Explainable Sequential Anomaly Detection via Prototypes

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

2023 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-8867-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/IJCNN54540.2023.10191703

***Abstract*—Sequential anomaly detection has received more and more attention because of its wide applications in various domains, such as debugging system failures via logs. Researchers have recently proposed many deep learning-based approaches for sequential anomaly detection. However, these approaches work as black-boxed models, not providing explanations for detected anomalies. On the other hand, explainability is a critical requirement to build trustworthiness in detection results. Moreover, domain experts would like to learn why a sequence is labeled as an anomaly. To overcome this challenge, in this paper, we propose a framework for Explainable Sequential Anomaly Detection (ESAD) in a semi-supervised setting. As there are various normal and abnormal behaviors in sequential data, ESAD derives multiple prototypes to describe diverse normal and abnormal sequences. Each prototype can encode one type of normal or abnormal behavior. Given a new sequence, if the sequence is similar to an abnormal prototype, the sequence will be detected as abnormal. After decoding the abnormal prototype as a prototypical sequence, domain experts can further understand the newly detected abnormal sequence by examining the prototypical sequence. We conduct experiments on one log dataset and two text datasets. Experimental results including quantitative and qualitative analysis on three datasets show the effectiveness of our model.**

***Index Terms*—sequential data, anomaly detection, explanations**

I. Introduction

Sequential anomaly detection has received much attention in recent years because of its wide applications. For example, detecting anomalies in system logs is an important task for building secure systems [1]–[5]. Identifying anomalies in the system logs can make a great contribution to debugging system errors or defending against attacks.

Recently, researchers have applied deep learning models for sequential anomaly detection, which outperform traditional approaches [6], [7]. However, deep learning models mostly are not transparent, which poses a barrier to widely applying them to anomaly detection problems. Predictions of the models reporting anomalies without explanations will not be well accepted by domain experts, especially for the high stake tasks. Therefore, it is an urgent task to explore interpretable sequential anomaly detection approaches. However, compared to the task of interpretable object detection in computer vision, where the interpretations can be the highlighted objects, providing human-understandable and helpful explanations for sequential anomaly detection is more challenging and yet not to be well explored.

In this paper, we aim to develop an interpretable anomaly detection model on sequential data via prototypes. Prototypes can provide intuitive explanations to model decisions, where a prototype is usually a representative case in an observed dataset [8]–[11]. Prototype-based explanations are analogous to how humans make decisions when a new case presents before them, i.e., comparing it to similar cases and then deriving a decision based on similar cases. In our scenario, for a sequential anomaly detection model, if the model can show that a newly detected anomaly is similar to some abnormal sequences in the observed dataset, then the domain expert can easily understand why a sequence is labeled as abnormal.

To this end, we develop a sequential anomaly detection model with explanations via prototypes. Because the explanations are provided by comparing the unlabeled samples to the observed (labeled) samples, anomaly detection is conducted under a semi-supervised setting, which assumes the availability of a small set of labeled samples and a large number of unlabeled samples [12].

Considering the diversity of normal and abnormal sequences, we aim to derive multiple prototypes, each of which can capture one type of normal or abnormal behavior. Therefore, we leverage contrastive learning and the k-means algorithm to detect abnormal sequences and provide explanations by showing the prototypes. Specifically, k-means is applied to cluster normal and abnormal sequences, respectively, where the sequence closest to the center of each cluster is the prototypical sequence of the cluster. Then, the sequences grouped into the same cluster can be explained by the corresponding prototypical sequence. On the other hand, in order to achieve meaningful clustering results, having distinguishable representations of sequences is critical. In this work, we leverage the idea of contrastive learning to derive sequence representations. In particular, contrastive learning in our approach consists of two parts. One is instance-wise contrastive learning, which is to learn a good representation of each sequence. Another is cluster-wise contrastive learning, which is to learn separable representations for each type of sequences so that the sequences with similar patterns would be grouped together while sequences with different patterns would be separated. Finally, to properly train the whole framework, the k-means

algorithm and contrastive learning are conducted in an iterative manner. The intuition is that after each iteration, k-means can get better clusters, and meanwhile, contrastive learning, especially cluster-wise contrastive learning, derives the positive and negative samples based on the clustering results. Then better clustering results can further help contrastive learning to derive more distinguishable sequence representations.

We summarize the contributions of ESAD as follows. First, ESAD leverages both instance-wise and cluster-wise contrastive learning loss to derive the sequence representations so that similar sequences can be grouped together. Second, by using the k-means algorithm to find the underlying clusters of normal and abnormal sequences, ESAD can leverage the prototypical sequence of each cluster to provide instance-based explanations. Third, the experimental results show that ESAD can accurately detect anomalies and provide insightful explanations.

