InterpretableSAD: Interpretable Anomaly Detection in Sequential Log Data

Xiao Han
Utah State University
Logan, UT, USA
xiao.han@usu.edu

He Cheng
Utah State University
Logan, UT, USA
he.cheng@usu.edu

Depeng Xu
University of Arkansas
Fayetteville, AR, USA
depengxu@uark.edu

Shuhan Yuan
Utah State University
Logan, UT, USA
shuhan.yuan@usu.edu

2021 IEEE International Conference on Big Data (Big Data) | 978-1-6654-3902-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/BigData52589.2021.9671642

***Abstract*—Anomaly detection in sequential log data is a common data analysis task as it contributes to detecting critical information, such as malfunctions of systems. However, due to the scarcity of anomalies, the traditional supervised learning approaches cannot be applied for anomaly detection tasks. Meanwhile, most of the existing studies only focus on identifying the anomalous log sequences and cannot further detect the anomalous events in a sequence. In this work, we present InterpretableSAD, an interpretable log anomaly detection framework that can achieve both anomalous sequence and fine-grained event detection. Given a set of normal log sequences, we propose a data augmentation strategy to generate a set of anomalous sequences via negative sampling so that we can train a binary classification model based on the observed normal sequences and the generated anomalous sequences. After training, the classification m odel is able to detect real anomalous log sequences. We then consider the anomalous event detection as a model interpretation problem and apply an interpretable machine learning technique in a novel way to detect which parts of the sequences, a.k.a, anomalous events, lead to anomalous issues. Experimental results on three log datasets show the effectiveness of our proposed framework.**

***Index Terms*—anomaly detection, negative sampling, integrated gradients, interpretability**

I. Introduction

Anomaly detection in sequential log data, which aims to identify sequences that deviate from the expected behavior or patterns, has received much attention due to its broad application [1]–[6]. For example, online services generate large amounts of log messages that record states of systems, where the log messages can be modeled as an event sequence [7]. Because online services are everywhere in our daily life, and a little jitter of the services could cause severe consequences, such as financial losses, it is crucial to detect anomalous states in a timely manner to ensure the reliability of the online services and mitigate the losses.

Traditional approaches for sequential anomaly detection are strictly rule-based and dependent on domain knowledge about the patterns of sequences [8]. Although the rule-based approaches can achieve good performance for specific anomalies, the limitations are still obvious, i.e., it is hard to extend to detect new types of anomalies. Currently, many machine learning-based approaches are proposed. Considering the lack of anomalous samples, many unsupervised learning models, such as Principal Component Analysis (PCA) [9], or one class classification models, such as one-class SVM [10], are used to detect anomalies. In order to further model the temporal information of sequential data, the state-of-the-art anomaly detection approaches are mainly based on deep learning models. For example, DeepLog [1] and LogAnomaly [2] utilize long-short term memory (LSTM) network to capture normal sequential patterns from normal samples and detect anomalies.

Because the anomalous samples are rare due to the nature of anomalies, most of the existing approaches are trained in an unsupervised learning manner. These approaches usually assume the normal samples are concentrated in a hypersphere, while the anomalous samples are outside the hypersphere [11], [12]. However, due to the dynamics of sequential data, such assumption can be easily violated for sequential anomaly detection. In this paper, different from the existing work, we propose a data augmentation strategy to generate the anomalous samples by negative sampling. Based on negative sampling, we can generate sufficient anomalous samples to train a binary classification model without making an assumption about normal data distribution. When the generated samples are large enough to cover the common anomalous scenarios, the classifier trained on generated anomalous samples can detect the real anomalies as well.

Furthermore, the existing log anomaly detection approaches can only detect anomalous log sequences and cannot identify anomalous events in the sequence. Specifically, if a log sequence is detected as anomalous, there must be one or more events in the sequence that deviate from the expected patterns. For example, if a system is under attack, the operations conducted by the attacker are anomalous events in a log sequence. As an online service system can generate hundreds of logs per second, the capacity of distinguishing anomalous events from normal ones in a sequence is also critical for system administrators to locate the anomalous operations besides detecting the anomalous sequences. However, detecting the anomalous events in a sequence faces several challenges. First, because the information of a single event is very limited, it is hard to identify useful features to represent an event. Second, an anomalous event could be caused by the broken of correlation among the events. It means we still need the information of the whole sequence to identify the anomalous events. To tackle these challenges, we creatively apply an

Authorized licensed use limited to: USU - Space Dynamics Laboratory. Downloaded on January 25,2022 at 04:00:06 UTC from IEEE Xplore. Restrictions apply.

interpretable machine learning technique, Integrated Gradients (IG) [13], for anomalous event detection. The motivation is that if a classifier predicts a sequence as anomalous, the model interpretation technique should be able to identify which part of the input sequence leads to the anomalous outcome. Then, the events that are responsible for the anomalous result are anomalous events.

In this work, we propose a framework, called InterpretableSAD, for detecting anomalous log sequences as well as anomalous events. Specifically, we propose a negative sampling algorithm to generate potential anomalous sequences automatically so that we can train a binary classification model through the observed normal samples and the generated anomalous samples. We further propose to apply Integrated Gradients for anomalous event detection. The events in a sequence that significantly contribute to the anomalous result are marked as anomalous events.

The main contributions of this paper are as follows. First, we propose a novel negative sampling strategy to generate potential anomalous samples based on the observed normal samples, and then we can train a binary classifier for anomaly detection. While data augmentation techniques are widely used in computer vision and natural language processing to improve model performance, it is under-exploited in the area of anomaly detection. Second, most existing anomaly detection models only achieve the anomalous sequence detection and cannot identify the anomalous events in the sequence. We novelly apply a model interpretation approach, Integrated Gradients (IG), to achieve anomalous event detection. Third, because IG relies on an appropriate baseline input for feature attributions, we further propose a novel baseline generation algorithm to improve the performance of anomalous event detection. Experimental results show that InterpretableSAD can achieve state-of-the-art performance on anomalous log sequence detection and further identify the anomalous events in the anomalous sequences with high accuracy.

II. Background

Anomaly Detection in Sequential Log Data. Many sequential anomaly detection approaches have been proposed in recent years. Several traditional anomaly detection approaches are based on supervised learning, such as logistic regression, decision tree [14], and Support Vector Machines (SVM) [15]. However, the major limitation of the supervised approaches is that they require an enormous number of labeled data for training, which is usually unavailable in anomaly detection scenarios. Hence, the unsupervised learning approaches have received more attention in the anomaly detection field, such as the dimensionality reduction-based approaches and clustering-based approaches [9], [16]. However, these approaches cannot capture the order information of sequence data.

