Sequential Anomaly Detection with Local and Global Explanations

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

Depeng Xu
University of North Carolina at Charlotte
Charlotte, NC, USA
depeng.xu@uncc.edu

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

2022 IEEE International Conference on Big Data (Big Data) | 978-1-6654-8045-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigData55660.2022.10020990

***Abstract*—Sequential anomaly detection has been studied for decades because of its wide spectrum of applications and obtained significant improvement in recent years by utilizing deep learning techniques. As an increasing number of anomaly detection models are applied to high-stake tasks involving human beings, it is critical to understand the reasons why the samples are labeled as anomalies. In this work, we propose a Globally and Locally Explainable Anomaly Detection (GLEAD) framework targeting sequential data. Especially, considering that the anomalies are usually diverse, we make use of the multi-head self-attention techniques to derive representations for sequences as well as prototypes, which capture a variety of patterns in anomalies. The attention mechanism highlights the abnormal entries with high attention weights in the abnormal sequences for the local explanation. Moreover, the prototypes of anomalies encoding the common patterns of abnormal sequences are derived to achieve the global explanation. Experimental results on two sequential anomaly detection datasets show that our approach can detect abnormal sequences and provide local and global explanations.**

***Keywords*—sequential data, anomaly detection, explanations**

I. Introduction

Anomaly detection on sequential data is an important subfield o f a nomaly d etection b ecause s equential d ata are ubiquitous in various applications. For example, detecting anomalies in log messages plays a critical role to build robust and reliable computer systems [1]. Recently, deep learning-based approaches have been developed and made big progress in sequential anomaly detection [2]. However, one limitation of the existing deep anomaly detection approaches is that as black-box models, they cannot provide explanations for why samples are labeled as anomalies. On the other hand, in practice, understanding why sequences are classified as abnormal is important and useful, which can help domain experts quickly locate the exact issue.

Explanations for sequential anomaly detection can be conducted at two levels, the local and global levels. The local explanation focuses on each individual sequence and aims to highlight the abnormal entries in the abnormal sequence. The global explanation shows the general patterns of abnormal sequence over the whole dataset. In our scenario, we consider the general patterns as types of abnormal behaviors represented by abnormal entries. For example, if a computer system is under various attacks, the global explanation is to group the abnormal entries to represent each type of attack.

In this work, we aim to equip the deep anomaly detection approach with an explainable component, which is able to not only detect anomalies but also provide local and global explanations for detection results. To this end, we propose a Globally and Locally Explainable Anomaly Detection (GLEAD) framework. Especially, considering that in real cases, the sequential anomalies can be diverse, GLEAD adopt the multi-head self-attention technique [3], [4] to derive representations of sequences, where each head learns to focus on one type of pattern. Therefore, the multi-head model is expected to capture different abnormal patterns. Meanwhile, GLEAD further derives the prototypes of normal and abnormal sequences and ensures the normal sequences close to the normal prototypes while abnormal sequences close to the abnormal prototypes. For the detected abnormal sequence, based on the head of its closest abnormal prototype, its representation and attention under this head best provide the information for global and local explanations. For the local explanation, GLEAD highlights the abnormal entries in the abnormal sequence resulting in the predicted label. For the global explanation, GLEAD picks out a list of top entries for each abnormal prototype that describes the common pattern of the abnormal sequences under that head. Experimental evaluation shows that our approach is able to detect abnormal sequences and provide local and global explanations.

II. Related Work

A. Anomaly Detection for Sequential Data

Because of the rarity of anomalies, it is difficult to obtain a large amount of well-labeled abnormal sequences for training a supervised model. Current anomaly detection approaches are usually trained in an unsupervised or semi-supervised manner, where only normal samples or a few labeled normal and abnormal samples, as well as a large number of unlabeled samples, are available [5]–[7]. These approaches share the same fundamental idea which is to learn patterns from normal data, so sequences that deviate from normal patterns can be labeled as abnormal ones. However, these approaches can make accurate predictions on abnormal sequence detection, but do not provide explanations of the decision results, which limits the wide application in the real world.

B. Explainable Machine Learning

Based on the scope of explainability, the approaches for explainable machine learning models can be categorized into local and global explanation approaches.

