Interpretable Anomaly Detection in Event Sequences via Sequence Matching and Visual Comparison

Shunan Guo, Zhuochen Jin, Qing Chen, David Gotz, Hongyuan Zha, and Nan Cao

Abstract—Anomaly detection is a common analytical task that aims to identify rare cases that differ from the typical cases that make up the majority of a dataset. When analyzing event sequence data, the task of anomaly detection can be complex because the sequential and temporal nature of such data results in diverse definitions and flexible forms of anomalies. This, in turn, increases the difficulty in interpreting detected anomalies. In this paper, we propose a visual analytic approach for detecting anomalous sequences in an event sequence dataset via an unsupervised anomaly detection algorithm based on Variational AutoEncoders. We further compare the anomalous sequences with their reconstructions and with the normal sequences through a sequence matching algorithm to identify event anomalies. A visual analytics system is developed to support interactive exploration and interpretations of anomalies through novel visualization designs that facilitate the comparison between anomalous sequences and normal sequences. Finally, we quantitatively evaluate the performance of our anomaly detection algorithm, demonstrate the effectiveness of our system through case studies, and report feedback collected from study participants.

1 Introduction

Event sequence analysis plays an important role in many application domains due to its ubiquity and extensive data sources [1]. Anomaly detection is a common task for event sequence analysis as it often contributes to the discovery of critical and actionable information [2]. Effective use of anomaly detection in event sequence data requires identifying sequences that deviate from the typically event patterns [3]. For example, a doctor may be interested in finding patients whose postoperative response is different from other patients who have had the same surgery, so as to provide personalized care plans for similar patients in the future. However, in the real-world scenario, event sequence data are usually noisy and complex, large in scale, diverse in event type, vary in sequence length, and events may occur in different orders and last for different durations, thus making the anomaly detection task especially difficult.

Existing anomaly detection techniques mainly fall into three categories: traditional machine learning models [4], [5], [6], [7], deep supervised or semi-supervised approaches [8], [9], and deep unsupervised methods [10], [11]. Regarding the performance of anomaly detection methods, traditional machine learning models can be sub-optimal for event sequence data since they often fail to capture complex data structures [12]. By contrast, deep learning models are more powerful in extracting high-level and complex data feautures [13]. In particular, deep supervised or semi-supervised models are trained to learn a specific classification boundary between normal and abnormal cases based on the given data labels. However, real-world event sequence datasets are often huge in scale with rare anomalies, making it difficult to obtain the required labels for these approaches. Despite that some recently proposed deep learning models are capable of detecting time-series anomalies in an unsupervised manner [10], [11], [14], [15], there is a lack of effective methods that are tailored specifically to discrete event sequence data.

Meanwhile, the black-box nature of deep learning models can introduce great difficulty in interpreting the detection results. Especially in event sequence analysis, anomalies are usually hard to define or interpret by specific rules. Generally speaking, anomalous sequences are those with progression patterns that are different from the majority of the sequences (what we call "normal sequences"). They are identified as outliers due to the unexpected occurrence or absence of some events (what we call "anomalous events"). For example, normal sequences of medical treatments(i.e., treatment plans) for patients with diabetes mainly include regular insulin use. However, some people with diabetes may suffer from complicated heart diseases; therefore, they have rare treatment plans with additional heart medications. The clinical pathways for these patients with diabetic heart diseases shall be identified as outliers due to the treatment plans that are different from the majority of the diabetics. The additional heart medications shall be identified as the event redundancy that signifies the anomalous clinical path.

To support the interpretation of the anomalous sequences, the model must explain how the anomalous sequences are different from the majority of the sequences and which events or subsequences characterize the anomaly. Besides, providing analysts with a comprehensive overview of the entire sequence progression is necessary for analysts to verify the detection results and obtain insights on the practical actions that could help avoid the anomaly. While previous studies have introduced a variety of visual ana-

- *Shunan Guo, Zhuochen Jin, Qing Chen, Nan Cao are with Intelligent Big Data Visualization Lab, Tongji University. Nan Cao is the corresponding author. Email: nan.cao@gmail.com*
- *Hongyuan Zha is with East China Normal University.*
- *David Gotz is with University of North Carolina at Chapel Hill.*

lytics techniques for summarizing event sequence progressions [16], [17], these methods can not efficiently support the needs of visual anomaly detection, such as comparing between anomalous and normal sequences and emphasizing different types of event anomalies.

In this paper, we introduce a visual anomaly detection method for event sequences, extending our previous work on this topic [18]. We leverage the Sequence-to-Sequence Variational AutoEncoder (VAE) to detect anomalous sequences in a collection of event sequences. To enhance the interpretability of the detected anomalous sequence, we further localize the event anomalies by comparing each event in the anomalous sequence with its expected occurrence likelihood derived from the reconstruction probabilities and matching each anomalous sequence with a group of similar normal sequences to investigate their differences. A novel interactive flow-based visualization is designed for summarizing sequence progressions and facilitate comparison of normal and abnormal sequences. Compared to our previous work, the novel aspects of this paper are as follows:

- We improve the previous anomalous event detection method with a sequence matching technique based on Kuhn-Munkres(KM) algorithm [19], which enables more flexible event comparisons across the entire sequence (introduced in Section 4.4). By matching the anomalous sequence with normal sequences, three types of event anomalies can be identified, including (1) event missing, which indicates that an event is expected to occur but is not present in the anomalous sequence; (2) event redundancy, which indicates that an event in the anomalous sequence is not expected to occur; and (3) temporal anomaly, which indicates that an event should occur in a different time interval or in a different order with other events. A sequence matching metric is proposed based on the reconstruction probabilities of events to support making event matches in the context of anomaly detection.
- We optimize previous visualization design with a clustering-based aggregation of normal sequences and a matrix-based visualization of event sequence (introduced in Section 5). The goal is to improve the scalability of the visualization to complex progression paths with a large number of event co-occurrences and facilitates the comparison of events in normal and abnormal sequences.
- We evaluate our proposed anomalous event detection method and the new visualization through two case studies conducted with real-world electronic health records and career paths. We also report subjective feedback collected from an interview with medical experts.

2 Related Work

2.1 Anomaly Detection for Event Sequences

Anomalies in event sequences can be outlier sequences, subsequences, or events, depending on the analysis granularity. For example, kernel-based techniques take each sequence as a data point, then detect outlier sequences from an event sequence dataset based on either distance between points [20] or underlying clustering structure [21]. Window-based techniques try to specify sequence anomaly to particular time intervals by dividing sequences into overlapping subsequences with fixed [22] or variant time intervals [23]. Markovian techniques leverage probabilistic models to predict the occurrence probability of each event, so as to detect anomalous events in all sequences.

A variety of neural network architectures, including recurrent neural networks (RNNs) [8], [24], and convolutional neural networks (CNNs) [25], have been proved to achieve better performance in modeling sequential structures and detecting outlier sequences when compared with machine learning models. For example, Du et al. [8] and Vinayakumar et al. [24] employed recurrent neural networks with a long short term memory (LSTM) architecture to detect anomalous entries in system log files. Similarly, Yun et al. [9] built a prediction engine using LSTM-RNN to detect malicious activities on computers. Kim et al. [25] leveraged the benefit of CNN in extracting spatial characteristics and combined CNN with LSTM to detect both spatial and temporal anomalies in web traffic signals. These models are generally trained in a supervised or semi-supervised manner and require samples with high-quality labels, which is not usually available in real-world settings.

Deep autoencoders (AE) have been gaining popularity due to their capability in identifying anomalies in an unsupervised manner. AE encodes the key features of the data samples into compact latent vectors and decodes them to the original data through reconstruction. Data samples receiving high reconstruction errors are respectively identified as anomalies. Zhou et al. [10] leveraged AE to separate noisy and outlier data from normal data according to their ease of reconstruction. Lu et al. [26] combined AE with RNNs to identify anomalous time windows in temporal data based on the reconstruction error. With the dissemination of the deep generative model, recent studies have also attempted to use Variational AutoEncoders(VAEs) for anomaly detection. Compared to AE, VAE incorporates probabilistic inferences from the data and has better robustness to data noises [27]. Xu et al. [28] employed VAE to detect anomalies in seasonal KPIs and provided an interpretation of the reconstruction probabilities based on kernel density estimation. Similarly, VAE has also been applied to many other time-series applications [29], [30], such as network intrusion detection [11], [31] and sensor monitoring [32].

Despite the wide application of these unsupervised learning models in the anomaly detection of time-series data, very few studies have been carried out on the anomaly detection of discrete event sequences. It is an even more challenging problem due to the irregular (i.e., not evenly-sampled) occurrence of events and a greater variety of anomalous cases introduced by diverse event types and sequential patterns. Moreover, the various forms of anomalies also pose a challenge to result interpretation, which has not been sufficiently studied in previous work [33]. In this paper, we leverage VAE to identify anomalous event sequences in a sequence dataset. We further explain the detected anomalous sequences by localizing anomalies on the individual events through the interpretation of reconstruction probabilities and comparing between the outlier sequences and the normal sequences.

2.2 Visual Anomaly Detection

The boundary between normal and outlier data is often not precisely defined and requires subjective judgment. To

incorporate human domain knowledge into the analytical process, researchers have developed many visual anomaly detection tools to support this procedure [34], [35], which allows domain experts to leverage their knowledge and experience can help overcome these challenges. For these tools, effectiveness and intuitiveness are both key design priorities, and a number of alternative visual analysis approaches have been proposed. This includes methods for the detection of anomalous user behaviors from sequence data [36]. Chae et al. [37] applied traditional control chart methods together with seasonal trend decomposition to extract outliers. Thom et al. [35] introduced a visual analysis system to monitor for anomalous bursts of keywords. More recently, FluxFlow [38] was developed to reveal and analyze anomalous information processes in social media.