In recent years, deep learning based sequential anomaly detection models are proposed to detect anomalies by checking differences between normal and anomalous patterns [1]–[3], [17], [18]. Specifically, when anomalous events occur, the pattern of sequences will be changed as well, and the models can detect the changes and then flag anomalies. Most of the existing approaches adopt recurrent neural networks to detect anomalous sequences in an unsupervised manner by modeling the patterns of normal sequences [1], [2]. These models adopt long-short term memory (LSTM) [19] to predict the next possible events based on previous events in a sequence. An anomalous sequence will be detected if the actual event is out of a candidate set of expected normal events. However, the existing cutting edge unsupervised approaches are only effective on sequence level anomaly detection and unable to provide detailed information of anomalies on the sub-sequence or event level.

Data Augmentation. Modern machine learning models usually require a large amount of labeled data for training. Unfortunately it is sometimes hard to achieve in reality due to cost and time factors. Data augmentation technique is to tackle the scarcity of labeled data issue by artificially expanding the labeled dataset. Currently, data augmentation is extensively used in image classification and natural language processing [20]–[22] to generate auxiliary training data. In practice, data augmentation is conducted by a set of transformation functions, such as rotation and flip for image data or synonym replacement for text data, on the existing dataset. The data generated by carefully designed transformation functions is beneficial to improve the performance of machine learning models.

Negative sampling as a special data augmentation technique is used to generate negative samples instead of positive ones when the negative samples are not available. Negative sampling is a key step in various applications, such as training word embeddings, knowledge graph embeddings, as well as recommender systems [23]–[25]. Since the main challenge of anomaly detection is the scarcity of anomalous samples, we propose the negative sampling strategy in our work to generate the potential abnormal sequences from the normal ones. Then, we can build a binary classification model to detect anomalies.

Interpretable Machine Learning. Although modern deep learning models have achieved great success in many applications, non-transparency is still a big issue for deploying highly complex models in production environments. As a consequence, interpretable machine learning, which aims at providing human understandable explanations about the decisions made by the models, has become an active research area [26]–[32]. Interpretable machine learning techniques can generally be categorized into two groups: intrinsic interpretability and post-hoc interpretability. Intrinsic interpretability indicates the models are interpretable due to simple structures, such as decision tree or linear regression, while the post-hoc interpretability implies creating a second model to provide explanations for an existing model [26], [27].

The interpretable anomaly detection models are very limited in the literature [32], [33]. Research in [33] focuses on detecting the outliers in attributed networks, while research in [32] also targets the interpretable anomaly detection in sequential data that leverages the attention mechanism to provide an attention score of each event in a sequence. However, the

Authorized licensed use limited to: USU - Space Dynamics Laboratory. Downloaded on January 25,2022 at 04:00:06 UTC from IEEE Xplore. Restrictions apply.

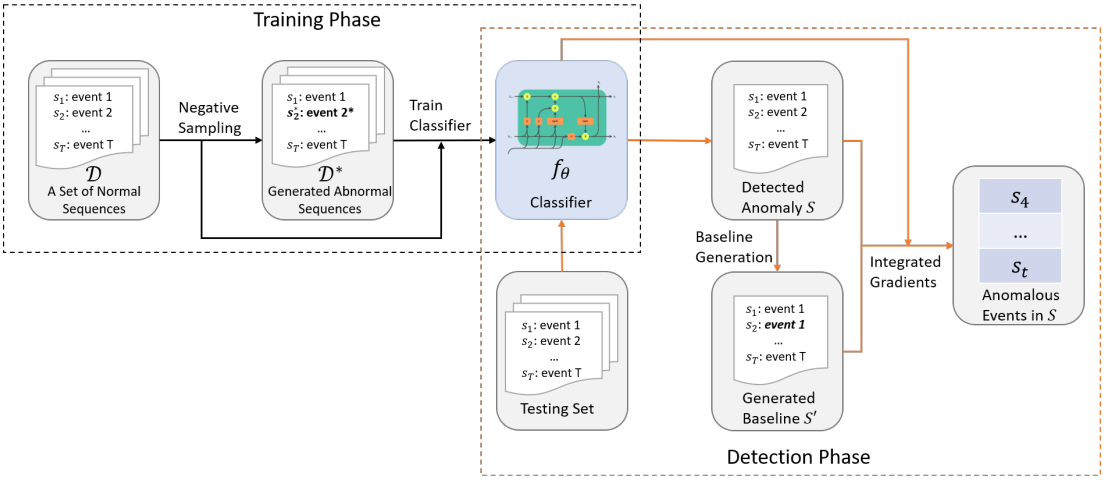


Fig. 1: Framework of InterpretableSAD

proposed model is trained to capture the normal patterns based on LSTM by predicting the next event given the previous events in a sequence. The attention scores, which are used for interpretation, are more about the correlation among events in a sequence instead of the correlation between events and the label (normal or anomalous). In our work, after training a classification model based on sequences generated by negative sampling, our framework can model the correlation between events and prediction outcome directly based on the interpretable machine learning technique.

III. Framework

Consider a log sequence of discrete events $S = \{s_1, ..., s_t, ..., s_T\}$, where $s_t \in \mathcal{E}$ indicates the event at the t-th position, and $\mathcal{E}$ is a set of unique events. In this work, we aim at predicting whether a log sequence S is anomalous based on a training dataset $\mathcal{D} = \{S^i\}_{i=1}^N$ that consists of only normal sequences. Meanwhile, for a sequence predicted as anomalous, we further target to identify anomalous events in the sequence so that the domain users can get insights about the detection model as well as anomalous sequences. To achieve the above two goals, in the training phase, assuming that we only have normal samples in our training dataset, we propose a negative sampling approach to generate the potential anomalous sequences so that we can train a classification model based on both positive and negative samples. After training, in the detection phase, once a sequence is predicted as anomalous by the classifier, we novelly leverage an interpretable machine learning technique to identify the anomalous events in the sequence. Specifically, we adopt the Integrated Gradients approach, which can explain the relationship between the prediction results and input features, to identify the anomalous events. Figure 1 shows our sequential data anomaly detection framework.

A. Data Augmentation via Negative Sampling

In the anomaly detection field, in most cases, we have plenty of normal samples but only observe a small number of anomalous samples. Hence, most anomaly detection approaches are based on one-class classification models. Different from the existing studies, in this work, we propose a data augmentation approach via negative sampling to generate the potential anomalous log sequences based on the observed normal sequences. Then, we can train a binary classification model based on two classes of samples.