II. Related Work

Interpretable Machine Learning. Interpretability is becoming a prominent desideratum of trustworthy machine learning, especially when people deploy machine learning models into high-stakes applications. Currently, most interpretable machine learning techniques can be grouped into two categories, inherent interpretability, and post-hoc interpretability. Inherently interpretable models are designed to justify their decisions based on their structures. For example, FCDD [13] is an inherently interpretable framework that can detect abnormal images and yield explainable heat maps. ProSeNet [9] is another inherently interpretable approach by learning prototypes for each class in classification tasks. To interpret the black-box machine learning models, post-hoc techniques are developed to provide explanations, such as perturbation-based and gradient-based approaches. LIME [14] and SHAP [15] are both perturbation-based approaches to identify important features based on the impact on the outputs given perturbated inputs. Gradient-based approaches evaluate the contribution of input features according to the gradient magnitude, such as Grad-CAM [16] and Integrated Gradient [17]. In general, a model with inherent interpretability can better show the internal behavior of the specific model but not be able to explain other models, while many post-hoc interpretability techniques are model agnostic and can be applied to explain multiple black-boxed models. One limitation of the existing interpretable machine learning models is that they are usually designed in a supervised learning setting, which is impractical for many anomaly detection tasks. For example, although ProSeNet can show prototypes of sequential data for case reasoning, it is trained by labeled samples for classification. Once we have the supervised signals, the number of classes usually means the number of prototypes we expect for explanations. On the other hand, for the anomaly detection tasks, in most cases, we may have a few anomalies available, but usually do not have information about the types of anomalies in advance. In such a case, how to derive prototypes for case reasoning is much more challenging compared with the task of supervised prototype learning.

Contrastive Learning. Contrastive learning is widely used for unsupervised representation learning, aiming to pull similar instances close and push different instances apart [18]–[23]. Recently, contrastive learning is also applied to learn the representation of sequential data, especially text data [20], [24]–[26]. For example, an efficient framework to learn sentence representations [20] is proposed by taking the contextual sentences as positive samples. Meanwhile, several approaches are also developed to improve the efficiency of contrastive learning. The memory bank approach [23] is proposed to learn discriminative individual instance representations by storing the features of all samples that are derived in the previous step. MoCo [19] employs a momentum encoder to obtain positive and negative samples and maintains a queue to keep data instance features. In this work, we propose a contrastive learning framework that considers the discriminative of samples at both the instance-level and cluster-level so that samples shared similar patterns can group together.

III. Methodology

A. Overview

In this paper, we aim to detect abnormal sequences and further identify prototypes for case reasoning as explanations. Specifically, we consider a semi-supervised setting, where a small set of normal and abnormal sequences as well as a large number of unlabeled samples are available.

We develop an Explainable Sequential Anomaly Detection (ESAD) approach to detect abnormal sequences in a semi-supervised setting. ESAD detects anomalies based on the prototypes of normal and abnormal sequences, where each prototype represents a specific pattern of normal or abnormal sequences. In particular, ESAD employs an encoder to encode sequences into an embedding space and trains the encoder with the contrastive loss. The encoder can learn sequence representations that make each sequence only close to its similar samples and far from other sequences. We separately perform a clustering algorithm, i.e., k-means, on the normal and abnormal sequence representations to obtain normal and abnormal clusters. We then derive the centroid of each cluster as the prototype and mark a sequence that is closest to the centroid as the prototypical sequence. Therefore, each prototype represents a specific normal or abnormal pattern, and the prototypical sequence can be used to understand why a new sequence is labeled as abnormal via case reasoning. Figure 1 illustrates the ESAD framework.

During the inference process, given a test sequence, we can search for the most similar prototype in the embedding space. The testing sequence will be predicted as abnormal if the prototype indicates an abnormal pattern. Furthermore, by examining the corresponding prototypical sequence of the test sample, the domain expert can understand why a sequence is labeled as abnormal.

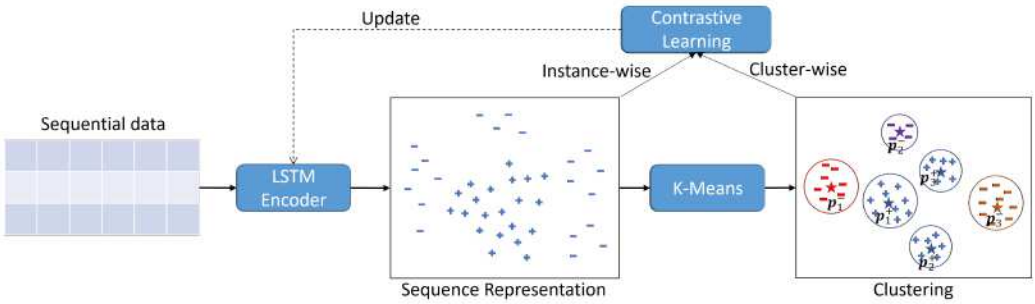


Fig. 1: Illustration of the ESAD Framework.