Although systems mentioned above are often designed to help detect anomalous points, few approaches focus on identifying anomalous sequential patterns. Recent advances in visual sequence summarization, such as EventFlow [39] and DecisionFlow [17], enabled sequence grouping by progression pathways. While these techniques are efficient in communicating rare sequential patterns that exist in only a small group of people, they rely on the permutations of event orders and may diffuse major sequence groups with similar progression patterns but slightly distinct event orders. To enable the detection of rare event sequences at a higher level, we leverage the benefit of deep learning model, and design our visualizations to fill the gap between the powerful data abstraction capability and the lack of interpretability for the model.

2.3 Visual Comparison Techniques

Visual comparison is a common task when investigating data similarities and differences [40]. In the information visualization domain, Gleicher et al. [41] classify visual comparison techniques into three categories: juxtaposition by comparing objects side-by-side, superposition by overlaying data with a shared reference in the same space (e.g., [42]), and explicit encoding by directly computing and presenting the differences or correlations (e.g., [43]). Each approach has its advantage, and multiple methods can be employed in combination to make a comparison. Within the three major categories, a variety of alternatives have been developed for specific tasks. For example, Kehrer et al. [44] proposed a formal model for hierarchically-partitioned category comparison with small-multiple displays. This approach was inspired by the ineffectiveness of juxtaposition when dealing with a large number of categories. Their work supports superposition and explicit encoding of differences for semantically meaningful comparisons.

Visual comparisons have also been studied in the context of event sequence analysis. MatrixWave [45] applied superposition with an explicit encoding of sequence differences when comparing two event sequences. EventAction [46] used a calendar view to show several temporal event sequences and placed them in a ranked list to compare different sequences via juxtaposition. Most of the previous work focuses on the visual comparison of single sequences (one-to-one), or of event sequence groups (many-to-many). However, for anomaly detection, comparing between one anomalous sequence in the context of similar normal sequences (i.e., one-to-many comparison) is crucial for understanding why the sequence is detected as an anomaly. Therefore, in this paper, we focus on designing visualizations that facilitates comprehensive comparison between an anomalous sequence and multiple normal sequences with similar progression patterns.

3 System Requirements and Design

The goal of our system is to support interactive exploration and interpretation of anomalies in event sequence data. The system was designed and iteratively improved following pre-established requirements. In this section, we elaborate on the detailed design requirements of our system, followed by the system overview description.

3.1 Design Requirements

The detailed design requirements of our system were distilled from (1) feedback collected in initial interviews with medical experts who had a need for detecting outlier patients in electronic health records, (2) authors' experiences with event sequence analysis, and (3)an extensive survey of existing techniques and their limitations.

R1.Provide inspection on the possible anomalies from unlabeled datasets. Real-world event sequence datasets often contain a large number of unlabeled sequences. Users often need to narrow the inspection scope to a smaller group of sequences that require attention. For example, the medical experts commented that they typically filter out problematic patients based on specific criteria before drilling into detailed clinical events.

R2.Facilitate interpretation by identifying anomalous events within abnormal sequences. The interpretability of the anomalous sequences highly relies on the analysis of low-level events. For example, the clinical path of a patient may be detected as an anomaly due to a misused medicine, and software may be considered suspicious due to abnormal file executions. However, real-world event sequences can be long in length and heterogeneous in types, which makes it challenging to identify anomalous events.

R3.Support anomaly analysis in the context of entire sequence progressions. Instead of focusing on the anomalous events, analyzing anomalies within the context of entire sequence progressions can help illustrate the cause and consequences of the anomalous events. Specifically, the medical experts stated that early prevention could be achieved if we know what led to the occurrence of an anomaly.

R4.Allow case-based reasoning to gain user trust and help explore higher-level anomalous patterns. The lack of explainability in deep learning models inhibits user trust in the analysis result. Recent studies tackled this issue through case-based reasoning [47], which generates explanations based on similar cases in the dataset. Medical experts expressed similar interests in following treatment plans under normal and abnormal circumstances to understand how the anomalous patient deviates from typical cases. Moreover, comparing normal and abnormal sequences can reveal higher-level anomalous patterns (e.g., anomalous subsequences or event ordering) beyond low-level ones.

R5.Provide multi-level aggregation and comparison to explore the full hierarchy and interpret various sequence analysis results. Applying different levels of aggregation for

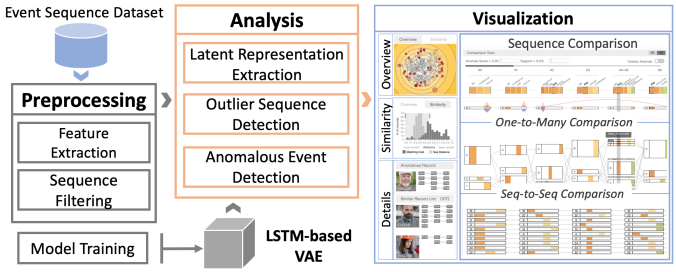


Fig. 1. The analytical pipeline integrates three major modules to support interactive visual anomaly detection of event sequence data, including a preprocessing module, an analysis module and a visualization module.

a group of sequences can result in distinct interpretations of the result. For example, the anomalous events detected by comparing an anomalous sequence with an individual normal sequence may be different from the result when comparing with a subgroup.

3.2 System Overview

Motivated by the requirements above, we developed an interactive visualization system to detect and visualize anomalies in temporal event sequences. As illustrated in Fig. 1, the system includes three major modules: (1) a data preprocessing module, (2) an anomaly detection module, and (3) a visualization module.

The data preprocessing module focuses on reducing noise to prepare high-quality event sequence data for subsequent training of the anomaly detection model. In particular, we measure the importance of each event using Term Frequency-Inverse Document Frequency (TF-IDF) scores to remove noisy events and exclude extremely short sequences (i.e., sequence length < 2) [16]. In the analysis module, we first highlight the anomalous sequences (**R1**) in the dataset using a trained VAE model with the LSTM networks. Then, we localize anomalous events within the detected outliers by referring to each event's occurrence probabilities in typical cases (**R2**) derived from the model. Moreover, by matching with normal sequences (**R4**), we identify three types of event anomaly, including event missing, redundancy, and temporal anomaly. The analysis results are then sent to the visualization module for interactive visual analysis of the sequence progression (**R3**) via multi-granular sequence exploration and comparative analysis with similar normal sequences (**R4, R5**).

4 VAE-Based Anomaly Detection

This section first illustrates how we design the algorithm to address the critical challenges of detecting anomalies in event sequences. Then, we introduce the VAE-based anomaly detection model for detecting anomalous sequences. Finally, we describe a sequence matching algorithm that compares an anomalous sequence with similar normal sequences for localizing anomalous events.

4.1 Algorithm Overview

Detecting anomalies in event sequences is analytically challenging for three reasons. First, the sequential and temporal nature of event sequences results in complex anomaly structures, which makes it difficult to determine the abnormality of the entire corresponding sequence. Second, event sequence datasets are typically diverse with different lengths and progression patterns, resulting in high variability within the training data. Third, given the characteristics above, it is difficult to characterize abnormalities for interpretation, let alone to further understand the reasons for the model's decisions.

To address these challenges, we adapted a Sequence-to-Sequence VAE to detect anomalies in event sequences interpretably. In particular, we leverage the merits of deep neural networks in learning complex sequential patterns to address the first challenge and the probabilistic foundation of VAE in capturing data variability to solve the second. Finally, we employ the reconstruction probabilities output from the VAE to facilitate the interpretation of the anomalous sequences. As shown in Fig. 2, the algorithm consists of four major steps. In the first step, we train a VAE-based model to extract low-dimensional feature representations (i.e., the latent vector $\boldsymbol{z}$) to characterize the progression of each input sequence. The second step employs the latent vectors to measure the outlierness for each sequence based on their Local Outlier Factor (LOF), which is then used to identify anomalous sequences **(R1)**. The latent vectors are fed to the decoder of the VAE model for sequence reconstruction in the third step, which recovers the expected probabilities for each event in each time slot of the input sequence. In the final step, the anomalous sequence is matched with normal sequence utilizing a matching metric based on the event probabilities derived from sequence reconstruction so as to detect event anomalies **(R2)**.

4.2 LSTM-Based Variational AutoEncoder

The Sequence-to-Sequence VAE model contains two modules: the VAE encoder and the VAE decoder. Both modules are designed using RNNs to better extract sequential patterns from event sequence data. In particular, the encoder captures the latent distribution of sequences, and the decoder inversely restores the distribution to estimate the occurrence probabilities of events in each time slot.