In order to train an accurate binary classifier, we aim to generate a dataset $\mathcal{D}^*$ with sufficient anomalous samples that can cover common anomalous scenarios. We consider two anomalous scenarios for anomalous log sequence generation. First, there are some rare events in the sequences. For example, if an online system is compromised by an attacker, the attacker could conduct some uncommon events on the system. Second, some regular events happen in an unusual context. For example, an attacker aims to evade detection by performing some regular activities, but these activities happen at the wrong time, which means that these activities are suspicious based on their context. To simulate these two scenarios, we propose an algorithm (shown in Algorithm 1) to generate the potential anomalous log sequences.

Given the training set $\mathcal{D}$, in order to consider both the event and its context information, we generate a set of bigram events, where each bigram is a sub-sequence of two adjacent events, e.g., (s_t, s_{t+1}), in the sequence S. We then build a bigram event dictionary $\mathcal{B}$, where the key is the bigram, and the value is the corresponding frequency of the bigram in $\mathcal{D}$. Then, given a normal sequence $S \in \mathcal{D}$, we use it as a template to generate the potential anomalous sample by randomly replace r number of events in S. Specifically, for a randomly selected event s_t, we will replace the event s_{t+1} with another event s^*_{t+1} so that the bigram (s_t, s^*_{t+1}) is rare or never observed in the training set $\mathcal{D}$. Due to the scarcity of anomalous events, the bigram with a low frequency is suspicious. Since we replace r events with low-frequency, we expect that there is a high possibility that the generated sequences are anomalous.

Authorized licensed use limited to: USU - Space Dynamics Laboratory. Downloaded on January 25,2022 at 04:00:06 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: Negative Sampling

Input : Training set $\mathcal{D}$, Negative sample size M
Output: Negative sample set $\mathcal{D}^*$
Generate a bigram event dictionary $\mathcal{B}$ based on $\mathcal{D}$
for $i = 0$ **to** M **do**
  Randomly select S from $\mathcal{D}$
  $ind \leftarrow$ Randomly select r indices of events from S
  for t *in* ind **do**
    $(s_t, s^*_{t+1}) \leftarrow$ randomly select or generate a rare or never observed bigram in $\mathcal{B}$
    $(s_t, s_{t+1}) \leftarrow (s_t, s^*_{t+1})$
  $S^* \leftarrow S, \quad \mathcal{D}^* += S^*$
return $\mathcal{D}^*$

B. Training a Classification Model

After generating a set of anomalous sequences $\mathcal{D}^*$, we use both $\mathcal{D}$ and $\mathcal{D}^*$ to train a binary classification model $f : S \rightarrow [0, 1]$. In particular, we first adopt word2vec [23] to get representations of events in $\mathcal{E}$ by training on the dataset $\mathcal{D}$. After mapping the events in a sequence to the embedding space, we train a neural network f_θ to predict whether a sequence is normal or anomalous:

$$\hat{y} = f_\theta(S), \tag{1}$$

where $\hat{y}$ indicates the predicted label. Because we have two classes of samples on hand, we further adopt the cross-entropy loss to train the neural network. The objective function is defined as:

$$\mathcal{L} = \sum_{j \in \mathcal{D}^* \cup \mathcal{D}} -y_j \log \hat{y}_j - (1 - y_j) \log(1 - \hat{y}_j). \tag{2}$$

It is worth noting that any neural network, which can model the sequential data, is able to be used in our framework for anomalous sequence detection.

C. Anomalous Event Detection via Integrated Gradients

After the classification model is trained based on the normal and generated anomalous sequences, we can deploy the model for detecting the anomalous samples for real. However, in practice, only detecting the anomalous sequences is far from sufficient. For example, because online service systems can generate a huge amount of messages per minute, only identifying anomalous sequences is not sufficient to help the system administrator locate the anomalous operations from attackers. Hence, we further aim at detecting the anomalous events in sequences. There are two key challenges in detecting anomalous events. First, an independent event in a sequence does not contain enough information to support anomaly detection. Second, whether an event is anomalous also depends on its context. To tackle these challenges, in this work, instead of designing a traditional detection model built on the event information, we consider the anomalous event detection in a sequence as a model interpretation problem. The motivation is that when a classification model predicts a sequence as anomalous, the model should detect some anomalous patterns in the sequence. By leveraging the model interpretation techniques, we can further identify the anomalous patterns, i.e., anomalous events, in the sequence.

In our framework, we adopt Integrated Gradients (IG) [13] to derive feature attributions of each sequence. IG is a model interpretable technique that can interpret prediction results by attributing input features in a human-understandable way for various classification tasks, such as image or text classification. For example, in an image or text classification task, IG can show which pixels or words are responsible for a certain label. In this work, we leverage IG to identify anomalous events that cause the sequential anomalies.

Formally, given a neural network $f_\theta : S \rightarrow [0, 1]$, integrated gradients are attributions of the prediction at input S relative to a baseline input S' as a vector $A_{f_\theta}(S, S') = (a_1, ..., a_T)$, where a_t is the contribution of s_t to the prediction $f_\theta(S)$. A large positive a_t indicates that feature strongly increases the network output f_θ, while a_t close to zero indicates that the feature did not influence f_θ. Hence, we consider the importance scores $A_{f_\theta}(S, S')$ as anomalous scores to detect the anomalous events.

Specifically, in our scenario, let S be a sequence, and S' be a baseline sequence. The integrated gradient for the t-th event for sequence S and baseline S' is defined as follows.

$$IG_t(S) \equiv (s_t - s'_t) \times \int_{\alpha=0}^{1} \frac{\partial f_\theta(S' + \alpha \times (S - S'))}{\partial s_t} d\alpha. \tag{3}$$

The integrated gradients have a property called completeness axiom, which indicates that the sum of integrated gradients over the whole sequence is the difference between the output of classification model f_θ on the input sequence and its baseline, i.e.,

$$\sum_{t=1}^{T} A_{f_\theta}(S_t, S'_t) = f_\theta(S) - f_\theta(S'). \tag{4}$$

The completeness axiom of IG ensures that the anomalous score of each event is proportional to the contribution of making S as an anomaly [13].