Local Explanation. Local explanation aims to explain individual predictions. A common idea to achieve the local explanation is the perturbation-based strategy, which provides post-hoc explanation to the prediction outcome by checking the performance change after perturbing the input features [8], [9]. For example, Anchors [9] aims to find a decision rule that can sufficiently make the same prediction as the original input. However, most of the existing explainable models are developed in a supervised setting, while anomaly detection models are usually trained in an unsupervised or semi-supervised setting.

Global Explanation. Global explanation methods mainly describe the behavior of a specific model or give explanations over an entire set of data instances. One typical approach is to detect a set of prototypes that can be used to represent the behaviors of a model [10], [11]. Meanwhile, another type of approach is to train a global surrogate model to imitate the behaviors of the black-box model [12]–[14], where the surrogate model is an explainable model, such as linear regression or logistic regression. Global explanation methods can explain the behaviors of black-box models, but the explanations can not always work for any single data instance.

III. Methodology

A. Overview

We aim to detect abnormal sequences with local and global explanations in a semi-supervised setting. Let $S = (e_1, \ldots, e_l, \ldots, e_L)$ be a sequence consisting of L entries. We assume the availability of a small set of labeled sequences $\mathcal{S} = \mathcal{S}^+ \cup \mathcal{S}^-$, where $\mathcal{S}^+ = \{(S_i^+, y_i = 0)\}_{i=1}^{|\mathcal{S}^+|}$ indicates a small set of normal sequences and $\mathcal{S}^- = \{(S_i^-, y_i = 1)\}_{i=1}^{|\mathcal{S}^-|}$ indicates a small set of abnormal sequences. Meanwhile, there is a large number of unlabeled sequences $\mathcal{U} = \{S_j\}_{j=1}^{|\mathcal{U}|}$. Following the basic assumption in anomaly detection, we assume a majority of sequences in $\mathcal{U}$ are normal.

Given $\mathcal{S}$ and $\mathcal{U}$, we aim to build a sequential anomaly detection model with local and global explanations. The goal of the local explanation is to highlight the suspicious entries in each abnormal sequence, while the global explanation is to find the abnormal entries that represent the common pattern of a group of anomalies. To achieve local and global explanations, we develop a Globally and Locally Explainable Anomaly Detection (GLEAD) framework which simultaneously learns the individual representation of each sequence as well as the prototype representations, where the prototype representations encode hidden patterns of a group of sequences that are similar. Considering the diversity of sequences, especially the abnormal sequences, we use multiple prototypes to capture various hidden patterns. Specifically, we leverage the multi-head self-attention network to derive the representation matrix of each individual sample and also to train prototype matrices

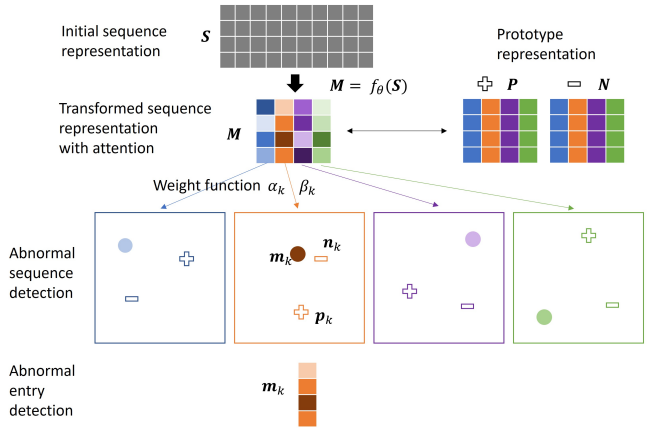

Fig. 1: Illustration of the Globally and Locally Explainable Anomaly Detection (GLEAD) framework by leveraging multiple prototypes for anomaly detection.

to capture the globally normal and abnormal patterns. Each column in the prototype matrices represents a channel for one prototype. The attention of each sequence weights on which channel the sequence is closer to. First, we compare the individual sequence representations to each channel of the prototype matrices to detect abnormal sequences. If a sequence is closer to the abnormal prototypes, the sequence will be labeled as abnormal. Then, the entry-level attention of the sequence and prototype representations are used to achieve local and global explanations, respectively. The key idea is to use an attention mechanism to highlight the abnormal entries with high attention weights in an abnormal sequence for the local explanation and to further use the prototypes of abnormal sequences to derive the common patterns of abnormal sequences for the global explanation. Figure 1 illustrates the GLEAD framework.