B. Sequence and Prototype Representation

Assume that we are given a small set of labeled sequences $\mathcal{S}$ consisting of a subset of normal sequences $\mathcal{S}^+ = \{(s_i^+, y_i = 0)\}_{i=1}^{|\mathcal{S}^+|}$ and a subset of abnormal sequences $\mathcal{S}^- = \{(s_i^-, y_i = 1)\}_{i=1}^{|\mathcal{S}^-|}$, and also given a large set of unlabeled sequence $\mathcal{U} = \{s_i^u\}_{i=1}^{|\mathcal{U}|}$. Following the typical assumption, we assume most of the sequences in $\mathcal{U}$ are normal samples [12].

Sequence Representation. We adopt a long short-term memory (LSTM) neural network [27] as an encoder to encode a sequence s_i into its representation, defined as $\mathbf{r}_i = f(s_i)$, where $f(\cdot)$ denotes the LSTM model and $\mathbf{r}_i$ is derived by the average of all hidden states in LSTM over the sequence.

Prototype Representation. We further aim to derive the prototype representations of both normal and abnormal sequences. In this work, we leverage the k-means algorithm to group sequences with a similar pattern together so that the centroid of the cluster can be naturally considered as the prototype of the group of sequences. To this end, given sequences $s_i^+ \in \mathcal{S}^+ \cup \mathbf{U}$, we first derive the sequence representations $\mathcal{R}^+ = \{\mathbf{r}_1^+, \mathbf{r}_2^+, \ldots, \mathbf{r}_{|\mathcal{S}^+|+|\mathcal{U}|}^+\}$. Note that as the majority of unlabeled sequences are normal, we combine the normal and unlabeled sequences together. Then, we run k-means on $\mathcal{R}^+$ and derive the prototypes $\mathcal{P}^+$ for the normal sequences, i.e.

$$\mathbf{p}_1^+, \mathbf{p}_2^+, \ldots, \mathbf{p}_{k^+}^+ = k\text{-means}(\mathcal{R}^+), \tag{1}$$

where k^+ is the number of normal clusters, and $\mathbf{p}_1^+, \mathbf{p}_2^+, \ldots, \mathbf{p}_{k^+}^+ \in \mathcal{P}^+$ are the centroids of normal clusters.

Similarly, for any abnormal sequence $s_i^- \in \mathcal{S}^-$, we use the same LSTM model to obtain its representation as $\mathbf{r}_i^- = f(s_i^-)$ and apply k-means on representations $\mathcal{R}^- = \{\mathbf{r}_1^-, \mathbf{r}_2^-, \ldots, \mathbf{r}_{|\mathcal{S}^-|}^-\}$ of sequences in $\mathcal{S}^-$ getting the prototypes $\mathcal{P}^-$ for the abnormal sequences:

$$\mathbf{p}_1^-, \mathbf{p}_2^-, \ldots, \mathbf{p}_{k^-}^- = k\text{-means}(\mathcal{R}^-), \tag{2}$$

where k^- is the number of abnormal clusters, and $\mathbf{p}_1^-, \mathbf{p}_2^-, \ldots, \mathbf{p}_{k^-}^- \in \mathcal{P}^-$ are the centroids of abnormal clusters.

C. Learning Objective

For any sequence $s_i \in \mathcal{S} \cup \mathcal{U}$, the LSTM encoder can map s_i into an embedding space where sequences with similar patterns are grouped together. Then, k-means can find clusters of sequences, and the centroid of a cluster can be a prototype of a group of sequences. To this end, we adopt contrastive learning to train the LSTM encoder, aiming to pull similar sequences close while pushing different sequences apart. Specifically, the objective function consists of two parts, instance-wise and cluster-wise contrastive learning.

Instance-wise Contrastive Learning. The instance-wise contrastive learning is similar to the existing contrastive learning approach [21], where the positive pair is from similar sequences while the negative pairs are from the discrepant sequences, i.e.

$$\mathcal{L}_{con} = \sum_{i=1}^{|\mathcal{S}|+|\mathcal{U}|} -\log \frac{exp(f(s_i) \cdot f(s_i')/\tau)}{\sum_{j=0}^{r} exp(f(s_i) \cdot f(s_j')/\tau)}, \tag{3}$$

where τ is a temperature hyper-parameter, and (s_i, s_i') are a positive pair, and s_j' includes one positive sample (i.e., when $j = i$) and r negative samples.

Contrastive learning needs positive and negative samples for the pretext task. The negative samples can be easily composed by sampling discrepant sequences from the training set. To generate positive samples, we develop a token replacement strategy for data augmentation. *Token replacement* is to replace the tokens in a sequence with similar tokens. To this end, we first generate normal and abnormal token dictionaries, respectively, based on the training set. Each dictionary consists of tokens from the corresponding labeled sequences. Then, for each sequence s_i, we randomly replace a few tokens to generate augmented samples.