VAE Encoder. The encoder is trained to abstract the input sequence $\{X = \boldsymbol{x}_i\}_{i=1}^{n}$ into a low-dimensional latent feature vector that describes a sequential distribution of events occurring in the sequence. In this input, n is the length of the sequence and $\boldsymbol{x}_i \in \{0,1\}^{|E|}$ is the multi-hot encoding of the events co-occurring in the i-th time slot. E is the set of unique events in the sequence dataset. Each coordinate of the multi-hot encoding corresponds to an event type, which is marked one if the corresponding event occurs in the i-th time slot, and 0 otherwise. After feeding the multi-hot vectors into the corresponding layer of RNN, the state of the entire sequence is extracted and represented in the hidden state vector $\boldsymbol{h}_{enc}$ of the last layer, which is denoted as follows:

$$\boldsymbol{h}_{enc} = encoder(X) \tag{1}$$

The hidden state vector $\boldsymbol{h}_{enc}$ is projected into two vectors $\boldsymbol{\mu}$ and $\boldsymbol{\delta}$ to parameterize a normal distribution, representing the mean value and standard deviation of the normal distribution respectively. To take the variability of the latent space into account (i.e., to represent the diversity present in normal cases), we draw a low-dimensional latent vector $\boldsymbol{z}$ by randomly sampling from the distribution. We then use

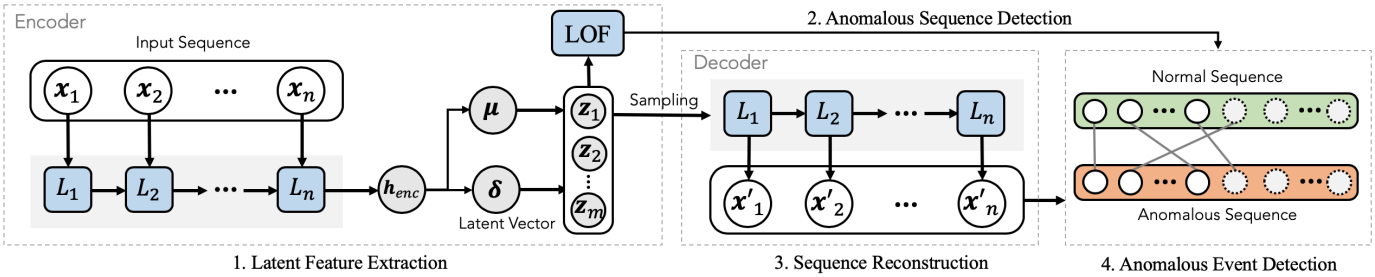


Fig. 2. Schematic diagrams of the model, (1) the VAE model to obtain the latent vector of the input sequence, (2) anomaly detection of the overall sequence, (3) uncover expected event occurrence probabilities from sequence reconstruction, and (4) detecting anomalous events by matching anomalous sequences with normal sequences.

this vector as a representative of the original distribution for subsequent decoding.

VAE Decoder. In the decoder, we reconstruct the input sequence from the extracted latent feature vector $\boldsymbol{z}$. Specifically, $\boldsymbol{z}$ is fed to each layer of the RNN to estimate the probability distribution of events for each time slot. We formally define the decoding procedure as follows:

$$X^{'} = decoder(z) \tag{2}$$

where $X^{'} = \{\boldsymbol{x}_i^{'}\}_{i=1}^{n}$, and the element $x_{i,j}^{'}$ in $\boldsymbol{x}_i^{'} \in \mathbb{R}^{|E|}$ represents the occurrence probability of the j-th event at the i-th time step.

Training Process. We train the model intending to maximize the likelihood of the observed data. The objective which we aim to maximize is defined as the variational lower-bound of the marginal likelihood, which can be formalized as follows:

$$L = L_r + w_{kl} \cdot L_{kl} \tag{3}$$

$$= E_{p(\boldsymbol{z}|X)}[\log p(X|\boldsymbol{z})] - w_{kl} \cdot KL(p(\boldsymbol{z}|X)\|p(\boldsymbol{z})) \tag{4}$$

where $p(\boldsymbol{z}|x)$ is the posterior distribution of $\boldsymbol{z}$ estimated by the encoder, and $p(x|\boldsymbol{z})$ is the likelihood of events estimated by the decoder. $p(\boldsymbol{z})$ is the prior distribution and is set to be the standard normal distribution $N(0, \mathrm{I})$. The first term L_r is the reconstruction of X that we aim to maximize, and the second term L_{kl} is the Kullback-Leibler (KL) Divergence, which estimates the difference between the posterior and prior distribution of $p(\boldsymbol{z})$, which we aim to minimize. In particular, L_r was optimized using Monte-Carlo estimation. To allow the error to get back-propagated through the network, we employed the reparametrization trick proposed in a prior study [48] to make the gradient descent feasible. The KL Divergence serves as a regularization term on the latent space to avoid over-fitting on L_r. These two terms are balanced with a parameter w_{kl}.

Parameter Settings. Both the encoder and decoder employ LSTM units [49] with 300 hidden nodes. We set the dimension of the latent vector to 16. The parameter w_{kl} adaptively increases from 0.1 to 0.5 during the training process to make sure the reconstruction loss is optimized with high priority. The overall objective is optimized with Adam optimizer [50].

4.3 Anomalous Sequence Detection

After training the model, we employ the latent vector $\boldsymbol{z}$ of each input sequence to detect anomalous sequences in the dataset **(R1)**. Although prior VAE-based anomaly detection methods typically use reconstruction probabilities as the detection metric [27], [28], they mostly focus on detecting anomalous data points in time-series. As stated in Section 4.2, the reconstruction loss is trained to maximize the likelihood of individual events, which may fail to represent the overall sequence progression. In contrast, the latent vectors are trained to learn a feature for each event sequence in the context of the entire sequence dataset, and can be better suited for identifying anomalous sequences from the sequence dataset. Therefore, we employ the Local Outlier Factor (LOF) [51] to evaluate the outlierness of each sequence in the latent space using the latent vector $\boldsymbol{z}$.

In the unsupervised anomaly detection process, it is assumed that the majority of the sequences in the dataset are normal. As the LOF score compares the local density of the latent vector with its neighborhood vectors, normal sequences should group within a dense space with smaller LOF scores, while instances in sparse areas will have larger LOF scores and will be identified as outliers. Specifically, the threshold of the LOF scores is bounded by the median absolute deviation (MAD) [52] considering its robustness to extreme LOF scores. A sequences with a LOF score that lie outside Median $\pm\, k * \mathrm{MAD}$ is considered as anomalous. We conservatively set $k = 3$ following a prior study [53].

4.4 Anomalous Event Analysis

To facilitate the interpretation of sequence anomalies, we further identify anomalous events that contribute to sequence abnormality **(R2)**. As mentioned earlier, the reconstruction probabilities are restored from the latent vector $\boldsymbol{z}$ that is sampled from the the latent space where the majority of the sequences are normal, and the training objective ensures that the reconstruction probabilities are also similar to the original input sequence. Therefore, the reconstruction probabilities of the anomalous sequences can be used to infer an expected occurrence likelihood of events in typical cases, where $\boldsymbol{x}_i'$ represents the expected occurrence likelihood of all events in the i-th time slot. Intuitively, we can consider events that violate their expected occurrence probabilities as abnormal. For example, if an event has a high likelihood of occurrence but is not presented in the anomalous sequence, it is likely to be an event missing anomaly. On the contrary, if an event has a low likelihood of occurrence but appears in the anomalous sequence, it is likely to be an event redundancy anomaly. However, the reconstruction probabilities are not always reliable as the values highly rely on the precedent occurrence of events.

For example, the reconstruction probabilities of all events in the time steps at the beginning of the sequence are generally very low, due to the lack of progression context from precedent events. In addition, the model provides little explanation on how the probabilities are estimated, and users may find it difficult to apply the probabilities for determining the boundary of the anomaly.

To support detecting anomalous events in a more interpretable manner, we further incorporate the progression of normal sequences **(R4)** into our analytical context. Specifically, we compare each sequence with a group of normal sequences that are close to the anomalous sequence in the latent space to investigate their differences. Wongsuphasawat et al. [54] introduced a sequence comparison method that aligns two sequences by events and quantifies event differences with a Match & Mismatch Measure. However, this measurement treats all types of mismatched events equally, which may not accord with the real-world situation, because not all events that occur in normal sequences should appear in the anomalous sequence. For example, in the medical scenario, patients in the normal group may be diagnosed with a certain complication that does not appear in the anomalous sequence. The complication, however, is not likely to occur under the progression context (i.e., previous lab test events or treatments) of the anomalous sequence, and should not be considered abnormal.

Therefore, we propose a new matching metric that incorporates the reconstruction probabilities of the anomalous sequences, so as to bring the progression context of the anomalous sequence into sequence matching. Specifically, we transform the comparison between two sequences into a bipartite graph matching problem, where only events of the same type are allowed for matching. The goal is to search for a maximal matching between two sequences, and event assignments with relatively high costs will be identified as abnormal. We first initialize the bipartite graph with two sets of nodes, corresponding to events in two sequences. In addition, we append a sufficient number (i.e., no less than the number of events on the opposite side) of *null* events to each set for null assignment (i.e., isolated events without matching). We leverage the Kuhn-Munkres algorithm [19] to find event correspondence with a minimum cost. Intuitively, three types of event anomalies can be identified in light of different event matching situations, including:

- Event missing, which corresponds to events in the normal sequence matching to *null* event, meaning that an event appears in the normal sequence but not in the anomalous sequence. For example, suppose a medical sequence contains a precedent diagnostic event of severe diabetes but no record of taking Insulin. In that case, it is very likely that an event missing of Insulin will be identified, since the majority of patients with severe diabetes have Insulin in their treatments.
- Event redundancy, which corresponds to the opposite of event missing, meaning that an event appears in the anomalous sequence but not in the normal sequence. This anomaly can happen, for example, if a patient with diabetes takes glucose in their treatment.
- Temporal anomaly, represented by events matching across time slots, meaning that the event appears in both normal and abnormal sequences but in different time slots. For example, the treatment events often occur after hospital admission. However, for patients that enter the hospital under an emergency, the treatment events can be recorded before the admission event. In this case, a temporal switch between two events may be derived from matching with normal medical records.