B. Sequence Representation

The first component of GLEAD is to derive the sequence representations. Traditionally, a sequence is usually represented by a vector in a hidden space derived from a neural network. However, considering the diversity of sequential data, representing a sequence as one vector could not sufficiently capture the variety of underlying patterns. Therefore, we develop a multi-head self-attention model to derive the representations of a sequence as a matrix, where each column (head) captures one aspect of common patterns that may be closely related to a group of sequences. Meanwhile, the corresponding attention matrix can be used for refining important entries in a sequence, where each attention weight vector highlights the entries related to one underlying pattern.

In particular, given a sequence S, we first represent each entry e_l in the sequence as an embedding vector, and then the sequence can be represented as $\mathbf{S} = [\mathbf{e}_1, \ldots, \mathbf{e}_L] \in \mathbb{R}^{d \times L}$, where d is the hidden dimension of the embedding vector. Then, the multi-head self-attention network takes the initial sequence representation $\mathbf{S}$ as the input and outputs an attention weight matrix $\mathbf{A} \in \mathbb{R}^{L \times r}$, which refines the importance of

each entry in the sequence and the closeness to each head. Specifically, the embedding matrix $\mathbf{S}$ is fed into a multi-layer perceptron (MLP) with one hidden layer by using tanh as the activation function. After that, a softmax function is adopted to derive the attention weights. Formally, the attention matrix $\mathbf{A}$ is computed as

$$\mathbf{A} = softmax(tanh(\mathbf{S}^T\mathbf{W}_1)\mathbf{W}_2), \tag{1}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d\times h}$ and $\mathbf{W}_2 \in \mathbb{R}^{h\times r}$ are trainable parameters, and h indicates the hidden dimension and r is the number of attention heads. The softmax function is applied column-wise so that each column vector in $\mathbf{A}$ has the attention weight vector sum up to 1. Then, the transformed sequence representations matrix $\mathbf{M} \in \mathbb{R}^{d\times r}$ can be derived as

$$\mathbf{M} = \mathbf{SA}. \tag{2}$$

Based on Equation 2, each column vector, i.e., an aspect representation of the sequence, is computed as a weighted linear combination of L entries in the sequence. For example, the k-th representation vector is computed as $\mathbf{m}_k = \mathbf{S}\mathbf{a}_k$.

We denoted the multi-head self-attention model as $\mathbf{M} = f_\theta(\mathbf{S})$ with trainable parameters $\theta = \{\mathbf{W}_1, \mathbf{W}_2\}$. The transformed representation $\mathbf{M}$ takes the attention into account, so it provides intuitive explanations to anomaly detection.

C. Prototypes of Normal and Abnormal Sequences

By using the multi-head self-attention model, we can encode a sequence S into a sequence embedding matrix $\mathbf{M}$. To further capture the global patterns of normal and abnormal sequences, we define prototype representations for normal sequences $\mathbf{P} \in \mathbb{R}^{d\times r}$ and abnormal sequences $\mathbf{N} \in \mathbb{R}^{d\times r}$ as matrices, respectively. Both prototype matrices are randomly initialized and will be updated during training.

The purpose of defining prototype matrices is that there could be various patterns in normal or abnormal sequences. We encode each prototype as a d-dimensional representation. Each prototype captures one type of hidden patterns in the normal or abnormal sequences. The sequences that are similar to the prototype devote high attention onto this dimension. We use r representation vectors to capture up to r different types of hidden patterns for normal or abnormal sequences. Note there are likely new types of anomalies unseen in the training set due to the rarity of anomaly sequences. We assume the model is agnostic to how many types and what kinds of anomalies are out there. We set r at a sufficient number to provide enough placeholders for all potential anomaly types. We expect that by decoding the prototype representations of abnormal sequences, we can provide global explanations for each type of anomaly.

D. Objective Function

The training objective consists of two parts. The first objective is to make the normal samples close to the normal prototypes while the abnormal samples close to the abnormal prototypes. The second objective is to ensure that the learned prototype representations are diverse.

Weighted Triplet Loss. The triplet loss minimizes the distance from an anchor to a positive sample and maximizes the distance from the anchor to a negative sample. We leverage the idea of triplet loss to ensure a sample close to the corresponding prototype and far away from the counterpart prototype. However, one limitation of the vanilla triplet loss is that both positive and negative samples as well as the anchor should be represented as a vector in order to calculate the distance. As we represent the sequences and prototypes as a set of vectors, we cannot directly apply the triplet loss.