Cluster-wise Contrastive Learning. In order to derive the prototypes, we also aim to learn various patterns underlying the normal and abnormal sequences. In general, the sequences that share a similar pattern should group together, meaning that similar sequences should be close to the corresponding prototype. On the other hand, sequences with different patterns should be apart, meaning that sequences should be far away from the prototypes representing different patterns. Therefore, for any prototype $\mathbf{p}_c \in \mathcal{P} = \mathcal{P}^+ \cup \mathcal{P}^-$ of a cluster c, $\mathbf{p}_c$ can be used for composing the positive sample of a sequence s_i^c, where s_i^c indicates a sequence in the cluster c. On the other hand, the prototypes except $\mathbf{p}_c$ can be used to compose the

negative samples of s_i^c. Therefore, the objective function can be defined as:

$$\mathcal{L}_{pro} = \sum_{i=1}^{|\mathcal{S}|+|\mathcal{U}|} -\log \frac{\exp(f(s_i^c)\cdot \mathbf{p}_c/\phi_c)}{\sum_{j=1}^{|\mathcal{P}|}\exp(f(s_i^c)\cdot \mathbf{p}_j/\phi_j)}, \quad (4)$$

where s_i^c is a sequence in cluster c and ϕ_c and ϕ_j indicates the concentration level of the clusters c and j, respectively. The concentration level of a cluster c can be proportional to the summation of Euclidean distance between $f(s_i^c)$ and $\mathbf{p}_c$ for all sequences in the cluster c, a small concentration level indicating a tight cluster.

Finally, to learn meaningful sequence representations and prototypes, we combine Equations 3 and 4 as the final objective function:

$$\mathcal{L} = \mathcal{L}_{con} + \lambda \cdot \mathcal{L}_{pro}, \quad (5)$$

where λ is a hyper-parameter to balance these two objectives.

D. Abnormal Sequence Detection via Prototypes

A new sequence can be labeled based on its closest prototype. Specifically, given a test sequence s_t, we search for the most similar prototype in $\mathcal{P}$ as:

$$\mathbf{p}^* = \arg\min_{\mathbf{p}\in\mathcal{P}} D(f(s_t), \mathbf{p}), \quad (6)$$

where $D(\cdot)$ indicates the Euclidean distance. If $\mathbf{p}^* \in \mathcal{P}^+$, then we predict s_t as normal, while if $\mathbf{p}^* \in \mathcal{P}^-$, we label s_t as abnormal. Meanwhile, the sequence in the training set that is closest to the corresponding $\mathbf{p}^*$ is a concrete instance of the prototype of the cluster, which can provide the explanation to domain experts for case reasoning.

E. Training Details

We briefly discuss how to train our framework.

Iterative Training. Our framework consists of two parts, k-means and contrastive learning. During the training phase, these two steps are run in an iterative manner. In the beginning, the LSTM encoder can not learn separable representations for sequences due to randomly initialized parameters. Therefore, the clustering results from k-means may not be well separated. In this stage, the objective of instance-wise contrastive learning plays a vital role in learning good representations of sequences. As in each iteration, we update the LSTM encoder, the clustering results from k-means would become better. Then, the objective for cluster-wise contrastive learning can further group sequences into different clusters, and the centroid of each cluster becomes a representative point of a cluster, i.e., the prototype. After training, the LSTM encoder should be able to map similar sequences into a cluster.

Momentum Encoder. As in contrastive learning, using the same encoder to encode sequences and their negative samples in different batches may cause an inconsistent problem, we adopt the idea of MoCo [19] to implement our contrastive learning framework, which includes an original encoder and a momentum encoder. Specifically, the representation of the original sequence (query sequence) $f(s_i)$ is derived by the original LSTM encoder, while the representations of the positive or negative samples (key sequences) $f(s_i')$ and $f(s_j')$ are derived by the momentum encoder. The relationship between the original encoder and momentum encoder in terms of their parameters can be defined as $\theta_m \leftarrow \alpha\theta_m + (1-\alpha)\theta_e$, where θ_e and θ_m are the parameters of the original and momentum LSTM encoders, respectively, and $m \in [0, 1)$ is a momentum coefficient. Meanwhile, similar to [28], the sequence representations used for k-means clustering are derived from the momentum encoder.

IV. Experiments

A. Datasets

We evaluate ESAD against existing baselines on three datasets:

TABLE I: Statistics of the training and testing sets in three datasets

Dataset		BGL	Reuters	20 Newsgroups
Training	Normal	100	200	200
	Abnormal	100	200	200
	Unlabeled	900	800	800
Test	Normal	10000	1000	1000
	Abnormal	1000	200	200

- **BlueGene/L (BGL)** [29] is a log dataset containing alert and non-alert log messages collected from a BlueGene/L supercomputer system at Lawrence Livermore National Labs.
- **Reuters** is a text dataset consisting of 90 categories from 7769 training documents and 3019 testing documents.
- **20 Newsgroups** is another text dataset containing about 18000 newsgroups posts from 20 different topics.