As shown in Equation 3, IG gets the importance score for each event in S by integrating the gradient for each event from a baseline S' to the sequence S, where the baseline is supposed to represent "absence" of features [34]. Hence, finding a reasonable baseline is an essential step for applying the IG method. For image classification models, the black image is widely used as a baseline, while the zero-embedding matrix is a common baseline for the text classification task. However, it is not straightforward to find a single baseline for anomaly detection on sequential data. Different from the text classification task, say sentiment analysis, where the key words contributed to the sentiment are usually positive or negative words, the anomalous events in sequences are related to the context of the sequences, and no widely accepted criteria can be used to quantify the abnormality. Therefore, we propose to generate a unique baseline for each sequence. Meanwhile, based on the completeness axiom shown in Equation 4, the sum of importance scores over events in a sequence is the

Authorized licensed use limited to: USU - Space Dynamics Laboratory. Downloaded on January 25,2022 at 04:00:06 UTC from IEEE Xplore. Restrictions apply.

prediction difference between the original sequence and the baseline. In order to have a reasonable IG value, the generated baseline is expected to be a positive sequence so that the sum of importance scores can be in a reasonable scale.

Algorithm 2: Baseline Generation

Input : Neural network f_θ, Anomalous sample S, Training set $\mathcal{D}$, Replacement Threshold τ
Output: Baseline S'
$i = 0$
while $f_\theta(S)$ *is not normal* & $i < \tau$ **do**
  $s_t \leftarrow$ Select the event in S with the lowest frequency based on $\mathcal{D}$
  $s_t \leftarrow s_{t-1}$, $i+ = 1$
$S' \leftarrow S$
return S'

Specifically, to generate the baseline sequence for an anomalous sequence, we first sort the events based on their frequencies in the training set $\mathcal{D}$ and then replace the lowest frequent event s_t with its preceding event s_{t-1} to generate a new sequence. We evaluate whether this new sequence is a normal sequence or not by the neural network f_θ. If this new sequence is still predicted as anomalous, we further replace the currently lowest frequent event with its preceding event until the generated sequence is predicted as normal based on f_θ. We consider the generated sequence as a baseline S' for the original sequence S to derive the IG values. The motivation of replacing the low-frequency events with their preceding events is that a normal sequence should have some sort of integrity at the event level. If there is a low-frequency event in the sequence, we aim at replacing that event with a normal event and keep the integrity of the sequence.

Meanwhile, followed by the strategy proposed in [35], we expect the generated baseline has a short distance to the original sequence in the embedding space. Hence, we set a maximum replacement number τ as a threshold. Once we place τ events in the original sequence, we will stop the replacement disregard the predicted label of the current generated baseline. Algorithm 2 shows the procedure of baseline generation for an anomalous sequence. It is worth noting that the purpose of the above procedure is to generate baselines instead of detecting anomalous events. We will show in our experiments that simply labeling the low-frequency events as anomalous events cannot achieve good performance.

After generating the baseline S', we can derive the anomalous scores of events in a sequence S. Based on the definition of IG, if we consider the anomalous sequence as a positive class, the events with positive scores are anomalous, i.e., making positive contributions to the prediction. Hence, by default, we can set a threshold $\eta = 0$ to identify the anomalous events. Moreover, if we have a small validation set consisting of anomalous sequences with fine-grained labeled information, we can further leverage the validation set to fine-tune the detection threshold η to identify an optimal value that can lead to better performance on anomalous event detection.

IV. Experiments

A. Experimental Setup

1) Datasets: We apply our framework on detecting the anomalous log sequences on the following three log datasets.

- Hadoop Distributed File System (HDFS) [9]. HDFS dataset is generated by running Hadoop-based map-reduce jobs on Amazon EC2 nodes and manually labeled through handcrafted rules to identify anomalies. HDFS dataset consists of 11,172,157 log messages.
- BlueGene/L Supercomputer System (BGL) [7]. BGL dataset contains 4,747,963 log messages that are collected from a BlueGeme/L supercomputer system at Lawrence Livermore National Labs. The log messages can be categorized into alert and not-alert messages. There are 348,460 alert messages that are labeled as anomalous.
- Thunderbird [7]. Thunderbird dataset is another large-scale system log dataset that is collected from a Thunderbird supercomputer system at Sandia National Labs. We select the first 5,000,000 log messages from the original dataset for our experiment.

Both BGL and Thunderbird datasets provide the fine-grained label for each log message, so we adopt these two datasets to evaluate our framework for anomalous sequence and event detection. The HDFS dataset only has labels in the sequence level, so we only use the HDFS dataset to evaluate the anomalous sequence detection. All datasets are available online [1].

Log Data Preprocessing. The raw log messages in the three datasets are unstructured text data. Following the typical preprocessing approach, we first we adopt the log parser, Drain [36], to extract log keys (string templates) from log messages (shown in Figure 2).

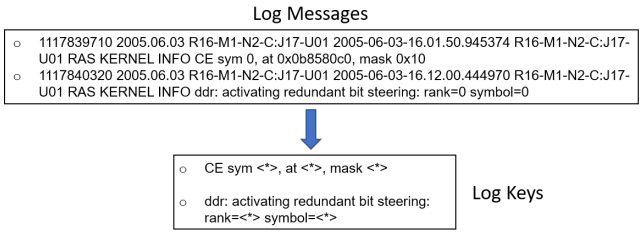


Fig. 2: Log messages and corresponding log keys

Then, similar to previous studies [37], for BGL and Thunderbird datasets, we adopt a sliding window to generate appropriate sequences. Especially for our experiment, we define the sliding window with a window size of 100 and a step size of 20. For HDFS, we group log keys into log sequences based on the session ID in the log messages. We compose a training dataset $\mathcal{D}$ that consists of 100,000 normal log sequences from each log dataset. Without a particular note, we generate an anomalous dataset $\mathcal{D}^*$ with 2,000,000 anomalous sequences, which is 20 times larger than the training dataset from each log

[1] https://github.com/logpai/loghub/

Authorized licensed use limited to: USU - Space Dynamics Laboratory. Downloaded on January 25,2022 at 04:00:06 UTC from IEEE Xplore. Restrictions apply.

dataset. When generating $\mathcal{D}^*$, the number of replaced log keys r in Algorithm 1 for each sequence is randomly set with the range from zero to the length of original sequence. The testing dataset consists of both normal and anomalous sequences. The statistics of test datasets are listed in Table I. The number in the brackets under the column "# of Unique Log Keys" indicates the number of unique log keys in the training dataset. Since in this work, we focus on the scenario that the anomalous events are rare, for BGL and Thunderbird datasets, we only select the anomalous sequences with anomalous log keys less than or equal to 10% in the testing sets.

