutchinson, B., Nichols, N.: Recurrent neural network attention mechanisms for interpretable system log anomaly detection. In: Proceedings of the First Workshop on MLCS. MLCS'18 (2018)
5. Chen, C., Li, O., Tao, C., Barnett, A.J., Su, J., Rudin, C.: This looks like that: deep learning for interpretable image recognition. In: NeurIPS (2019)
6. Cheng, H., Xu, D., Yuan, S.: Sequential anomaly detection with local and global explanations. In: 2022 IEEE International Conference on Big Data (Big Data), pp. 1212–1217 (2022). https://doi.org/10.1109/BigData55660.2022.10020990
7. Cheng, H., Xu, D., Yuan, S.: Explainable sequential anomaly detection via prototypes. In: 2023 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2023). https://doi.org/10.1109/IJCNN54540.2023.10191703
8. Du, M., Li, F., Zheng, G., Srikumar, V.: DeepLog: anomaly detection and diagnosis from system logs through deep learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (2017)
9. Farzad, A., Gulliver, T.A.: Unsupervised log message anomaly detection. ICT Express **6**(3), 229–237 (2020)
10. Glasser, J., Lindauer, B.: Bridging the gap: a pragmatic approach to generating insider threat data. In: 2013 IEEE Security and Privacy Workshops (2013)
11. Han, D., et al.: DeepAID: interpreting and improving deep learning-based anomaly detection in security applications. In: CCS (2021)
12. Han, X., Cheng, H., Xu, D., Yuan, S.: Interpretablesad: interpretable anomaly detection in sequential log data. In: 2021 IEEE International Conference on Big Data (Big Data), pp. 1183–1192 (2021). https://doi.org/10.1109/BigData52589.2021.9671642
13. He, P., Zhu, J., Zheng, Z., Lyu, M.R.: Drain: an online log parsing approach with fixed depth tree. In: 2017 IEEE International Conference on Web Services (ICWS), pp. 33–40 (2017). https://doi.org/10.1109/ICWS.2017.13
14. Liznerski, P., Ruff, L., Vandermeulen, R.A., Franks, B.J., Kloft, M., Müller, K.R.: Explainable deep one-class classification. In: ICLR (2021)
15. Molnar, C.: Interpretable Machine Learning (2019)

16. Nguyen, Q.P., Lim, K.W., Divakaran, D.M., Low, K.H., Chan, M.C.: GEE: a gradient-based explainable variational autoencoder for network anomaly detection. In: 2019 IEEE Conference on CNS, pp. 91–99 (2019)
17. Oliner, A., Stearley, J.: What supercomputers say: a study of five system logs. In: 37th Annual IEEE/IFIP International Conference on DSN. IEEE (2007)
18. Ruff, L., et al.: Deep one-class classification. In: ICML (2018)
19. Ruff, L., et al.: Deep semi-supervised anomaly detection. In: ICLR (2020)
20. Yuan, S., Zheng, P., Wu, X., Tong, H.: Few-shot insider threat detection. In: CIKM (2020)
21. Sundararajan, M., Najmi, A.: The many Shapley values for model explanation. In: ICML, pp. 9269–9278. PMLR (2020)
22. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (2018)
23. Wang, Z., Chen, Z., Ni, J., Liu, H., Chen, H., Tang, J.: Multi-scale one-class recurrent neural networks for discrete event sequence anomaly detection. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 3726–3734 (2021)
24. Xu, K., et al.: Show, attend and tell: neural image caption generation with visual attention. In: ICML, pp. 2048–2057. PMLR (2015)
25. Yu, M., Chang, S., Zhang, Y., Jaakkola, T.S.: Rethinking cooperative rationalization: introspective extraction and complement control. In: EMNLP (2019)
26. Zhou, R., et al.: LogAnomaly: unsupervised detection of sequential and quantitative anomalies in unstructured logs. In: IJCAI, pp. 4739–4745 (2019)