For BGL, we consider the sequences with alert messages as anomalies, while for the Reuters and 20 Newsgroups, we select a set of categories as normal classes and another set of categories as abnormal classes. In particular, for the Reuters dataset, we set "earn" and "acq" as the normal classes, while "interest", "wheat", "dlr", "gnp", and "crude" as abnormal classes. For the 20 Newsgroups dataset, we consider that "talk.politics.mideast", "talk.politics.misc", and "talk.religion.misc" are normal classes, while "misc.forsale", "rec.sport.baseball", "sci.med", "soc.religion.christian", and "rec.autos" are abnormal classes.

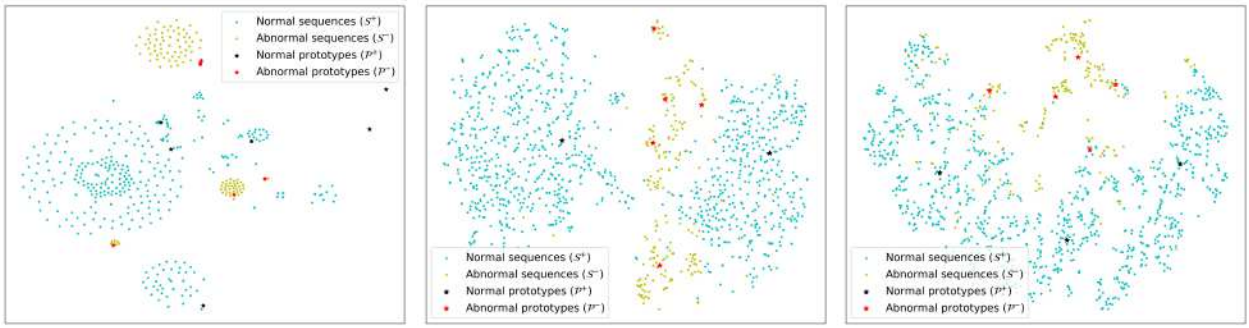
Table I shows the statistics of the three datasets in our experiments. For the BGL dataset, we use 100 normal, 100 abnormal, and 900 unlabeled sequences for training, while the test set includes 10000 normal and 1000 abnormal sequences. For the Reuters and 20 Newsgroups datasets, the training set consists of 200 normal and 200 abnormal documents as well as 800 unlabeled documents, while the test set consists of 1000 normal documents and 200 abnormal documents.

B. Baselines

We compare our approach with the following unsupervised and semi-supervised baselines for detecting abnormal sequences.

TABLE II: Results of abnormal sequence detection (mean ± std.)

Dataset	Metric	iForest	OCSVM	DeepSAD	ESAD
BGL	Precision	21.92±5.74	20.27±10.20	85.43±11.81	**90.88**±4.81
	Recall	49.65±13.94	60.05±18.50	97.46±2.52	**97.49**±2.05
	F-1 score	30.40±8.13	29.95±13.36	90.63±7.13	**93.99**±2.51
	AUC	57.28±7.67	53.47±16.37	97.79±1.54	**98.24**±0.96
Reuters	Precision	41.80±2.61	42.67±1.91	76.86±8.14	**86.03**±3.28
	Recall	58.15±4.49	41.15±3.58	88.90±12.80	**94.85**±1.16
	F-1 score	48.59±2.97	41.85±2.55	82.25±9.67	**90.19**±1.89
	AUC	70.97±2.21	65.05±1.60	91.79±6.79	**95.87**±0.70
20 Newsgroups	Precision	27.40±1.65	26.50±2.21	**72.92**±13.98	71.53±10.29
	Recall	41.30±3.03	35.60±3.05	85.35±11.18	**87.10**±5.63
	F-1 score	32.91±1.90	30.38±2.55	77.19±7.20	**77.82**±5.26
	AUC	59.69±1.42	57.93±1.79	88.91±4.11	**89.71**±1.65



(a) BGL (b) Reuters (c) 20 Newsgroups

Fig. 2: Visualization of the normal sequences (cyan), abnormal sequences (yellow), normal prototypes (black), and abnormal prototypes (red)

- **Isolation Forest (iForest)** is an unsupervised outlier detection algorithm that recursively partitions the observations based on decision trees [30].
- **One Class Support Vector Machine (OCSVM)** is a one-class novelty detection method that learns the pattern of observations and recognizes samples whose behaviors deviate from the learned pattern [31].
- **DeepSAD** is a state-of-the-art semi-supervised anomaly detection approach that takes advantage of the labeled anomalies as well as the unlabeled samples to achieve good performance on anomaly detection. [32].

For iForest and OCSVM, we use count vectors to represent sequences by transferring training text data to a matrix of token counts. For DeepSAD, we use an LSTM to encode sequences to representations and update the parameters of the encoder during the training period.