TABLE I: Statistics of Test Datasets

Dataset	# of Unique Log Keys	# of Log Sequences		# of Log Keys in Anomalous Sequences	
		Normal	Anomalous	Normal	Anomalous
HDFS	48 (19)	458,223	16,838	N/A	N/A
BGL	396 (318)	19,430	4,190	326,491	7,139
Thunderbird	806 (774)	22,538	76,189	6,866,417	479,883

2) Baselines: We use two sets of baselines to evaluate the performance of InterpretableSAD for anomalous sequence and event detection, respectively. Note that to distinguish the terminology "baseline" used as a benchmark for experiments and the generated input sequence for the IG method, in this section, we call the baseline used in IG as "feature attribution baseline".

Baselines for Anomalous Log Sequence Detection

- Principal Component Analysis (PCA) [9]. PCA builds a counting matrix based on the frequency of log keys and then map the original counting matrix into a low dimensional space. PCA-based anomaly detection can efficiently detect extreme values.
- One-Class SVM (OCSVM) [11]. One-Class SVM is a one-class classification model that can detect anomalies based on the observed normal samples.
- Isolation Forest (iForest) [38]. Isolation forest is a tree-based anomaly detection method. It constructs trees based on the features in normal samples and captures the anomalies that deviate from normal samples.
- LogCluster [16]. LogCluster is a clustering-based one-class approach, which groups normal samples into clusters and detects the anomalies based on distances to the clusters.
- DeepLog [1]. DeepLog is a deep learning-based log anomaly detection approach. DeepLog utilizes LSTM to model the patterns of normal log sequences by training on a normal dataset and detects the anomalous sequences based on the log key prediction. If DeepLog cannot correctly predict the next log key in a sequence, the sequence will be labeled as anomalous.
- LogAnomaly [2]. LogAnomaly is another deep learning approach for anomaly detection. It combines sequential and quantitative patterns to discover the anomalous log sequences. Similarly to DeepLog, the anomalous sequence is detected based on whether the LSTM model, which is trained on the normal samples, can correctly predict the next log key.

Baselines for Anomalous Event Detection

- Anchors [28]. Anchors is a model-agnostic algorithm for interpretation of any black-box classification model. Anchors discovers a decision rule (anchors) for each input sample, and identified anchors contain essential parts of the input that determine the prediction. To conduct a fair comparison for anomalous event detection, we adopt our neural network model f_θ trained based on normal and generated anomalous samples as the black-box classification model.
- Low-Freq. The baseline for deriving the integrated gradients of each anomalous sequence is generated by replacing the low frequency events with high frequency events in the context. We further evaluate the performance of only considering the low frequency events in the sequences as anomalous events.
- Integrated Gradients (IG). We also evaluate the performance of IG without using our feature attribution baseline generation algorithm (shown in Algorithm 2) for anomalous event detection. We adopt the zero embedding matrix as the feature attribution baseline, which is widely used in text classification tasks.

3) Evaluation Metrics: We consider the anomalous class as the target class and adopt Precision, Recall, and F1 score to measure the performance of our framework.

4) Implementation Details: Regarding baselines, we leverage the package *Loglizer* [14] to evaluate PCA, OCSVM, iForest as well as LogCluster, and adopt the open source deep learning-based log analysis toolkit *LogDeep* to evaluate DeepLog and LogAnomaly [2]. We use the open source repository of Anchor to evaluate its performance on anomalous event detection [3].

Regarding our model, We adopt the long short-term memory (LSTM) network as the neural network model f_θ for anomaly detection. For BGL and Thunderbird datasets, the embedding size of log keys is 8, while for the HDFS dataset, the embedding size is 4 due to the small number of unique log keys (19 in the training set). Regarding the LSTM structures, we set different hyper-parameters on the basis of the characteristics of each dataset. For the BGL dataset, we use a single-direction LSTM with the hidden size of 128; for the Thunderbird dataset, we use a bidirectional LSTM with the hidden size of 256; for the HDFS dataset, we use a single-direction LSTM with the hidden size of 64. The number of training epochs is set as 10 for all datasets. Our code is available online [4].

B. Experimental Results on Anomalous Log Sequence Detection

Table II shows the performance of our model as well as baselines for anomalous sequence detection on three datasets. On the BGL dataset, only PCA can achieve reasonable performance, while all other baselines have poor F-1 scores.

[2]https://github.com/donglee-afar/logdeep
[3]https://github.com/marcotcr/anchor
[4]https://github.com/hanxiao0607/InterpretableSAD

Authorized licensed use limited to: USU - Space Dynamics Laboratory. Downloaded on January 25,2022 at 04:00:06 UTC from IEEE Xplore. Restrictions apply.

TABLE II: Results on Anomalous Log Sequence Detection

Method	BGL			Thunderbird			HDFS		
	Precision	Recall	F-1 score	Precision	Recall	F-1 score	Precision	Recall	F-1 score
PCA	67.91	99.79	80.82	94.83	84.43	89.33	97.77	42.12	58.88
iForest	73.13	38.19	50.17	95.06	17.92	30.15	41.59	58.80	48.72
OCSVM	24.60	100	39.49	87.13	100	93.12	6.68	90.58	12.44
LogCluster	8.03	15.97	10.69	86.56	22.94	36.26	98.37	67.45	80.03
DeepLog	42.39	52.08	46.74	82.42	81.36	81.89	56.98	48.37	52.32
LogAnomaly	42.58	53.17	47.29	81.69	82.11	81.90	55.85	48.03	51.65
InterpretableSAD	94.25	88.47	**91.27**	97.31	96.42	**96.86**	92.31	87.04	**89.60**

PCA can achieve an extremely high recall value, but cannot find a good balance between precision and recall on the anomalous sequence detection. On the Thunderbird dataset, all the baselines can achieve reasonable values in terms of precision, while the iForest and LogCluster fail to gain an acceptable performance on recall values. It means that they can only detect a small number of anomalous sequences. On the HDFS dataset, most baselines cannot achieve good performance. Meanwhile, surprisingly, on all three datasets, the deep learning-based approaches, DeepLog and LogAnomaly, cannot achieve remarkable performance even compared with the traditional anomaly detection models, like PCA. This could be because for BGL and Thunderbird, we focus on a more challenging scenario that aims at detecting the anomalous log sequences with small ratios of anomalous log keys (less than 10%). When only having a small number of anomalous events in a log sequence, the anomalous signal is not strong enough to make the models label it as anomalous. For the HDFS dataset, we generate the log sequences based on session IDs, which leads to long sequences, while DeepLog and LogAnomaly detect anomalous sequences based on the prediction accuracy of the last log keys, which is insufficient for long sequences. On the other hand, InterpretableSAD achieves the best performance in terms of F-1 scores on all three datasets. It means that the negative samples generated based on the Algorithm 1 represent the true anomalous log sequences in real datasets. Meanwhile, the good performance also show that once we can generate appropriate negative samples, a classification model that is trained on two classes of log sequences can achieve better performance compared with one-class models.