Base on these intuitions, we define the cost for matching two events – (e_1, t_1) in the anomalous sequence and (e_2, t_2) in the normal sequence – as follows:

$$C((e_1,t_1),(e_2,t_2)) = C_0((e_1,t_1),(e_2,t_2)) + C_1(t_1,t_2) \quad (5)$$

where C_0 represents the gap between the expected likelihood of occurrence derived from the reconstruction probabilities and the actual event occurrence, and C_1 represents the time gap between two matched events. In particular, C_0 is defined as follow:

$$C_0(\cdot) = \begin{cases} 1 - P(e_1,t_1) & e_1 \neq null, e_2 = null \\ P(e_2,t_2) & e_1 = null, e_2 \neq null \\ |P(e_1,t_1) - P(e_2,t_2)| & e_1 = e_2 \neq null \end{cases} \quad (6)$$

where $P(\cdot)$ represents the reconstruction probability of the event in the corresponding time slot. The first case represents event redundancy, where the event in the anomalous sequence is matched with *null* event in the normal sequence. By contrast, the second case represents the event missing. The third case represents matching between two events, which allows not only matching between two events in the same time slot but also across different slots. The cost of event matching across different time slots is also determined by C_1, which indicates the time gap between two events defined as $C_1(\cdot) = |t_1 - t_2|$. Intuitively, the cost of matching between the same events within the same time slot shall be 0 and considered as normal. Event missing with high expected occurrence probability and event redundancy with low expected occurrence probability also have high matching costs. Event matching across different time slots indicates a temporal shifting of event, and the cost is determined by the temporal distance of two events. The time complexity for matching two sequences is $O(n^3)$, which can be a concern if the sequence length and the number of sequences are large. In our implementation, we pre-calculated and stored the sequence matching results to avoid affecting the subsequent visual analytics.

Discussion. While some AE/VAE-based model [27], [28] support identifying anomalous data points in time-series data from the value of reconstruction errors/probabilities, which can be adapted to detect event anomalies in discrete event sequences, our method enhanced the interpretability of the anomalies from two perspectives. First, our model filters out a small set of anomalous sequences by considering the context of each sequence regarding the entire dataset. In contrast, models relying on reconstruction errors/probabilities focus more on the context of an event regarding the progression of a sequence, which can produce a relatively large set of anomalous sequences for inspection. Second, it can be difficult to interpret the meaning of reconstruction errors/probabilities and how they may relate to the event anomalies. By matching the anomalous sequence with normal sequences, users can easily interpret the event anomalies through matches and mismatches of

events and better understand how the anomalous sequences are different from the normal sequences.

5 VISUALIZATION

5.1 User Interface

The visualization system comprises seven key views to visually analyze anomalous sequences (Fig. 3). The analysis starts from the *anomaly overview* (Fig. 3(1)), which provides an MDS projection of the latent vectors for all anomalous sequences in the dataset and allows users to select an anomalous sequence for subsequent analysis **(R1)**. The *similarity view* (Fig. 3(2)) displays the distribution of all normal sequences and their similarities to the selected anomalous sequence, which is derived from the distance of corresponding latent vectors. In particular, we measure the sequence similarity in two ways: their cost for matching events in the normal sequences to the selected anomalous sequence (noted as *matching cost*) and the distance between the latent vectors of normal sequences and the anomalous sequence (noted as *sequence distance*). Sequences with small matching costs generally have more similar events, while sequences with small distances to the anomalous sequence usually contain key progression patterns, such as an indicator of a particular type of disease. From this view, users can switch between different similarity measurements to inspect different distributions and select a group of normal sequences to compare with the anomalous sequence in the main panel **(R4)**.

The main panel supports visual interpretation of the selected anomalous sequence via sequence comparison. Specifically, the main view is vertically divided into three major parts, including 1) a *reconstruction view*(Fig. 3(3)) showing events occur in each time slot and their occurrence likelihood; 2) an anomalous sequence view (Fig. 3(4)) showing the progression of the selected anomalous sequence, with the type of abnormality being marked out on anomalous events **(R2, R3)**, and 3) a *normal sequence view* (Fig. 3(5)) summarizing the progression of similar normal sequences that are selected by the user from the *similarity view* **(R4)**. Access to raw sequence data is provided via the *anomalous record view* (Fig. 3(6)) showing details about the selected anomaly and *similar record list* (Fig. 3(7)) showing detailed events in similar normal sequences.

5.2 Interactive Anomaly Interpretation

We design the main panel of our system to enable interactive exploration of the analysis result and facilitate comparison between abnormal and normal sequences. As shown in Fig. 3, the progression of selected anomalous sequence is displayed in the *anomalous sequence view* (Fig. 3(4)) **(R3)**. Views at the top and the bottom aim to support a comprehensive interpretation of the anomalous sequence from two perspectives: 1) the expected occurrence probabilities of each event inferred from the reconstruction probabilities introduced in Section 4.2, which is displayed in the *reconstruction view* (Fig. 3(3)), and 2) the event differences derived from a comparison between the anomalous sequence and similar normal sequences through the *normal sequence view* (Fig. 3(5)) **(R4)**. In the following, we introduce the design of each view in detail, respectively.

5.2.1 Reconstruction View

To leverage the interpretable information from the anomaly detection model, we design a *reconstruction view* to provide an overview of all events in each time slot and their corresponding occurrence probabilities. The time slots are determined by segmenting the selected anomalous sequence into event groups with close timestamps. In particular, events with a time distance below a threshold (e.g., a day or a year) will be put into the same time slot, indicating that all events in the slot occur in a certain time interval. The threshold of the time distances is dependent on the time span of the anomalous sequence and the frequency of events. For example, in medical data, treatments or lab tests are generally applied on a daily basis, and the time slot can be set as a day. In contrast, a person's career path may span over several years with sparse key events (e.g., education, promotions), and the time slot can be set as a year.

Event orders and frequencies in each slot are omitted, but the temporal order of events across different slots are preserved, ranging from left to right with the slot number labeled at the top (Fig. 4(1)). Events in each time slot are shown as rectangular nodes, ordered from left to right by their occurrence probabilities. The occurrence probabilities are also globally encoded with an orange-to-green color gradient and is consistent in all views in the main panel. To help users better track events across different views, we vertically align the same type of events in each time slot events, and the labels of event types can be found at the top of each node.

5.2.2 Anomalous Sequence View

The selected anomalous sequence is organized into sequence segments aligning with the time slots (Fig. 4(2)). Events in adjacent time slots are separated by a label of the sequence ID. In addition, events in each time slot are vertically aligned with events in the *reconstruction view* to help identify the event labels.

We emphasize the event anomalies in the anomalous sequence with a set of glyphs (Fig. 4(a-c))) representing the three anomaly types. The glyphs are designed based on the metaphor of editing symbols, intending to convey insights on operations that are required to transform an anomalous sequence into normal. Specifically, we represent event redundancy with a *delete* symbol (Fig. 4(a)), event missing with an *insert* symbol (Fig. 4(b)), and temporal anomaly with a *move* arrow (Fig. 4(c)) pointing from the observed time slot to the expected time slot. Event missing and the endpoint of the temporal anomaly is encoded using a white rectangular node with a dashed border, indicating that an event is expected to occur but is not present.

Additionally, we retrieve the subgroup of normal sequences for each type of event anomaly that "support" the corresponding event to be abnormal as the comparison group. For example, in Fig. 3, the comparison group for the event redundancy in the first slot ("#0") shall be the top two clusters in the *normal sequence view* below where the event is also not presented. We summarize the abnormality of events analyzed by matching the anomalous sequence to all selected normal sequences and integrating all anomaly types identified for each event. By default, we select the dominant type of anomaly that has the largest comparison group for

Fig. 3. The user interface of the system consists of seven key views to support comparison-based visual anomaly detection, which includes an (1) anomaly overview, a (2) similarity view, a (3) reconstruction view, an (4) anomalous sequence view, a (5) normal sequence view with two variants (5A, 5B) in cluster mode and sequence mode, respectively, an (6) anomalous record view and a (7) similar record list.

display. However, users can change their focus by selecting a different subgroup of normal sequences during analysis for comparison. We calculate a *support rate* as the proportion of the selected normal sequences in the comparison group, and the *anomaly score* as the average matching cost for each event anomaly to help users justify the level and the confidence of abnormality. The anomaly score is displayed with the height of a red peak above, and the support rate is encoded with the height of a blue peak under the anomalous event.

5.2.3 Normal Sequence View

The selected normal sequences are displayed in the normal sequence view (Fig. 3(5)). To allow comparative analysis at different granularity of sequence aggregation (**R5**), the normal sequence view is designed to support two visualization modes: the sequence mode (Fig. 3(5B)) for preserving the individual details of the normal sequences and the cluster mode (Fig. 3(5A)) for enabling inspecting the progression paths of a larger number of normal sequences at a time. Users can switch between two visualization mode using the button at the top of the main panel (Fig. 3(c)).

Sequence Mode. The sequence mode of the *normal sequence view* displays the sequences of normal records individually, aiming to support sequence-to-sequence comparison and efficient access to the raw data of normal sequences. As shown in Fig. 3(5B), the normal sequences are displayed in a scrollable list. The encoding schema of each individual sequence is kept consistent with the anomalous sequence for easy comparison. Users can select any individual sequence to compare the anomalous sequence with, and the event anomalies marked in the *anomalous sequence view* shall be updated accordingly. The sequences are ranked from top to bottom with their gradually increasing matching cost or sequence distances to the anomalous sequence in the latent space, depending on the metric utilized in the selection of normal sequences in the *similarity view* (Fig. 3(a)). Analysts can focus their comparison to the first few sequences to investigate the minimum effort of turning the anomalous sequence into normal, or sequences with similar progression context in order to avoid introducing noisy event comparison results.