C. Implementation Details

For the BGL dataset, we first apply a log parser, Drain [33], to transfer raw log messages to log templates. We then apply a sliding window with size 20 to generate log sequences. For the Reuters and 20 Newsgroups dataset, we preprocess the raw text data by removing the stopwords, stemming, and tokenizing, as well as splitting them into training and test sets. We then employ Word2Vec to represent words and update the vectors during the training period.

In BGL, we represent log templates as 100-dimensional embedding vectors. After randomly initializing the structured log template embeddings, we adopt an LSTM as the encoder and generate the sequence representation with 256 dimensions. We also employ Adam as the optimizer with a learning rate of 0.005 and weight decay of 1e-6. We trained our model for 150 epochs with a batch size of 20. We set the temperature hyper-parameter τ in Equation 3 as 0.05. To balance the loss terms $\mathcal{L}_{con}$ and $\mathcal{L}_{pro}$, we set the hyper-parameter λ in Equation 5 as 1.0. The k in the k-means algorithm is set as the number of normal or abnormal classes in each dataset. The source code is available online [1].

D. Experimental Results

1) Sequential Anomaly Detection: Table II shows the performance of abnormal sequence detection on three datasets. We run the experiments ten times and report the mean and standard deviation. First, the traditional anomaly detection approaches, iForest and OCSVM, cannot achieve reasonable performance on sequential anomaly detection as using count vectors to represent sequences cannot well capture the sequential patterns. Second, DeepSAD as a state-of-the-art semi-supervised anomaly detection method can achieve good performance, which shows the advantage of leveraging a few labeled samples for training. However, in terms of F-1 score and AUC, ESAD is also better than DeepSAD. Overall, ESAD achieves the best performance with the highest F-1 score and AUC compared to the baselines. Meanwhile, ESAD can

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

TABLE III: Results of sequence anomaly detection and clustering with various pre-defined k^-

Dataset	Metric		$k^- = 1$	$k^- = 2$	$k^- = 3$	$k^- = 4$	$k^- = 5$	$k^- = 6$	$k^- = 7$
Reuters	sequence anomaly detection	Precision	94.79	92.68	91.90	90.87	87.84	90.52	90.82
		Recall	91.00	95.00	96.50	94.50	97.50	95.50	94.00
		F-1 score	92.86	93.83	94.15	92.65	92.42	92.94	92.38
	clustering	Rand Index	0.87	0.88	0.88	0.88	0.87	0.88	0.87
		Mutual Information	0.73	0.84	0.83	0.84	0.84	0.85	0.85
20 Newsgroups	sequence anomaly detection	Precision	83.87	83.78	82.81	85.25	79.81	75.44	70.17
		Recall	78.00	77.50	79.50	78.00	83.00	86.00	83.50
		F-1 score	80.83	80.52	81.12	81.46	81.37	80.37	76.26
	clustering	Rand Index	0.69	0.69	0.69	0.69	0.71	0.71	0.71
		Mutual Information	0.39	0.39	0.40	0.42	0.44	0.48	0.48

further group abnormal samples into multiple clusters and provide explanations via prototypical sequences.

2) Visualization: We further visualize normal and abnormal sequence representations in BGL, Reuters, and 20 Newsgroups. We select 1000 normal and 200 abnormal sequences and feed them to the momentum encoder to obtain sequence representations. We then employ t-SNE [34] to map sequence representations into a two-dimensional space. Figure 2 shows the visualization results. First, we can observe that normal and abnormal sequences are overall separated. Second, both normal and abnormal sequences can be grouped into multiple clusters, where each cluster consists of sequences with similar patterns. Meanwhile, as the anomalies are diverse, we can notice that the distribution of abnormal clusters is more diverse compared with normal clusters. Furthermore, we can notice that the prototypes of all normal and abnormal sequences are far from each other, which indicates that by using contrastive learning, different patterns of normal and abnormal sequences are separated.

3) Clustering Analysis: Because the Reuters and 20 Newsgroups datasets provide ground truth information about the class of each text, we further conduct clustering analysis to check the impact of pre-defined numbers of abnormal prototypes on the performance of sequence anomaly detection and clustering. Specifically, we vary the value of k^- and evaluate the performance of sequence anomaly detection using precision, recall, and F-1 score, as well as the performance of detecting various anomalies using Rand Index and Mutual Information [35]. The ground-truth k^- for both Reuters and 20 Newsgroups datasets is 5. We evaluate the performance of our approach by changing the value k^- from 1 to 7. The results are shown in Table III. First, in terms of clustering, on both datasets, most of the texts in the same class are grouped into one cluster, leading to a high rand index and mutual information score as long as we set a reasonable k^- value. Meanwhile, our findings suggest that clustering results may be unsatisfactory for much smaller values of k^-, as a reduced number of abnormal prototypes may fail to effectively summarize multiple anomalies. In terms of anomaly detection, increasing the value of k^- does not significantly affect the accuracy of sequence anomaly detection on Reuters, but can lead to a decrease in accuracy on 20 Newsgroups, particularly for values of k^- exceeding the golden number.