To extend the triplet loss to handle multiple dimensions and to leverage multiple sets of anchors, we develop a weighted triplet loss function. Given an abnormal sequence S^-, after obtaining the sequence representation as $\mathbf{M}^- = f_\theta(\mathbf{S}^-)$, the weighted triplet loss is defined as

$$\begin{aligned} T(\mathbf{N}, \mathbf{P}, \mathbf{M}^-) = \sum_{k=1}^{r} \max\{&\alpha_k(\mathbf{n}_k, \mathbf{M}^-)t(\mathbf{n}_k, \mathbf{m}_k^-) \\ &- \beta_k(\mathbf{p}_k, \mathbf{M}^-)t(\mathbf{p}_k, \mathbf{m}_k^-) + \mu, 0\}, \end{aligned} \tag{3}$$

where $t(u, v) = \frac{1}{2}(1 - \cos(u, v))$ indicates the cosine distance between u and v, μ indicates the margin, and $\alpha_k(\mathbf{n}_k, \mathbf{M}^-)$ and $\beta_k(\mathbf{p}_k, \mathbf{M}^-)$ indicates two weight functions, defined as

$$\begin{aligned} \alpha_k(\mathbf{n}_k, \mathbf{M}^-) &= \frac{exp(-t(\mathbf{n}_k, \mathbf{m}_k^-))}{\sum_{k'=1}^{r} exp(-t(\mathbf{n}_{k'}, \mathbf{m}_{k'}^-))}, \\ \beta_k(\mathbf{p}_k, \mathbf{M}^-) &= \frac{exp(t(\mathbf{p}_k, \mathbf{m}_k^-))}{\sum_{k'=1}^{r} exp(t(\mathbf{p}_{k'}, \mathbf{m}_{k'}^-))}. \end{aligned} \tag{4}$$

We adopt r heads to capture variant patterns of abnormal samples and further develop prototypes with r representations to encode the common patterns. The weight function $\alpha_k(\mathbf{n}_k, \mathbf{M}^-)$ computes the weights between the sequence representations $\mathbf{M}^-$ and abnormal prototype representations $\mathbf{N}$ based on their vector-wise distances. The idea is that one abnormal sequence usually contains one type of abnormal pattern, which means it should be close to one prototype representation, say $\mathbf{n}_k$. Then, the weight function would set a high weight between the sequence representation $\mathbf{m}_k^-$ and the prototype representation $\mathbf{n}_k$. By training in this way, one head of self-attention in $f_\theta(\mathbf{S}^-)$ should focus on one pattern. Meanwhile, as the prototype matrix $\mathbf{N}$ is a trainable parameter, one prototype representation $\mathbf{n}_k$ would be also trained to capture one pattern of abnormal samples. On the other hand, the abnormal sequence should have large distances to the normal prototype representations. Therefore, the weight function $\beta_k(\mathbf{p}_k, \mathbf{M}^-)$ sets a high weight if the abnormal sequence has a large distance to one normal prototype representation.

Similarly, for the normal sequence S^+, we can define the weighted triplet loss as $T(\mathbf{P}, \mathbf{N}, \mathbf{M}^+)$, which is to make the representations of the normal sequence close to the normal prototypes and far from the abnormal prototypes. Then, the

final objective function is

$$\mathcal{L}_t = \frac{1}{|\mathcal{U}|+|\mathcal{S}^+|}\sum_{i=1}^{|\mathcal{U}|+|\mathcal{S}^+|} T(\mathbf{P},\mathbf{N},\mathbf{M}_i^+) + \frac{1}{|\mathcal{S}^-|}\sum_{j=1}^{|\mathcal{S}^-|} T(\mathbf{N},\mathbf{P},\mathbf{M}_j^-), \tag{5}$$

Diversity Constraint. To better capture the diversity of normal and abnormal patterns, we regularize the prototypes $\mathbf{N}$ and $\mathbf{P}$ to orthogonality:

$$\mathcal{L}_D = \|concat(\mathbf{N},\mathbf{P})^T concat(\mathbf{N},\mathbf{P}) - \mathbf{I}_r\|_F^2, \tag{6}$$

where $concat(\mathbf{N},\mathbf{P})$ indicates the concatenation of two prototype matrices, and $\mathbf{I}_r$ is the identity matrix. The diversity constraint ensures the prototype representations are different from each other so that different patterns can be encoded.