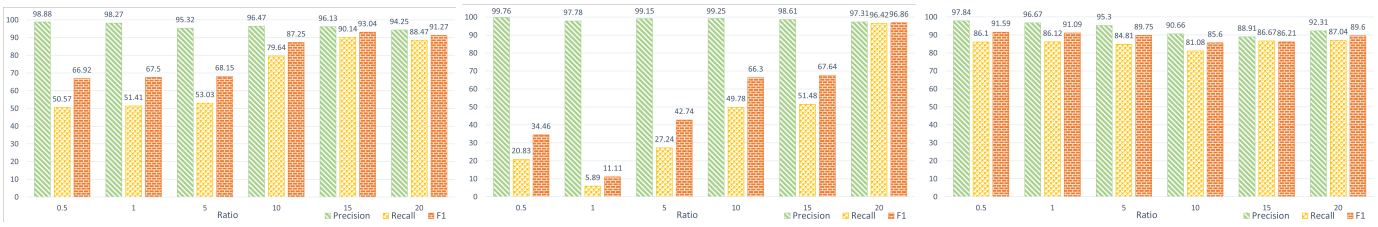
Sensitivity analysis on the size of generated anomalous sequences. In our work, because we adopt negative sampling to generate potential anomalous sequences, technically, we can generate an infinite number of anomalous sequences. We further investigate the impact of generated anomalous sample size on anomaly detection performance. In particular, we generate six anomalous datasets with different sizes, where the ratios of generated anomalous datasets $|\mathcal{D}^*|$ to the training dataset $|\mathcal{D}|$ are 0.5, 1, 5, 10, 15, 20, respectively. Figure 3 shows the performance of anomaly detection on three datasets by training on different sizes of datasets. We have the following observations. First, for all the datasets, the precision values are high for different training sizes. Second, for the BGL and Thunderbird datasets, the recall values almost keep increasing along with the increase of sizes of anomalous datasets. For example, for the BGL dataset (shown in Figure 3a), the best performance is achieved when the size of generated anomalous dataset is 15 times larger than the training dataset, while for the Thunderbird dataset (shown in Figure 3b), a good performance is achieved only when the anomalous dataset is 20 times larger than the training dataset. It indicates that after training on a set of generated anomalous samples, the classification model can always detect some anomalies based on the observed samples. Hence, the precision values are high even with a small set of anomalous samples. It also shows the effectiveness of the negative sampling algorithm. However, in order to detect more anomalies (increasing recall), we need to generate more anomalous samples to cover various anomalous scenarios. Once we have sufficient anomalous samples to train the classification model, we can get good recall values as well as the F-1 scores. For HDFS, we notice that the overall performance keeps stable over different sizes of anomalies. This is because HDFS only has 19 unique log keys in the training set, which means the search space is relatively small. Comparing with BGL and Thunderbird, which have 318 and 774 unique log keys, respectively, the number of potential anomalous scenarios in HDFS is much smaller. As a consequence, generating 50,000 anomalies is sufficient enough to cover most of the anomalous scenarios.

Visualization. We consider the last hidden state in the LSTM model as the sequence representation and adopt the t-SNE algorithm [39] to map the sequence representations into a two-dimensional space. For each dataset, we randomly select 1000 normal, anomalous, and generated samples, separately. As shown in Figure 4, for all datasets, the generated samples via negative sampling can cover the space of real anomalous samples. Especially, for BGL and HDFS datasets (Figures 4a and 4c), the points of generated anomalous samples and true anomalous samples are highly overlapped, while the majority of normal samples are outside the regions of anomalous samples. For the Thunderbird dataset (Figure 4b), the generated samples and abnormal samples are on left side of the space, while the normal samples are on the right side. Based on the visualization results, it is straightforward to notice that the LSTM model trained on the normal sequences and generated anomalous sequences can detect the real anomalous sequences for all three datasets.

C. Experimental Results on Anomalous Event Detection

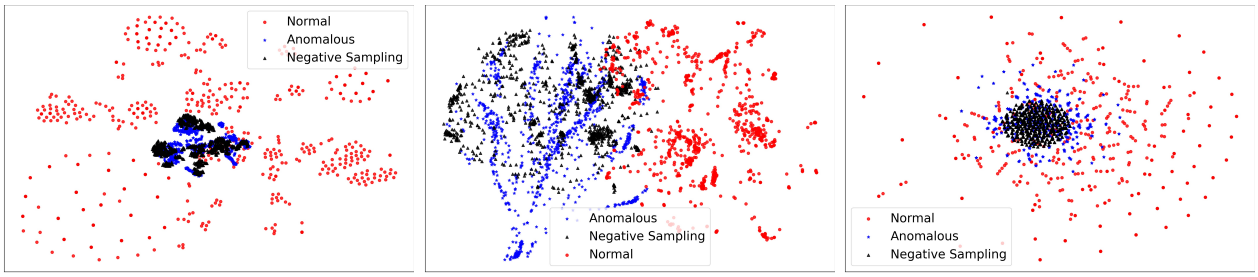
We then study the performance of InterpretableSAD on anomalous event detection. When evaluating InterpretableSAD

Authorized licensed use limited to: USU - Space Dynamics Laboratory. Downloaded on January 25,2022 at 04:00:06 UTC from IEEE Xplore. Restrictions apply.



(a) BGL (b) Thunderbird (c) HDFS

Fig. 3: Impact of the negative sampling ratio on the anomalous sequence detection



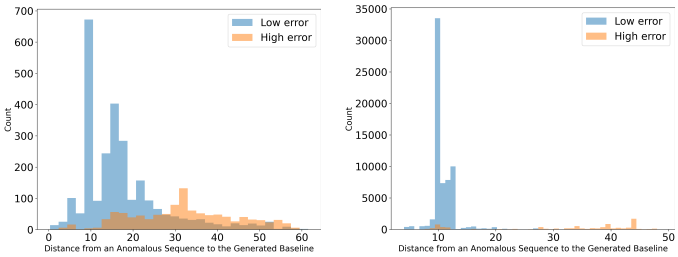
(a) BGL (b) Thunderbird (c) HDFS

Fig. 4: Visualization of the normal, anomalous, and generated anomalous sequences.