E. Case study

One major advantage of ESAD is that the derived prototypes can provide human-understandable explanations for case reasoning after they are instantiated by real-world sequences. We further conduct a case study for each dataset to show the advantage. Since in practice, we are usually more interested in why a sequence is labeled as abnormal, in this case study, we focus on showing how to explain the detected abnormal sequence based on the prototypical sequence.

1) Case Study 1: Explainable Text Data Anomaly Detection: Table IV shows the test cases of anomaly detection on Reuters and 20 Newsgroups. In the Reuters dataset, given a query text that is detected as abnormal due to its closeness to an anomaly prototype, we can figure out the reason why this text is abnormal by checking the corresponding prototypical sequence. For example, the prototypical sequence of Query I is about the bank providing help to the market. Once the domain experts know the prototype, they can figure out Query I should describe a similar topic. This is confirmed when we further check the text of Query I. Both Query I and the prototypical sequence are from the category "interest" in the Reuters dataset. Similarly, the prototypical sequence of Query II is about wheat. With that information in mind, once a new text, such as Query II, is detected, the domain expert can know Query II should have a similar topic. Hence, ESAD detects abnormal sequences via prototypes, i.e., when new sequences have similar abnormal patterns to the prototypes, ESAD can capture them.

We have similar observations in the 20 Newsgroups dataset. Given a text, Query I, ESAD detects the text as abnormal. Then, by checking the corresponding prototypical sequence, which is about the topic "autos", we can be safely sure that Query I is detected as abnormal due to describing a similar topic. If we further examine Query I, both Query I and the prototypical sequence are about "autos". Moreover, given a text, Query II, the prototypical sequence can explain why Query II is labeled as an anomaly.

2) Case Study 2: Detecting and Explaining Anomalies in Log Data: Log data record the system or user activities so we can use them to debug system faults or identify malicious attacks. Generally, there are multiple abnormal patterns in log data, especially for complicated systems, so we further evaluate ESAD on the BGL dataset which is a good representative

TABLE IV: Case study on text data

Reuters	
Query I	U.K. MONEY MARKET GIVEN FURTHER 166 MLN STG HELP The Bank of England said it provided the market with further help totalling 166 mln stg ...
Prototype	U.K. MONEY MARKET RECEIVES 205 MLN STG LATE HELP The Bank of England said it has provided around 205 mln stg late assistance to the market ...
Query II	SRI LANKA GETS USDA APPROVAL FOR WHEAT PRICE Food Department officials said the U.S. Department of Agriculture approved the Continental Grain Co sale of 52,500 tonnes of soft wheat at 89 U.S. Dlrs a tonne C and F from Pacific Northwest to Colombo ...
Prototype	CHINA BUYS U.S. HARD AND SOFT WHEAT Private exporters said China bought a total of 550,000 tonnes of U.S. wheat under the export enhancement program U.S. Department of Agriculture of the subsidies still awaited ...
20 Newsgroups	
Query I	They beat Ford to the market with the Camaro/Firebird, but really only in words. Production of these vehicles will be limited until the end of the year, keeping selling prices above MSRP for the most part since there are so many twitching Camaro fans out there ...
Prototype	On BOTH cars, the rubber seals around the window and door fell off. ... The panel gaps were large and non-uniform between the 2 cars I saw - the kind of thing you expect and accept on a Mustang, but not from Chrysler's savior ...
Query II	Public revelation, which is the basis of Catholic doctrine, ended with the death of St John, the last Apostle. Nothing new can be added. Every so often, the Pope declares that some departed Christian is now in Heaven, and may be invoked in the public rites of the Church ...
Prototype	If it were a sin to violate Sunday no one could ever be forgiven for that for Jesus never kept Sunday holy. He only recognized one day of the seven as holy. Jesus also recognized other holy days, like the Passover. Acts 15 says that no more should be layed on the Gentiles than ...

TABLE V: Top abnormal log template in each anomaly cluster

Cluster	Top abnormal log pattern and its frequency
1	KERNMC, 2
2	KERNDTLB, 6
3	KERNSTOR, 449
4	APPREAD, 40
5	KERNDTLB, 1380

of log data.

After running ESAD on the test set, we collect the detected abnormal sequences and calculate the frequency of abnormal log templates in the anomaly clusters. Table V shows the top abnormal log pattern in each anomaly cluster. We can notice that each cluster captures one dominant abnormal pattern. Because BGL includes some common abnormal patterns and several rare abnormal patterns [29], also shown in Figure 2a, we can notice some clusters have a lot of abnormal sequences while some clusters only have a few abnormal sequences. As we group the abnormal sequences into 5 clusters, we successfully identify 4 different abnormal patterns.