Then by combining Equations 5 and 6, our overall objective function can be defined as:

$$\mathcal{L} = \mathcal{L}_t + \lambda \mathcal{L}_D, \tag{7}$$

where λ is a hyperparameter.

After training, each sample should be close to one prototype representation of the same class but far from the prototypes of the other class. Meanwhile, the prototype representations are diverse in the hidden space.

Abnormal Sequence Detection. Given a testing sequence S with its representation $\mathbf{M}$, we derive the anomaly score of the sequence by comparing the distance between $\mathbf{M}$ and $\mathbf{P}$ as well as the distance between $\mathbf{M}$ and $\mathbf{N}$, which is defined as:

$$s(\mathbf{M}) = \frac{1}{r}\sum_{k=1}^{r}\big(t(\mathbf{p}_k,\mathbf{m}_k) - t(\mathbf{n}_k,\mathbf{m}_k)\big). \tag{8}$$

If the sequence representation $\mathbf{M}$ is closer to abnormal prototypes $\mathbf{N}$, i.e., $s(\mathbf{M}) > 0$, we will report it as abnormal.

E. Explanation

One advantage of our framework is that it can provide local and global explanations. The local explanation is to explain a single prediction, while the global explanation can be a description of a set of sequences, such as the common pattern of a group of sequences. In the scenario of anomaly detection, it is more critical to explain the abnormal outcomes compared with the normal ones. Therefore, we focus on deriving the local explanation for a sequence predicted as abnormal and the global explanation of abnormal patterns.

Local Explanation (Abnormal Entry Detection). Given a sequence detected as abnormal, the local explanation is to identify the abnormal entries in the sequence leading to its anomaly. We leverage the attention weights as the contribution indicators of all the entries, where a high attention weight could indicate a suspicious entry. In the training phase, we represent an abnormal sequence with r vectors and force one representation vector $\mathbf{m}_k$ close to one prototype representation $\mathbf{n}_k$. Each $\mathbf{n}_k$ captures a type of hidden pattern which is shared by the sequences close to $\mathbf{n}_k$. Therefore, if a sequence is detected as abnormal, we only use the attention weights from the "best" head that is the closest to one prototype representation. Formally, the "best" head from r heads is $b = \arg\min_k t(\mathbf{n}_k,\mathbf{m}_k)$. Then, we use b-th vector of $\mathbf{A}$, i.e., $\mathbf{a}_b$, as the attention weights to highlight the entries in S. The attention under this head captures the abnormal behavior to identify the abnormal entries. The entries with attention weights higher than a threshold will be labeled as abnormal entries. The high attentions indicate that the corresponding entries best explain the closeness to the abnormal prototype, i.e., the reason why S is labeled as an anomaly.

Global Explanation. In order to achieve the global explanation for anomalies, we make use of the prototype of abnormal sequences $\mathbf{N}$ to decipher the hidden patterns of different types of anomalies. We assume that the same type of anomalous sequences shares a common pattern in terms of the anomalous entries they include. Therefore, abnormal sequences captured by each diversified attention head (along with its abnormal prototype) should contain the same or similar abnormal entries. To study what type of anomalies the attention heads capture, we create a top entry list for each attention head by counting the abnormal entries in the abnormal sequences captured by attention heads. Specifically, for each abnormal prototype $\mathbf{n}_k$, we first choose the abnormal sequences with the "best" head representations that are closest to $\mathbf{n}_k$. Then, given each abnormal sequence closest to $\mathbf{n}_k$, we pick top-z entries with the highest attention weights in $\mathbf{a}_k$. Finally, by combining these entries, we can derive a list of the high-frequency entries with high attention weights, which decodes the hidden pattern of abnormal sequences in $\mathbf{n}_k$. Because the prototype $\mathbf{N}$ consists of r representation vectors, we can get r lists of entries showing r different patterns in abnormal sequences.

TABLE I: Statistics of Two Datasets

Dataset		BGL	Thunderbird
Training	Normal	20	20
	Abnormal	20	20
	Unlabeled	2000	2000
Validation	Normal	200	200
	Abnormal	20	20
Test	Normal	20000	20000
	Abnormal	2000	2000

IV. Experiments

A. Experimental Setup

Datasets. We evaluate the performance of GLEAD for sequential anomaly detection on two datasets: **BlueGene/L (BGL)** and **Thunderbird** [15], which are datasets consisting of system logs from BlueGene/L and Thunderbird supercomputer systems, respectively.