Cluster Mode. The cluster mode of the normal sequence view summarizes the progression of normal sequences into a flow-based visualization by clustering sequence segments in each time slot. In particular, the sequence segments in each slot are clustered using Mean Shift Clustering [55] based on the multi-hot vectors introduced in Section 4. Note that we include the segments in the anomalous sequence to generate the clusters, and take the anomalous sequence away from the corresponding clusters when displaying only the normal sequences.

The visualization is designed to support comparing the anomalous sequence with subgroups of normal sequences that have particular progression patterns. Each cluster is represented with a rectangular node (Fig. 4(d)), consisting of a left-side label displaying the number of sequences clustered in each node, and the main content depicting the event occurrence of the clustered sequence segments. Note that the text in the left-side label is rotated to distinguish from the sequence id annotated in the presentation of individual sequences. The occurrence of each event is encoded with a vertical bar (Fig. 4(e)), horizontally aligned with the same type of event in the *reconstruction view* and the *anomalous sequence view*. The height of each bar is proportional to the number of sequences in each cluster having the corresponding event occurrence. The color encoding is consistent with other views showing the occurrence probability. We set the vertical position of cluster nodes in each time slot using a layout algorithm introduced in [56] to illustrate the similarity between clusters and minimize link crossing intuitively. Cluster nodes with similar event occurrences are grouped together, illustrating a higher-level structure of sequence progression. Cluster nodes in adjacent time slots are connected with light grey links (Fig. 4(f)) to uncover the

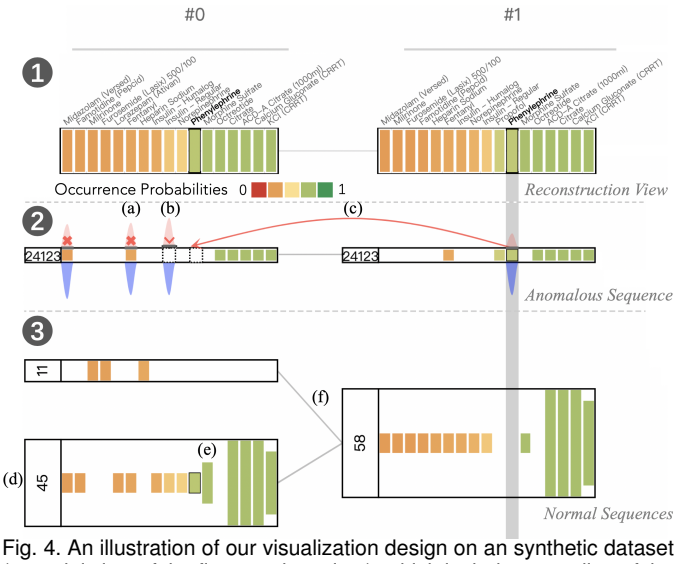


Fig. 4. An illustration of our visualization design on an synthetic dataset (a partial view of the first two time slots), which includes encoding of the (1)reconstruction probabilities of events in each time slot, (2)events in the anomalous sequence with different types of event anomalies(a–c) highlighted, and (3)the progression of normal sequences for comparison. Segments of normal sequences in each time slot are clustered(d), and each cluster node is encoded with the pattern of event occurrences(e). The links connecting the clusters indicates the progression of normal sequences(f).

transition patterns of sequences between clusters.

The cluster nodes can be expanded to show specific sequence segments within and contracted back to a summarized cluster node in response to a double click (as shown in Fig. 3(h)). This design aims to help users decide the granularity of analysis with more flexibility. The system also supports clustering analysis with the selected anomalous sequence incorporated to help analysts overview how the progression of anomalous record deviates from the normal group. By switching on the overlay button (Fig. 3(d)), the anomalous sequence segments are added back to the clusters. The cluster nodes and transition links that the anomalous sequence progress through are highlighted in red (as shown in Fig. 6(1)) so that users can quickly identify problematic time slots in which the anomalous sequence fall into the clusters with a small population.

5.3 Other Views

The system includes several contextual views to display statistical information and provide access to raw data. These views are coordinated with the selections and filters in other views to support the interpretation of the anomaly.

Anomaly Overview. The *anomaly overview* (Fig. 3(1)) is designed to help select sequences of high anomaly degree for subsequent analysis. It shows the distribution of normal and abnormal sequences in the latent space via the Multi-Dimensional Scaling (MDS) projection of their latent vectors $\mathbf{z}$. Each sequence is represented as a circle. The anomalous sequences are colored in red and normal sequences are colored in white. The size of the red circles indicates the LOF score (i.e., level of abnormality), and the color saturation shows the average matching cost. The distance between the two circles reflects their similarity, and a colored contour map is designed to illustrate the local density of circles. Intuitively, circles with larger size and in low-density areas are the most likely anomalies. To accommodate large event sequence datasets, we cluster the nodes of normal sequences using CURE [57] if the size of the dataset exceeds 300, preserves only the representative nodes for display, so as to avoid node overlaps while preserving the relative positions of the normal and abnormal sequences in the latent space.

Similarity View. The *similarity view* (Fig. 3(2)) displays the distribution of all normal sequences in the dataset by their similarity to the anomalous sequence, which aims to help users select a proper group of normal sequences to initiate the comparative analysis. Users can switch between two similarity measurements by clicking on the legend (Fig. 3(a)), and then select normal sequences to get an overview of how the anomalous sequence deviates from the distribution of normal sequences in the main panel.

Anomalous Record View and Similar Record List. The *anomalous record view* (Fig. 3(6)) and the *similar record list* (Fig. 3(7)) demonstrates raw data of the selected anomalous sequence and normal sequences, further supporting interpretation with the low-level detailed evidence. In particular, records displayed in the *similar record list* are kept consistent with the user's selection in the *normal sequence view*.

5.4 Interactions

The system additionally includes the following interactions to facilitate exploratory analysis.

Stage Merging. We leverage a recently proposed progression analysis technique [56] to segment the anomalous sequence into different stages. As illustrated in Fig. 3(e), stages are marked with line segments under the identifier of the time slots. Users can click on a stage identifier to merge or expand all visual elements in the main panel that align with the corresponding time slots. This interaction aims to reduce the length of sequences for a more efficient exploration while also providing a different level of aggregation granularity for normal sequences in the temporal-level other than the sequence-level (i.e., individual and clustering of normal sequences).

Selecting and Filtering. Our system allows users to navigate the visualization and make more focused inspections through flexible data selection and filtering. For example, the system allows users to select both individual sequences of interest or subgroups of sequences following particular progression patterns from the sequence mode and clustering mode of the *normal sequence view*, respectively. After a selection, the system reruns the comparison between the anomalous sequence and the selected normal sequences to update the comparison result in the *anomalous sequence view*.

The system incorporates two types of filters for users to tune the event anomalies, including an anomaly score filter and a support rate filter (Fig. 3(b)) for supporting the dynamic adjustment of the detection boundaries. Users can tune the filters to preserve only event anomalies with a high abnormality level and high confidence as supported by most of the normal sequences.

Highlights and Tooltips. The system is equipped with linked-highlighting, which helps users track the occurrence of a selected event type across different views. Specifically, when users hover their mouse over an event, all visual elements representing the same event type will be simultaneously highlighted in all views. A vertical bar (Fig. 3(f))

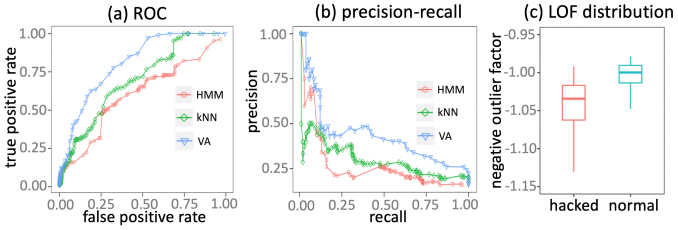


Fig. 5. Performance evaluation results of our VAE-based algorithm (VA) in comparison with two baseline methods (kNN, HMM). The (a) ROC curves and (b) precision-recall curves indicate that our approach (VA) effectively detects anomalies and outperforms the baseline methods. (c) The distribution of the negative outlier factor on the anomalous and normal sequences.

is displayed to track the corresponding type of events in the same time slot to facilitate visual comparison. Moreover, when users select an individual sequence or a progression path in the *normal sequence view*, all corresponding visual elements are highlighted to mark the users' selection. Finally, descriptive tooltips (Fig. 3(g)) are triggered when hovering over any visual elements in the system.

6 Evaluation

We assess the effectiveness of the analytical model through a quantitative evaluation, and the visualization system through case studies on two real-world data sets in different domains. We finally report subjective feedback gathered from the medical experts who participated in our case study.

6.1 Quantitative Evaluation

Our anomaly detection method is designed to first detect event sequence anomalies within a collection of sequences. Then, we interpret the event sequence anomalies by further identifying event anomalies in each anomalous sequence. To the best of our knowledge, existing anomaly detection techniques for event sequence data are either developed to detect anomalous sequences or anomalous events only. It is not easy to find prior work to make a comprehensive comparison with our technique. Therefore, in this section, we focus on validating the VAE-based anomaly detection algorithm (denoted as VA) in the first step of detecting anomalous sequences to ensure that the set of anomalous sequence for inspection is in high quality, and also comparing its performance with two baseline methods. The dataset we used is an intrusion detection dataset, snd-cert [58], which consists of sequences or operating system calls that are labeled in terms of the system state (i.e., normal or hacked) when running these operations.