TABLE VI: Case study on log data

BGL	
Query	... 1840cbfe, 828a502b, 65f23e3e, 147cfcff, 6ede2c6c, 38a7307d, 38a7307d ...
Prototype	...38a7307d, 38a7307d, 38a7307d ...

We further conduct a case study to check whether we can get insightful explanations via prototypical sequences. Given a query sequence, we detect the sequence as abnormal because it is close to an anomaly prototype. By checking the prototypical sequence (shown in Table VI), we can notice that the prototypical sequence consists of a series of log templates "38a7307d", which indicates a specific abnormal pattern "KERNDTLB". Based on this information, the domain experts can expect the detected log sequence also has the same abnormal pattern "KERNDTLB". That is to say, we have already obtained the prototype which includes an abnormal pattern "KERNDTLB", so when a new sequence containing the same abnormal pattern comes, we can classify it as an abnormal sequence.

V. Conclusions

In this paper, we have developed a framework, ESAD, for identifying diverse anomalies in sequential data based on the idea of contrastive learning and clustering technique. ESAD is able to learn prototypes for normal and abnormal sequences, where each prototype represents a specific normal or abnormal pattern. Given a new sequence, ESAD predicts the new sequence based on the label of its closest prototype. Meanwhile, by instantiation of the prototype, ESAD can provide the prototypical sequence for the case reasoning. The prototypical sequence can provide insightful information for domain experts to understand why a sequence is labeled as abnormal. In the future, one potential direction for us is to explore whether we can apply the idea of prototype learning to explainable anomaly detection in an unsupervised setting.

Acknowledgement

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.

[2] X. Han, H. Cheng, D. Xu, and S. Yuan, "Interpretablesad: Interpretable anomaly detection in sequential log data," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021, pp. 1183–1192.

[3] 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. 19, no. 7, 2019, pp. 4739–4745.

[4] S. Yuan, P. Zheng, X. Wu, and H. Tong, "Few-shot insider threat detection," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 2289–2292.

[5] H. Guo, S. Yuan, and X. Wu, "Logbert: Log anomaly detection via bert," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.

[6] 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 amp; Data Mining*, ser. KDD '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1480–1490. [Online]. Available: https://doi.org/10.1145/3292500.3330932

[7] K. Doshi and Y. Yilmaz, "Any-shot sequential anomaly detection in surveillance videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[8] P. Hase, C. Chen, O. Li, and C. Rudin, "Interpretable Image Recognition with Hierarchical Prototypes," *arXiv:1906.10651 [cs, eess, stat]*, Aug. 2019.

[9] Y. Ming, P. Xu, H. Qu, and L. Ren, "Interpretable and steerable sequence learning via prototypes," in *KDD*, ser. KDD '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 903–913. [Online]. Available: https://doi.org/10.1145/3292500.3330908

[10] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su, "This looks like that: Deep learning for interpretable image recognition," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/adf7ee2dcf142b0e11888e72b43fcb75-Paper.pdf

[11] A. Das, C. Gupta, V. Kovatchev, M. Lease, and J. J. Li, "ProtoTEx: Explaining model decisions with prototype tensors," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 2986–2997. [Online]. Available: https://aclanthology.org/2022.acl-long.213

[12] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft, "Deep Semi-Supervised Anomaly Detection," in *ICLR*, Feb. 2020.

[13] P. Liznerski, L. Ruff, R. A. Vandermeulen, B. J. Franks, M. Kloft, and K.-R. Müller, "Explainable deep one-class classification," *arXiv preprint arXiv:2007.01760*, 2020.

[14] M. T. Ribeiro, S. Singh, and C. Guestrin, "" 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.

[15] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[16] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[17] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 3319–3328.

[18] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[19] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[20] L. Logeswaran and H. Lee, "An efficient framework for learning sentence representations," *arXiv preprint arXiv:1803.02893*, 2018.

[21] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[22] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *European conference on computer vision*. Springer, 2020, pp. 776–794.

[23] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3733–3742.

[24] J. Giorgi, O. Nitski, B. Wang, and G. Bader, "Declutr: Deep contrastive learning for unsupervised textual representations," *arXiv preprint arXiv:2006.03659*, 2020.

[25] M. Vinay, S. Yuan, and X. Wu, "Contrastive learning for insider threat detection," in *International Conference on Database Systems for Advanced Applications*. Springer, 2022, pp. 395–403.

[26] J. Chen, R. Zhang, Y. Mao, and J. Xu, "Contrastnet: A contrastive learning framework for few-shot text classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, 2022, pp. 10 492–10 500.

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

[28] J. Li, P. Zhou, C. Xiong, and S. Hoi, "Prototypical contrastive learning of unsupervised representations," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=KmykpuSrjcq

[29] 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, pp. 575–584.

[30] 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.

[31] 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.

[32] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft, "Deep semi-supervised anomaly detection," *arXiv preprint arXiv:1906.02694*, 2019.

[33] 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.

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

[35] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.