We use a log parser, Drain [16], to transfer raw log messages to log templates and then we apply the sliding window technique on these log templates to obtain log sequences. Table I shows the statistics of the labeled set $\mathcal{S}$ and the unlabeled set $\mathcal{U}$ of training data. The labeled sets consist of 20 normal sequences and 20 abnormal sequences. The unlabeled sets consist of 2000 sequences. We also build a small validation set for each dataset to tune the hyper-parameters and derive thresholds for identifying abnormal entries. For a fair

comparison, for all models, we keep the models with the best performance on the validation set and then apply these models on the test set to identify abnormal sequences and entries.

Baselines. We compare our approach with the following baselines for **sequential anomaly detection**.

- **Isolation Forest (iForest)** is an unsupervised anomaly detection algorithm that is built using decision trees [17].
- **One Class Support Vector Machine (OCSVM)** is a one-class novelty detection algorithm that focuses on learning the pattern of known normal data samples [18].
- **LSTM with Attention (LSTM-Attention).** We train an LSTM model with an attention layer using labeled sequences in a supervised manner.
- **DeepSAD** is a semi-supervised anomaly detection method that leverages both labeled and unlabeled samples to improve the performance of anomaly detection [7].

For iForest and OCSVM, we build a count vector to represent a sequence, where each dimension indicates a unique entry and the value indicates the frequency of the entry in the sequence. Therefore, iForest and OCSVM cannot capture the temporal information of the sequences.

Our approach can also provide local explanations for the abnormal sequences by detecting the abnormal entries. We further compare our approach with baselines that can achieve **entry-level anomaly detection**.

- **iForest** and **OCSVM**. We use iForest and OCSVM for abnormal entry detection. The input to both models for abnormal entry detection is the count vector derived from the log message, where each dimension indicates a unique word and the value indicates the word frequency.
- **LSTM-Attention.** After training the LSTM with the attention model, we also use the attention weights to predict abnormal entries.
- **Shapley** is widely used to explain individual predictions. Shapley values are calculated as contributions that features make for predictions. As a post-hoc explanation model, we train an LSTM for sequential anomaly detection and use Shapley to identify abnormal entries.

Implementation Details. In order to represent the entries in sequences as embedding vectors, we randomly initialize the embedding vectors of entries with a dimension of 150, i.e., $d = 150$. The embedding layer encodes the sequences with initialized vectors and then updates them in the training period. We set the number of attention heads as $r = 5$. . For both datasets, we train GLEAD in 150 epochs with the diversity constraint weight $\lambda = 1.0$. After training, we derive a threshold from the validation set for detecting the abnormal entries. An entry with an attention weight higher than the threshold will be labeled as abnormal. The source code is available online **https://github.com/Serendipity618/GLEAD/**.

Evaluation Metrics. We adopt precision, recall, F-1 score, and Area Under Receiver Operating Characteristic Curve (AUC) to evaluate the performance of abnormal sequence and entry detection. We run all experiments 10 times by randomly selecting 10 seeds and report the mean and standard deviation.

B. Experimental Results

TABLE II: Abnormal Sequence Detection

Dataset	Metric	iForest	OCSVM	LSTM-Attention	DeepSAD	GLEAD
BGL	Precision	31.09±6.63	24.68±13.35	75.00±28.52	**99.28**±0.45	97.86±2.67
	Recall	77.84±11.62	**94.44**±0.60	93.21±1.47	93.87±4.20	94.08±2.99
	F-1 score	44.30±8.52	37.44±16.91	79.52±22.64	**96.45**±2.26	95.91±2.39
	AUC	80.03±7.08	75.42±16.00	92.87±6.12	96.90±2.10	**96.93**±1.55
Thunderbird	Precision	12.69±6.22	8.79±0.08	53.16±19.11	**97.31**±2.47	96.96±1.01
	Recall	36.59±18.5	76.58±1.35	82.46±23.58	91.81±5.92	**97.83**±2.27
	F-1 score	18.84±9.3	15.77±0.15	63.17±20.02	94.36±3.17	**97.38**±1.11
	AUC	55.95±9.58	48.56±0.40	87.15±11.91	95.77±2.92	**98.76**±1.12