Baseline Methods and Evaluation Metrics. We select two representative baseline methods for detecting anomalies in event sequence data in an unsupervised manner under the categories of kernel-based and Markovian anomaly detection techniques: Nearest Neighbor (kNN) [59] and Hidden Markov Model (HMM) [60]. Both methods have been shown efficient for detecting anomalies in event sequence data [61], [62], [63]. We did not include other deep learning approaches in our comparison because most unsupervised deep learning models for analyzing event sequence anomalies are based on auto-encoders (as discussed in Section 2.1), and the model structure is very similar to the one we used in this work.

Specifically, the longest common subsequence (LCS) was used as the distance metric in kNN. We use standard information retrieval metrics (precision, recall, and ROC) to evaluate the performance of our approach and these two baseline methods. Because the number of positive and negative instances are imbalanced in the dataset, we use the precision-recall curves and ROC curves to comprehensively illustrate the performance of the algorithms.

Evaluation Results. Our algorithm outperforms the baseline methods as shown in Fig 5. The ROC plot (Fig 5(a)) illustrates that VA achieves higher true positive rates when the false-positive rates remain low (below 0.25) compared to the other two baseline methods. The precision-recall plot (Fig 5(b)) shows that VA had overall higher precision than the baseline methods. The results indicate that our VAE-based anomalous sequence detection method can produce a set of high quality suspicious sequence comparing to the baseline algorithms. We also compared the distributions of negative outlier factors of the ground-truth anomalous sequences and normal sequences. As shown in (Fig 5(c)), the model separates the abnormal and normal sequence in the latent space, with the negative outlier factors of the normal sequences concentrated around -1 as expected. Using the designed visualization, the system can further support the interpretation of detected anomalies.

6.2 Case Studies

6.2.1 Anomalous Use of Clinical Medicines

We applied our system to analyze MIMIC [64], a publicly accessible critical care database with de-identified electronic health records for 46,520 patients with 12,487 event types. Due to the diversity of sequence progression for patients with different diseases, training with the entire database could introduce noise and produce inaccurate anomaly results. With this consideration, we selected a subgroup of 7,537 patients who were once diagnosed with cardiovascular diseases to produce a more homogeneous set of sequences for training. We train the model on an Nvidia Tesla K80 graphics card with a batch size of 80 for each training step. Each training epoch takes approximately 10.5 seconds on average.

We invite two medical experts, one cardiologist (**E1**) and one nephrologist (**E2**) to participate in our study, considering that severe heart diseases are usually complicated with kidney injuries. Both E1 and E2 are familiar with typical medical events and treatments in their domain. Prior to the experiment, the doctors were asked to select a list of 76 key medical events under the category of only *prescriptions* and *lab events*, which we preserved to clearly illustrate the physical condition of each patient and the treatments plans they followed. We made sure that the experts have no difficulty in understanding the meaning of each event type.

After training the anomaly detection model, 176 out of 7,537 patients were detected as anomalous for subsequent analysis. The study session lasted approximately 1.5 hours, starting with a 15-minute introduction to the dataset and the system design. We then took 10 minutes to demonstrate an example use case. The doctors were asked to explore the clinical records of anomalous patients and provide domain-relevant insights on interpreting the analysis result. The experts were asked to think out loud and make comments

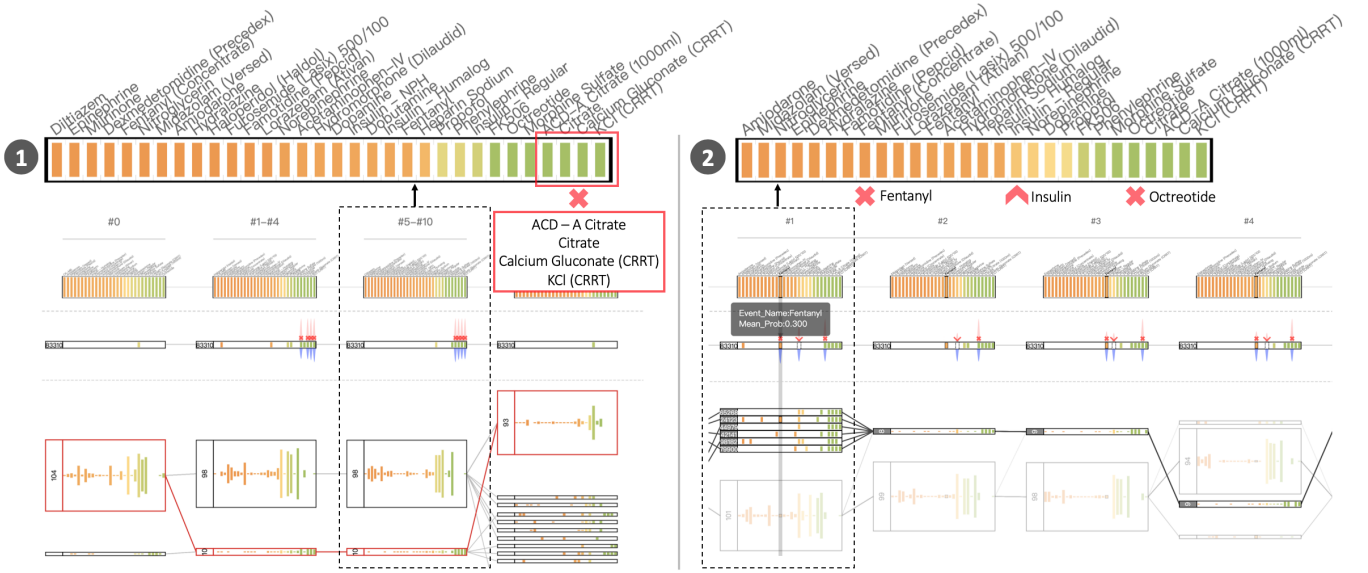


Fig. 6. The anomaly detection result of the MIMIC dataset. (1) The anomalous sequence deviates from the major group of normal sequences and falls into the cluster of a small population in the second(marked as "#1-#4") and third stage(marked as "#5-#10"), due to the occurrence of four medicines. (2) When comparing only with the small cluster, three event anomalies occur across slot 1 to 4.

at any time during the study. The main findings from this study are demonstrated in Fig. 6. This section only reports their findings and comments regarding the study data. Their comments on the system level are discussed in Section 6.3.

The experts selected a dark and large node from the *overview*, which represents a patient with both a high anomaly score and matching cost. They retrieved a group of 107 normal patients similar to the selected anomaly in terms of distance of the latent vector. After the main panel was loaded, the experts scrolled back and forth to review the anomalous events emphasized by the editing symbols in the *anomalous sequence view*. The anomalous events spanned from slot 1 to slot 10, and the experts merged slots in the same stage to narrow the scope of the investigation. After reviewing the overlay mode, they found the anomalous sequence is clustered into the rare groups in the second (slot 1–4) and third stage (slot 5–10) as shown in Fig. 6(1). They noticed that the anomalous events are mainly related to treatments during continuous renal replacement therapy (CRRT), as **E2** said, "it seems that patients in the rare clusters mainly distinguish themselves by particular treatments they used during CRRT. This might indicate that those patients have more severe kidney injury compared with the main group." **E1** also explained, "while kidney failure is a common complication for heart disease, it is rare to see cardiac patients with very severe kidney injury." The experts then select the rare cluster in the *normal sequence view* to figure out why some of the patients taking CRRT treatments are also identified as normal. The event anomalies displayed in the *anomalous sequence view* were updated according to matching results when comparing with the selected normal sequences, and all event anomalies lied in the second stage. Thus, the experts expand the stage for more detailed inspection. As illustrated in Fig. 6(2), the experts noticed three event anomalies, including the missing use of Insulin, redundant use of Octreotide and Fentanyl. **E2** confirmed the missing use of Insulin is indeed abnormal, as he stated, "the blood glucose level can change dramatically during CRRT, it is important to use exogenous Insulin to control blood glucose." **E1** explained the use of Fentanyl, "as Fentanyl is a very strong pain reliever often used in the case of emergency. It is rare to see patients regularly use this treatment in continuous time intervals, unless the patient had experienced rescues multiple times." Regarding the use of Octreotide, both experts do not find it abnormal in the sense of medical usage. "It is sometimes applied to cure cirrhosis," as **E2** stated, "it probably indicates that this patient also has liver damage."