TABLE III: Results on Anomalous Event Detection

Method	BGL			Thunderbird		
	Precision	Recall	F-1 score	Precision	Recall	F-1 score
Anchors	0.31	8.56	0.60	4.58	14.62	6.98
Low-Freq	38.76	93.59	54.82	52.61	99.00	68.70
IG w/o val	6.56	90.27	12.23	10.36	85.65	18.49
IG w/ val	42.43	73.83	53.89	20.92	44.48	28.45
InterpretableSAD w/o val	50.87	89.23	64.80	94.98	86.79	90.70
InterpretableSAD w/ val	68.92	82.53	**75.11**	93.84	98.31	**96.02**

and IG with the zero embedding matrix as the feature attribution baseline, we consider two scenarios, with or without a validation set consisting of 10% anomalous sequences in the testing datasets to tune a detection threshold η. Recall that we only consider the events with anomalous scores greater than η are anomalous. The default value of η is 0 without tuning on a validation set. As shown in Table III, InterpretableSAD with a validation set achieves the best performance for anomalous event detection on both datasets. Meanwhile, even we use the default threshold $\eta = 0$ to detect the anomalous events, the performance is still good. The performance of IG with the zero embedding matrix as feature attribution baseline is poor, even we use a validation set to tune η. It indicates the importance of designing a good feature attribution baseline for the IG model, and the zero embedding matrix that is widely used as the feature attribution baseline in interpreting text classification models is not suitable for sequential anomaly detection. Moreover, we notice that simply labeling low frequent events as anomalous cannot achieve good results in terms of precision and F-1 score, even though the recall values are high on both datasets. It indicates that anomalous events are usually low frequent in the training dataset, but many normal events could also have low frequent, which will lead to low precision. It is hard to balance the precision and recall simply based on the frequency of events. For Anchors, it cannot achieve reasonable performance on both datasets. This could be because long sequences have huge search spaces to locate the anchors.



(a) BGL (b) Thunderbird

Fig. 5: The correlation between the performance of anomalous event detection and the distances from sequences to corresponding baselines. Low error indicates given an anomalous sequence, InterpretableSAD correctly detect at least 80% of anomalous events.

Sensitivity analysis on the distances between sequences and the feature attribution baselines. The performance of Integrated Gradients heavily relies on the feature attribution baselines. When we generate the feature attribution baselines, we expect the baseline has a short distance to the original sequence. To further show the impact of baselines on anomalous event detection, we consider the anomalous event detection results into two categories, low error and high error. If InterpretableSAD correctly detects at least 80% of anomalous events in an anomalous sequence, we consider

Authorized licensed use limited to: USU - Space Dynamics Laboratory. Downloaded on January 25,2022 at 04:00:06 UTC from IEEE Xplore. Restrictions apply.

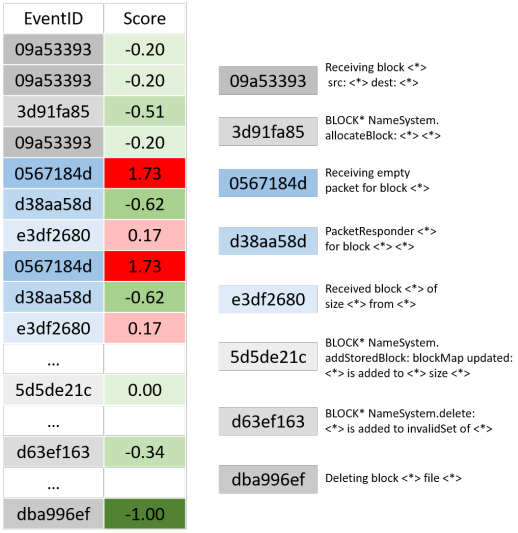


Fig. 6: An anomalous sequence in the HDFS dataset and the corresponding anomalous scores

the prediction result as low error; otherwise, we consider the prediction result as high error. Then, we explore the correlation between the performance of anomalous event detection and the distances from sequences to corresponding feature attribution baselines. The distance is the L2 distance between the embedding matrices of the original sequence and baseline sequence. As shown in Figure 5, for both datasets, if baseline sequences have small distances to the original anomalous sequences, in most cases, we can achieve low error for anomalous event detection. For example, because we achieve a high F-1 score (96.02%) on the Thunderbird dataset, most sequences have small distances to the corresponding feature attribution baselines (shown in Figure 5b), while only a few sequences have large distances to the baselines. Similar observations on Figure 5a, the majority of sequences with high errors have larger distances to the feature attribution baselines. Hence, based on Figure 5, we have two findings. First, IG is sensitive to the feature attribution baselines, so choosing a good baseline is critical for anomalous event detection. Second, in practice, a good feature attribution baseline should meet the following requirements: 1) the feature attribution baseline can be predicted as normal sequence; 2) the distance from the original sequence to the feature attribution baseline should be small.

Case Study. For the HDFS dataset, we do not have the fine-grained event labels. We apply the case study to show the effectiveness of InterpretableSAD on anomalous event detection. Figure 6 shows an example of the model prediction on an anomalous sequence in the HDFS dataset. The model predicts a sequence (session ID: "blk_-4364732810285057372") with 22 events as anomalous. Besides detecting the anomalous sequence, InterpretableSAD further derives the anomalous score for each event in the sequence. Specifically, this log sequence records a set of operations about failing to create a block. We notice that the event "0567184d" has a high anomalous score (1.73), which means it is responsible for the anomalous prediction outcome. "0567184d" indicates receiving an empty packet for the block. Based on the highlighted event "0567184d", we can understand that the failure of creating a block is caused by receiving empty packets for the block several times. Meanwhile, for all other operations in this sequence, such as block allocation, adding the block to an invalid set, and block deletion, InterpretableSAD assigns negative scores, which means these operations are common in a block creating procedure and not anomalous. Based on this case study, we show that according to the derived anomalous scores, system administrators can quickly locate the exactly anomalous events without manually examining each event in a sequence. This improvement can further help the system administrators to effectively mitigate the system failure.