The Performance of Abnormal Sequence Detection. Table II shows the performance of detecting abnormal sequences on both datasets. First, GLEAD achieves very good performance for abnormal sequence detection with a high F-1 score and AUC. Second, the traditional one-class anomaly detection models (iForest and OCSVM) cannot achieve reasonable performance. This could be because using a count vector to represent a sequence loses too much information. Third, LSTM-Attention has much better performance compared with iForest and OCSVM by using a deep learning model to capture sequential information. However, as a supervised model, due to the limited labeled samples, the performance of LSTM-Attention is still worse than GLEAD, let alone its lack of global explainability. Fourth, DeepSAD, as a semi-supervised model, which also uses both labeled and unlabeled datasets, achieves comparable performance with our approach, especially on the BGL dataset. While on Thunderbird, our approach is still better than DeepSAD with a large margin. Meanwhile, compared with DeepSAD, GLEAD can further achieve local and global explanations of detection results.

TABLE III: Abnormal Entry Detection

Dataset	Metric	iForest	OCSVM	LSTM-Attention	Shapley	GLEAD
BGL	Precision	18.39±6.48	21.66±0.55	75.50±27.85	**98.24**±1.13	97.62±2.96
	Recall	75.33±28.33	**99.19**±0.57	76.89±2.10	89.87±4.66	96.71±1.90
	F-1 score	29.55±10.52	35.56±0.74	72.98±19.48	93.79±2.26	**97.14**±1.96
	AUC	71.36±14.16	81.65±0.61	85.72±5.29	94.86±2.29	**98.25**±0.99
Thunderbird	Precision	79.87±42.09	16.21±1.42	10.46±6.14	37.92±12.04	**98.06**±1.20
	Recall	79.92±42.12	**100.00**±0.02	26.26±19.13	83.92±16.32	98.71±1.32
	F-1 score	79.89±42.11	27.88±2.09	14.42±9.16	51.62±12.87	**98.37**±0.54
	AUC	89.90±21.17	73.96±2.61	61.77±9.37	91.09±8.18	**99.34**±0.66

The Performance of Abnormal Entry Detection (Local Explanation). Table III shows the performance of detecting abnormal entries on the test sets. In short, by leveraging the attention weights for abnormal entry detection, GLEAD achieves very good performance. By using log messages for abnormal entry detection, iForest and OCSVM can detect some abnormal entries with explicit abnormal words, but cannot detect the abnormal entries without explicit error messages. LSTM-Attention also does not achieve good performance due to the limited labeled samples to train the attention layer. As a result, it is difficult to obtain meaningful attention weights from the attention layer for abnormal entry detection. Shapley as a post-hoc explanation approach gets good performance among baselines but is still worse than our approach. By leveraging multi-head self-attention, GLEAD can capture the patterns in abnormal sequences even though the anomalies are diverse and sparse.

Capturing Abnormal Patterns (Global Explanation). GLEAD creates a top entry list for each attention head by counting the abnormal entries in the abnormal sequences

TABLE IV: Anomaly Types in Test Sets of BGL and Thunderbird

Dataset	Representative abnormal entries for each abnormal type
BGL	**H/KERNDTLB**: data TLB error interrupt **H/KERNSTOR**: data storage interrupt **S/APPREAD**$^{\#}$: ciod: failed to read message prefix on control stream **S/KERNRTSP**$^{\#}$: rts panic! - stopping execution
Thunderbird	**KERNEL_IB**: Fatal error (Local Catastrophic Error)

TABLE V: Most Frequent Entries Related to Each Prototype Vector

Prototype	BGL	Thunderbird
$\mathbf{n}_1$	S/APPREAD$^{\#}$, S/KERNRTSP$^{\#}$	KERNEL_IB
$\mathbf{n}_2$	*	*
$\mathbf{n}_3$	H/KERNSTOR	*
$\mathbf{n}_4$	*	*
$\mathbf{n}_5$	H/KERNDTLB	*

captured by the attention head. Since raw entries are encoded in the data preprocessing period, here we only show the corresponding abnormal content of the top entry of each head.