6.2.2 Rare Career Path Detection

We also tested the capability of our system in identifying anomalies in a small event sequence dataset using a career path dataset [65]. The dataset contains ten types of milestone events of 40 university professors, such as receiving degrees, publishing papers, and changing academic positions. We trained the anomaly detection model and selected the anomalous career path with the highest anomaly score for review. The study findings are demonstrated in Fig. 7 After a brief inspection of the stage analysis result and the anomalous sequence, we identified the first stage from slot 0 to 2, the second stage in slot 3, and the fourth stage in slot 6 as degree obtainment period for bachelor, master, and Ph.D., respectively. After selecting a major group of scholars in slot 6 (Fig. 7(a)), the editing symbols demonstrated three event anomalies in the *anomalous sequence view*. As shown in Fig. 7(b), the first stage contains a temporal anomaly of acquiring an Assistant Professor title. The arrow point towards slot 6, indicating this event should normally appear after obtaining a Ph.D. degree. However, the anomalous career path shows that the scholar acquired the title at the stage of obtaining a bachelor's degree before obtaining the master's degree. We also noticed another event anomaly in the first stage: the missing publication of a journal (Fig. 7(c)), which can be verified from the *normal sequence view* that all

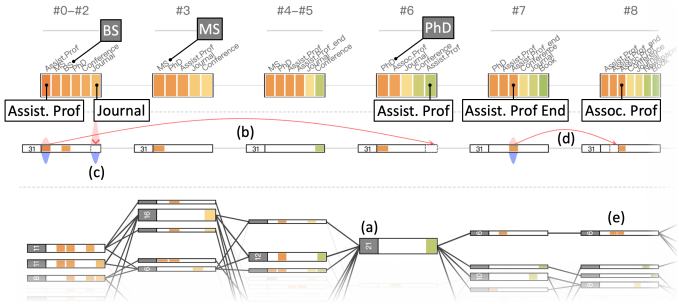


Fig. 7. Anomalies detected when analyzing the career paths for a group of scholars. The system identified three event anomalies (b-d) when comparing with the main group of normal career paths (a).

normal career paths contain conference or journal paper being published in this stage. This also accords with common sense that it is generally considered necessary to publish at least one paper before obtaining the master's degree.

Another temporal anomaly was detected in slots 7 and 8 on the end of Assistant Professor title (Fig. 7(d)), pointing toward the same slot where the scholar acquires the Associate Professor title. This indicates that the promotion from one title to another should normally appear simultaneously, rather than having a time gap in between. The normal situation can be observed from the *normal sequence view* (Fig. 7(e)), supporting the analysis result.

6.3 Expert Feedback

This section reports subjective feedback collected from the medical experts who participate in our first case study. We summarize their comments around three themes: usefulness, system usability, and visualization design.

Usefulness. Both experts agreed that the system is useful in finding anomalies in complex electronic health records. Specifically, **E1** found our system can be very time saving, as he commented: "The analysis tool we currently have is only capable of identifying abnormal lab test values. For clinical decisions, we need to manually check if it is aligned with the medical guideline, which is very time-consuming." **E2** is especially impressed that our system is able to identify various types of anomalies: "I'm surprised to see that the system is not only able to detect the anomalies, but also identify the type of anomalies, which makes it much easier to understand the abnormality and verify the correctness." The experts also stated that the abnormality defined in the context of our system is a bit different from what it is in the sense of medical analysis, as mentioned by **E1**: "While this technique attempts to detect anomalies in the broad sense by identifying the rare cohorts, it would be interesting to see it combining with the determination criteria of anomalies in the medical scenario, [such as anomalous lab test value, medicine usage that violates the guideline, etc]."

System Usability. According to the experts' feedback, the system is very easy to use. Both experts commented that the workflow of the system is "easy to follow". **E2** commented that "I think the system is generally easy to learn and use. It provides guidance for us to explore and find anomalous sequences and events progressively." Although he found most views demonstrated in the system is useful, the *reconstruction view* is a bit redundant: "When using the system, I seldom put my attention to the *reconstruction view*. The [event] anomalies are already marked out clearly, [which makes] the probabilities [of events] less useful." The experts also expressed a desire for the system to support the analysis of multiple anomalous sequences simultaneously, as they felt it is inefficient to analyze anomalous sequences one after another.

Visualization and Interaction Design. Both experts found no difficulty in understanding the visualization design in our system. In particular, **E1** also participated in the case study of our previous work [56], and he applauded the visualization demonstrated in this version: "I felt the aggregation of normal sequences is much more clear compared with the old version, as there is generally fewer nodes and edge crossings in the [*normal sequence*] view." He also liked that the event types being labeled in the *reconstruction view*. "In the previous version, I need to hover on each visual element repeatedly to find the corresponding event type," as **E1** stated, "it is much easier to explore with the labels being displayed". **E2** agreed that tracking events is easy, as he mentioned: "I think aligning events [in each slot] greatly facilitates the comparison. I can notice the missing and redundancy of events at a glance when comparing with the normal sequences." **E2** also appreciated the use of editing symbols to represent event anomalies, as he found it is "very intuitive and easy to understand."

7 Conclusion

We have presented a visual analysis technique for detecting, exploring, and interpreting anomalies in event sequence data. The system incorporates an unsupervised VAE-based anomaly detection model and matches events in anomalous sequences to the normal sequences for detecting event anomalies. Based on the detection result, a visualization system is developed to facilitate interpretation via sequence comparison. We evaluate the effectiveness and usefulness of our system through a quantitative comparison of the performance of our algorithm and two case studies with real-world datasets.

Limitations and Future Directions. To provide a complete context of events for interpreting the anomalies, we preserve each individual events in our system, which obstacles our system to be scaled to numerous types of events. One potential solution is to incorporate event aggregation methods [1] to reduce the number of events without affecting the informativeness for interpreting the anomalies. While there is a lack of unsupervised deep learning model that our model can compare with (as discussed in Section6.1), comparing the VAE model we used in this paper with existing supervised anomaly detection model for event sequence data [9], [24] could help identify the gap of our approach. Moreover, comparing our sequence matching approach for identifying anomalous events with other techniques designed for detecting event anomalies (e.g., [27], [28]) is also an important future work for validating the performance of our algorithm. Our study results also shed light on several future research directions, including incorporating domain knowledge in the determination criteria of anomalies and supporting the analysis of multiple anomalous sequences simultaneously.

REFERENCES

[1] D. Gotz, J. Zhang, W. Wang, J. Shrestha, and D. Borland, "Visual analysis of high-dimensional event sequence data via dynamic hierarchical aggregation," *IEEE TVCG*, vol. 26, no. 1, pp. 440–450, 2019.

[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, p. 15, 2009.

[3] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer Networks*, vol. 51, no. 12, pp. 3448–3470, 2007.

[4] A. Qayyum, M. Islam, and M. Jamil, "Taxonomy of statistical based anomaly detection techniques for intrusion detection," in *Proceedings of the IEEE Symposium on Emerging Technologies*. IEEE, 2005, pp. 270–276.

[5] G. Xiong, J. Cheng, X. Wu, Y.-L. Chen, Y. Ou, and Y. Xu, "An energy model approach to people counting for abnormal crowd behavior detection," *Neurocomputing*, vol. 83, pp. 121–135, 2012.

[6] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *IEEE ICDM*. IEEE, 2008, pp. 413–422.

[7] G. Münz, S. Li, and G. Carle, "Traffic anomaly detection using k-means clustering," in *GI/ITG Workshop MMBnet*, 2007, pp. 13–14.

[8] 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*, 2017, pp. 1285–1298.

[9] Y. Shen, E. Mariconti, P. A. Vervier, and G. Stringhini, "Tiresias: Predicting security events through deep learning," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 592–605.

[10] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD*. ACM, 2017, pp. 665–674.

[11] Q. P. Nguyen, K. W. Lim, D. M. Divakaran, K. H. Low, and M. C. Chan, "Gee: A gradient-based explainable variational autoencoder for network anomaly detection," in *2019 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2019, pp. 91–99.

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

[13] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, no. 1, p. 1, 2015.

[14] A. Singh, "Anomaly detection for temporal data using long short-term memory (lstm)," 2017.

[15] M. Sölch, J. Bayer, M. Ludersdorfer, and P. van der Smagt, "Variational inference for on-line anomaly detection in high-dimensional time series," *arXiv preprint arXiv:1602.07109*, 2016.

[16] S. Guo, K. Xu, R. Zhao, D. Gotz, H. Zha, and N. Cao, "Eventthread: Visual summarization and stage analysis of event sequence data," *IEEE TVCG*, vol. 24, no. 1, pp. 56–65, 2018.

[17] D. Gotz and H. Stavropoulos, "Decisionflow: Visual analytics for high-dimensional temporal event sequence data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1783–1792, 2014.

[18] S. Guo, Z. Jin, Q. Chen, D. Gotz, H. Zha, and N. Cao, "Visual anomaly detection in event sequence data," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 1125–1130.

[19] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[20] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *ACM Sigmod Record*, vol. 29, no. 2. ACM, 2000, pp. 427–438.

[21] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A geometric framework for unsupervised anomaly detection," in *Applications of Data Mining in Computer Security*. Springer, 2002, pp. 77–101.

[22] J. B. Cabrera, L. Lewis, and R. K. Mehra, "Detection and classification of intrusions and faults using sequences of system calls," *Acm sigmod record*, vol. 30, no. 4, pp. 25–34, 2001.

[23] J. Yang, W. Wang, and P. S. Yu, "Infominer: mining surprising periodic patterns," in *Proceedings of the 7th ACM SIGKDD*. ACM, 2001, pp. 395–400.

[24] R. Vinayakumar, K. Soman, and P. Poornachandran, "Long short-term memory based operation log anomaly detection," in *2017 ICACCI*. IEEE, 2017, pp. 236–242.

[25] T.-Y. Kim and S.-B. Cho, "Web traffic anomaly detection using c-lstm neural networks," *Expert Systems with Applications*, vol. 106, pp. 66–76, 2018.

[26] W. Lu, Y. Cheng, C. Xiao, S. Chang, S. Huang, B. Liang, and T. Huang, "Unsupervised sequential outlier detection with deep architectures," *IEEE TIP*, vol. 26, no. 9, pp. 4321–4330, 2017.

[27] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture on IE*, vol. 2, pp. 1–18, 2015.