V. Conclusion

In this work, we have developed InterpretableSAD for anomalous log sequence and more fine-grained anomalous log event detection. Considering the rare of anomalous samples, InterpretableSAD leverages the data augmentation strategy to generate anomalous samples by proposing a novel negative sampling algorithm. Then, a binary classification model can be trained on observed normal and generated anomalous sequences. A well-trained classifier is able to detect the real anomalous sequences. Since an anomalous log sequence usually consists of a large number of events, only detecting anomalous sequences is not sufficient to help domain experts locate the exact anomalies. InterpretableSAD further applies an interpretable machine learning technique, Integrated Gradients (IG), to detect the potential anomalous events in sequences. IG is able to show the importance of each feature to the prediction outcome of the classifier. We consider the importance scores derived from IG as anomalous scores to detect anomalous events. To apply IG for anomalous event detection, we propose a novel feature attribution baseline generation algorithm because a good baseline is critical for IG to derive reasonable scores of events. Experimental results on three log datasets show that our model can achieve state-of-the-art performance on the anomalous sequence and event detection. In the future, we plan to study how to efficiently generate negative samples so that a small ratio of generated samples can still cover the majority anomalous scenarios and also explore the baseline generation algorithms for anomaly detection with theoretical guarantees. Another direction of future work is to study an intrinsically interpretable model that is able to detect anomalous sequences and events in an end-to-end manner.

Acknowledgment

This work was supported in part by NSF 2103829.

References

[1] M. Du, F. Li, G. Zheng, and V. Srikumar, "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, pp. 1285–1298.

Authorized licensed use limited to: USU - Space Dynamics Laboratory. Downloaded on January 25,2022 at 04:00:06 UTC from IEEE Xplore. Restrictions apply.

[2] W. Meng, Y. Liu, Y. Zhu, S. Zhang, D. Pei, Y. Liu, Y. Chen, R. Zhang, S. Tao, P. Sun *et al.*, "Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs." in *IJCAI*, vol. 7, 2019, pp. 4739–4745.

[3] X. Zhang, Y. Xu, Q. Lin, B. Qiao, H. Zhang, Y. Dang, C. Xie, X. Yang, Q. Cheng, Z. Li *et al.*, "Robust log-based anomaly detection on unstable log data," in *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2019, pp. 807–817.

[4] M.-h. Oh and G. Iyengar, "Sequential anomaly detection using inverse reinforcement learning," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1480–1490.

[5] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[6] G. Pang, C. Shen, L. Cao, and A. v. d. Hengel, "Deep learning for anomaly detection: A review," *arXiv preprint arXiv:2007.02500*, 2020.

[7] A. Oliner and J. Stearley, "What supercomputers say: A study of five system logs," in *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07)*. IEEE, 2007, pp. 575–584.

[8] T.-F. Yen, A. Oprea, K. Onarlioglu, T. Leetham, W. Robertson, A. Juels, and E. Kirda, "Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks," in *Proceedings of the 29th Annual Computer Security Applications Conference*, 2013, pp. 199–208.

[9] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan, "Detecting large-scale system problems by mining console logs," in *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*, 2009, pp. 117–132.

[10] Y. Wang, J. Wong, and A. Miner, "Anomaly intrusion detection using one class svm," in *Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, 2004.* IEEE, 2004, pp. 358–364.

[11] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[12] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International conference on machine learning*. PMLR, 2018.

[13] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3319–3328.

[14] S. He, J. Zhu, P. He, and M. R. Lyu, "Experience report: System log analysis for anomaly detection," in *2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2016, pp. 207–218.

[15] Y. Liang, Y. Zhang, H. Xiong, and R. Sahoo, "Failure prediction in ibm bluegene/l event logs," in *Seventh IEEE International Conference on Data Mining (ICDM 2007)*. IEEE, 2007, pp. 583–588.

[16] R. Vaarandi and M. Pihelgas, "Logcluster-a data clustering and pattern mining algorithm for event logs," in *2015 11th International conference on network and service management (CNSM)*. IEEE, 2015, pp. 1–7.

[17] K. Zhang, J. Xu, M. R. Min, G. Jiang, K. Pelechrinis, and H. Zhang, "Automated it system failure prediction: A deep learning approach," in *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 2016, pp. 1291–1300.

[18] F. Liu, Y. Wen, D. Zhang, X. Jiang, X. Xing, and D. Meng, "Log2vec: a heterogeneous graph embedding based approach for detecting cyber threats within enterprise," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 1777–1794.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[21] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," *arXiv preprint arXiv:1901.11196*, 2019.

[22] S. Kobayashi, "Contextual augmentation: Data augmentation by words with paradigmatic relations," *arXiv preprint arXiv:1805.06201*, 2018.

[23] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[24] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Neural Information Processing Systems (NIPS)*, 2013, pp. 1–9.

[25] J. Ding, Y. Quan, X. He, Y. Li, and D. Jin, "Reinforced negative sampling for recommendation with exposure data." in *IJCAI*, 2019, pp. 2230–2236.

[26] C. Molnar, *Interpretable machine learning*. Lulu. com, 2020.

[27] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Communications of the ACM*, vol. 63, no. 1, pp. 68–77, 2019.

[28] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[29] ——, "" 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*, 2016, pp. 1135–1144.

[30] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *arXiv preprint arXiv:1705.07874*, 2017.

[31] Y.-H. H. Tsai, M. Ma, M. Yang, R. Salakhutdinov, and L.-P. Morency, "Multimodal routing: Improving local and global interpretability of multimodal language analysis," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[32] A. Brown, A. Tuor, B. Hutchinson, and N. Nichols, "Recurrent neural network attention mechanisms for interpretable system log anomaly detection," in *Proceedings of the First Workshop on Machine Learning for Computing Systems*, 2018, pp. 1–8.

[33] N. Liu, X. Huang, and X. Hu, "Accelerated local anomaly detection via resolving attributed networks." in *IJCAI*, 2017, pp. 2337–2343.

[34] P. Sturmfels, S. Lundberg, and S.-I. Lee, "Visualizing the impact of feature attribution baselines," *Distill*, vol. 5, no. 1, p. e22, 2020.

[35] J. Sipple, "Interpretable, multidimensional, multimodal anomaly detection with negative sampling for detection of device failure," in *International Conference on Machine Learning*. PMLR, 2020, pp. 9016–9025.

[36] P. He, J. Zhu, Z. Zheng, and M. R. Lyu, "Drain: An online log parsing approach with fixed depth tree," in *2017 IEEE International Conference on Web Services (ICWS)*. IEEE, 2017, pp. 33–40.

[37] Z. Wang, Z. Chen, J. Ni, H. Liu, H. Chen, and J. Tang, "Multi-scale one-class recurrent neural networks for discrete event sequence anomaly detection," *arXiv preprint arXiv:2008.13361*, 2020.

[38] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*. IEEE, 2008, pp. 413–422.

[39] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

Authorized licensed use limited to: USU - Space Dynamics Laboratory. Downloaded on January 25,2022 at 04:00:06 UTC from IEEE Xplore. Restrictions apply.