Table IV shows the ground truth of representative abnormal entries for each abnormal type in test sets of BGL and Thunderbird. There are four abnormal patterns on BGL and only one abnormal pattern on Thunderbird. Meanwhile, on both datasets, each pattern can be represented by one abnormal entry. Note that on BGL, only two abnormal patterns, H/KERNSTOR and H/KERNDTLB, are available in the training set, while the other two abnormal patterns, S/APPREAD and S/KERNRTSP, are only observed in the test set. The purpose of global explanation is to identify these representative abnormal entries for all abnormal patterns.

Table V shows the most frequent entries in the abnormal sequences captured by each vector in the abnormal prototype matrix $\mathbf{N}$. The special symbol "*" indicates that the prototype vector does not capture any abnormal patterns, i.e., no abnormal sequences are assigned to the vector. On BGL, there are two types of abnormal patterns in the training set, H/KERNSTOR and H/KERNDTLB. Then, GLEAD can successfully capture both abnormal patterns by two prototype vectors ($\mathbf{n}_3$ and $\mathbf{n}_5$). Meanwhile, for the new abnormal patterns, S/APPREAD and S/KERNRTSP, GLEAD can still detect the abnormal sequences with unknown types of anomalies by one prototype vector ($\mathbf{n}_1$). However, because the model does not observe them in the training, these two types of abnormal patterns are assigned to one head. On Thunderbird, there is only one abnormal pattern in the test set. The prototype vector $\mathbf{n}_1$ captures this abnormal pattern, which is caused by KERNEL_IB, while the top entry list related to other prototype vectors is empty.

Therefore, from Table V, we notice that each prototype encodes one type of abnormal pattern. It also works for new anomaly types that are unseen in the training set. Meanwhile, if the number of abnormal patterns is less than the number of pre-defined prototypes, some heads remain empty as the prototypes do not have any sequences close to them.

V. Conclusions

In this paper, we have developed GLEAD, a sequential anomaly detection model that can achieve local and global explanations in a semi-supervised setting. GLEAD leverages the multi-head self-attention technique to capture different aspects of information of sequences. The local explanation, which aims to detect abnormal entries in the abnormal sequences, can be achieved based on the attention weights in the attention model. Meanwhile, two prototype matrices are developed to encode the global patterns of normal and abnormal sequences. By decoding the prototype representations by using abnormal entries, we can explain why a group of sequences is detected as abnormal. Experiments on two datasets show that our approach can achieve high accuracy on abnormal sequence detection, and the attention weights can provide precise local explanations for each individual sample. The identified abnormal entries represent the global patterns of abnormal sequences. A potential future direction is to extend our approach to a one-class setting, which can provide local and global explanations for anomaly detection results by only using normal samples.

Acknowledgment

This work was supported in part by NSF 2103829.

References

[1] W. Meng, Y. Liu, Y. Zhu, S. Zhang, D. Pei, Y. Liu, Y. Chen, R. Zhang, S. Tao, P. Sun, and R. Zhou, "Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs," in *IJCAI*, 2019.

[2] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," in *CCS*, 2017.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[4] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," in *ICLR*, 2017.

[5] L. Ruff, R. A. Vandermeulen, N. Görnitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *ICML*, 2018.

[6] L. Ruff, Y. Zemlyanskiy, R. Vandermeulen, T. Schnake, and M. Kloft, "Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text," in *ACL*, 2019.

[7] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft, "Deep semi-supervised anomaly detection," in *International Conference on Learning Representations*, 2020.

[8] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *KDD*, 2016.

[9] R. Marco Tulio, S. Sameer, and G. Carlos, "Anchors: High-precision model-agnostic explanations," in *AAAI*, 2018.

[10] Y. Ming, P. Xu, H. Qu, and L. Ren, "Interpretable and steerable sequence learning via prototypes," *KDD*, 2019.

[11] C. Chen, O. Li, C. Tao, A. J. Barnett, J. Su, and C. Rudin, "This looks like that: Deep learning for interpretable image recognition," 2019.

[12] N. Puri, P. Gupta, P. Agarwal, S. Verma, and B. Krishnamurthy, "Magix: Model agnostic globally interpretable explanations," 2017.

[13] N. Frosst and G. Hinton, "Distilling a neural network into a soft decision tree," 2017.

[14] O. Bastani, C. Kim, and H. Bastani, "Interpreting blackbox models via model extraction," 2017.

[15] 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)*, 2007.

[16] 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)*, 2017, pp. 33–40.

[17] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 413–422.

[18] 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, 07 2001.