[28] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng *et al.*, "Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications," in *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee, 2018, pp. 187–196.

[29] Y. Guo, W. Liao, Q. Wang, L. Yu, T. Ji, and P. Li, "Multidimensional time series anomaly detection: A gru-based gaussian mixture variational autoencoder approach," in *Asian Conference on Machine Learning*, 2018, pp. 97–112.

[30] L. Li, J. Yan, H. Wang, and Y. Jin, "Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[31] R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. AbuMallouh, "Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic," *IEEE sensors letters*, vol. 3, no. 1, pp. 1–4, 2018.

[32] Y. Guo, T. Ji, Q. Wang, L. Yu, G. Min, and P. Li, "Unsupervised anomaly detection in iot systems for smart cities," *IEEE Transactions on Network Science and Engineering*, 2020.

[33] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection for discrete sequences: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 823–839, 2012.

[34] N. Cao, C. Lin, Q. Zhu, Y.-R. Lin, X. Teng, and X. Wen, "Voila: Visual anomaly detection and monitoring with streaming spatiotemporal data," *IEEE TVCG*, vol. 24, no. 1, pp. 23–33, 2018.

[35] D. Thom, H. Bosch, S. Koch, M. Wörner, and T. Ertl, "Spatiotemporal anomaly detection through visual analysis of geolocated twitter messages," in *IEEE PacificVis*. IEEE, 2012, pp. 41–48.

[36] A. Bock, A. Pembroke, M. L. Mays, L. Rastaetter, T. Ropinski, and A. Ynnerman, "Visual verification of space weather ensemble simulations," in *IEEE SciVis*. IEEE, 2015, pp. 17–24.

[37] J. Chae, D. Thom, H. Bosch, Y. Jang, R. Maciejewski, D. S. Ebert, and T. Ertl, "Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition," in *IEEE VAST*. IEEE, 2012, pp. 143–152.

[38] J. Zhao, N. Cao, Z. Wen, Y. Song, Y.-R. Lin, and C. Collins, "Fluxflow: Visual analysis of anomalous information spreading on social media," *IEEE TVCG*, vol. 20, no. 12, pp. 1773–1782, 2014.

[39] M. Monroe, R. Lan, H. Lee, C. Plaisant, and B. Shneiderman, "Temporal event sequence simplification," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 12, pp. 2227–2236, 2013.

[40] J. Kehrer and H. Hauser, "Visualization and visual analysis of multifaceted scientific data: A survey," *IEEE TVCG*, vol. 19, no. 3, pp. 495–513, 2013.

[41] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts, "Visual comparison for information visualization," *Information Visualization*, vol. 10, no. 4, pp. 289–309, 2011.

[42] Y. Tu and H.-W. Shen, "Visualizing changes of hierarchical data using treemaps," *IEEE TVCG*, vol. 13, no. 6, pp. 1286–1293, 2007.

[43] J. Guerra-Gómez, M. L. Pack, C. Plaisant, and B. Shneiderman, "Visualizing change over time using dynamic hierarchies: Treeversity2 and the stemview," *IEEE TVCG*, vol. 19, no. 12, pp. 2566–2575, 2013.

[44] J. Kehrer, H. Piringer, W. Berger, and M. E. Gröller, "A model for structure-based comparison of many categories in small-multiple displays," *IEEE TVCG*, vol. 19, no. 12, pp. 2287–2296, 2013.

[45] J. Zhao, Z. Liu, M. Dontcheva, A. Hertzmann, and A. Wilson, "Matrixwave: Visual comparison of event sequence data," in *Proceedings of the ACM CHI*. ACM, 2015, pp. 259–268.

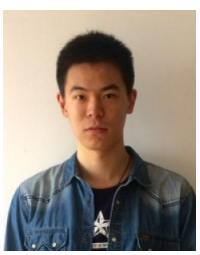
[46] F. Du, C. Plaisant, N. Spring, and B. Shneiderman, "Eventaction: Visual analytics for temporal event sequence recommendation," *Proceedings of the IEEE VAST*, 2016.

[47] J. L. Kolodner, "An introduction to case-based reasoning," *Artificial Intelligence Review*, vol. 6, no. 1, pp. 3–34, 1992.

[48] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," *Advances in neural information processing systems*, vol. 28, pp. 2575–2583, 2015.

[49] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[51] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," in *ACM SIGMOD Record*, vol. 29, no. 2. ACM, 2000, pp. 93–104.
[52] P. J. Rousseeuw and C. Croux, "Alternatives to the median absolute deviation," *Journal of the American Statistical association*, vol. 88, no. 424, pp. 1273–1283, 1993.
[53] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata, "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median," *Journal of Experimental Social Psychology*, vol. 49, no. 4, pp. 764–766, 2013.
[54] K. Wongsuphasawat and B. Shneiderman, "Finding comparable temporal categorical records: A similarity measure with an interactive visualization," in *2009 IEEE VAST*. IEEE, 2009, pp. 27–34.
[55] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE TPAMI*, vol. 17, no. 8, pp. 790–799, 1995.
[56] S. Guo, Z. Jin, D. Gotz, F. Du, H. Zha, and N. Cao, "Visual progression analysis of event sequence data," *IEEE TVCG*, vol. 25, no. 1, pp. 417–426, 2019.
[57] S. Guha, R. Rastogi, and K. Shim, "Cure: an efficient clustering algorithm for large databases," *ACM Sigmod record*, vol. 27, no. 2, pp. 73–84, 1998.
[58] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, "A sense of self for unix processes," in *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE, 1996, pp. 120–128.
[59] Y. Liao and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," *Computers & Security*, vol. 21, no. 5, pp. 439–448, 2002.
[60] Y. Xie and S.-Z. Yu, "A large-scale hidden semi-markov model for anomaly detection on user browsing behaviors," *IEEE/ACM Transactions on Networking*, vol. 17, no. 1, pp. 54–65, 2009.
[61] S. Budalakoti, A. N. Srivastava, R. Akella, and E. Turkov, "Anomaly detection in large sets of high-dimensional symbol sequences," 2006.
[62] Y. Qiao, X. Xin, Y. Bin, and S. Ge, "Anomaly intrusion detection method based on hmm," *Electronics Letters*, vol. 38, no. 13, pp. 663–664, 2002.
[63] X. Zhang, P. Fan, and Z. Zhu, "A new anomaly detection method based on hierarchical hmm," in *Proceedings of IEEE PDCAT*. IEEE, 2003, pp. 249–252.
[64] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific Data*, vol. 3, p. 160035, 2016.
[65] "Professors dataset," 2016, human-Computer Interaction Lab, University of Maryland. Retrieved from https://eventevent.github.io/.

Shunan Guo received Ph.D. degree in software engineering from East China Normal University, ShangHai, China. Her research interests include visual analytics and human-computer interaction, especially visual analytics approaches for temporal event sequences. For more information, please visit http://guoshunan.com/.



Zhuochen Jin received the BS degree in Computational Mathematics from the Zhejiang University, China, in 2017. He is currently working toward the PhD degree in the Intelligent Big Data Visualization (iDV^x) Lab, Tongji University. His research interests include artificial intelligence and data visualization.

Qing Chen received the B.Eng degree from the Department of Computer Science, Zhejiang University and the PhD degree from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology (HKUST). After receiving her PhD degree, she worked as a postdoc at Inria and Ecole Polytechnique. She is currently an assistant professor at Tongji University. Her research interests include information visualization, big data visual analytics, human-computer interaction, online education, visual storytelling, intelligent healthcare and design.

David Gotz received the Ph.D. degree in computer science from the University of North Carolina at Chapel Hill (UNC), Chapel Hill, NC, USA, in 2005. He is currently an Associate Professor of information science with the School of Information and Library Science, UNC. He directs the Visual Analytics and Communication Lab and conducts research on a range of topics at the intersection of data visualization, HCI, machine learning, and statistical analysis. He is also the Assistant Director for the Carolina Health Informatics Program and an Associate Member of the UNC Lineberger Comprehensive Cancer Center. He spent nearly a decade as a Research Scientist at the IBM T.J. Watson Research Center, New York, NY, USA before returning to join the UNC faculty in 2014.

Hongyuan Zha received the Ph.D. degree in scientific computing from Stanford University, in 1993. He has been working on information retrieval, machine learning applications, and numerical methods. He is currently a Professor with East China Normal University and with the School of Computational Science and Engineering, College of Computing, Georgia Institute of Technology. He was a recipient of the Second Prize of Leslie Fox Prize, in 1991, of the Institute of Mathematics and its Applications, the Outstanding Paper Awards of the 26th International Conference on Advances in Neural Information Processing Systems (NIPS 2013), and the Best Student Paper Award (advisor) of the 34th ACM SIGIR International Conference on Information Retrieval (SIGIR 2011). He served as an Associate Editor for the IEEE Transactions on Knowledge and Data Engineering.

Nan Cao received the Ph.D. degree in computer science and engineering from the Hong Kong University of Science and Technology (HKUST), Hong Kong, China, in 2012. He is currently a professor at Tongji University and the assistant dean of the Tongji College of Design and Innovation. He also directs the Tongji Intelligent Big Data Visualization Lab (iDV^x Lab) and conducts interdisciplinary research across multiple fields, including data visualization, human computer interaction, machine learning, and data mining. Before his PhD study at HKUST, he was a staff researcher at IBM China Research Lab, Beijing, China. He was a research staff member at the IBM T.J. Watson Research Center, New York, NY, USA before joining the Tongji faculty in 2016.