Visual Anomaly Detection in Event Sequence Data

Shunan Guo Department of Software Engineering East China Normal University Email: g.shunan@gmail.com

David Gotz

School of Information and Library Science University of North Carolina at Chapel Hill Email: gotz@unc.edu Zhuochen Jin College of Design and Innovation Tongji University Email: chjzcjames@gmail.com

Hongyuan Zha Department of Software Engineering East China Normal University Email: zha@cc.gatech.edu Qing Chen College of Design and Innovation Tongji University Email: jane.qing.chen@gmail.com

Nan Cao College of Design and Innovation Tongji University Email: nan.cao@gmail.com

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 applied to the analysis of 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 an unsupervised anomaly detection algorithm based on Variational AutoEncoders (VAE) to estimate underlying normal progressions for each given sequence represented as occurrence probabilities of events along the sequence progression. Events in violation of their occurrence probability are identified as abnormal. We also introduce a visualization system, EventThread3 (ET³), to support interactive exploration and interpretations of anomalies within the context of normal sequence progressions in the dataset through comprehensive one-to-many sequence comparison. Finally, we quantitatively evaluate the performance of our anomaly detection algorithm and demonstrate the effectiveness of our system through a case study.

Index Terms—data visualization; visual analytics; event sequence data; anomaly detection

I. INTRODUCTION

Anomaly detection is a common task for event sequence data analysis as it often contributes to the discovery of critical and actionable information [1]. Effective use of event sequence data can require identifying sequences that deviate from the typically occurring behavior [2]. 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 that the doctors can provide personalized care plans for similar patients in the future. A variety of techniques, including traditional statistical models [3], [4], supervised or semi-supervised approaches [5], and unsupervised methods [6] have been applied to detect anomalies in event sequences. However, due to the temporal characteristics of event sequence data and the black-box nature of machine learning models, it is challenging to interpret anomalous sequences once identified. For analysts to derive actionable insights, they must be able to understand how anomalies are different from "normal" sequences, which event or series of events characterize the anomaly, and which events suggest actions that could help avoid such cases in the future.

In this paper, we propose an unsupervised anomaly detection model for event sequence data that builds upon LSTMbased Variational AutoEncoders (VAE). VAE use a probabilistic encoder for modeling the distribution of the latent variables. Such probabilities give more principled criteria for identifying anomalies and do not require model-specific thresholds, which in turn, better facilitate objective judgments for deciding the boundary of anomalous sequences compared to other unsupervised algorithms. We train the model to learn a latent representation for each event sequence and identify anomalous sequences based on their deviation from the overall distribution. A mean sequence is computed from the reconstruction probabilities for each sequence detected as an anomaly, which shows the occurrence probabilities of events in normal circumstances, representing a corresponding "normal" sequence progression for the anomaly. For example, a patient having internal bleeding should normally be sent to emergency for surgery, thus the reconstruction probabilities shall identify surgical events with high probabilities after hospital admission. To facilitate interpretation of the anomaly, we also present an interactive graphical interface system to compare the anomaly sequence with a collection of normal sequences through oneto-many comparison mechanism and uncover their critical differences through a specifically designed comparison glyph.

II. RELATED WORK

Anomaly detection has been extensively studied over the past years [1]. Methods for anomaly detection can be broadly categorized into tensor-based algorithms [7], statisticsbased algorithms [8], classification-based algorithms [5], and neighbor-based or distance-based algorithms [9]. More recent work with deep learning-based anomaly detection (DAD) algorithms has been developed to pursue better performance. Types of DAD models include unsupervised (e.g., autoencoder, generative adversarial, variational), semi-supervised (e.g., reinforcement learning), hybrid (e.g., feature extractor + traditional algorithms) [10], and one-class neural networks [11]. In this work, we leverage VAE which can both deal with large volumes of unlabeled data, and identify anomalous patterns with probability measures [12]. Furthermore, we utilize the reconstruction probabilities to generate a output close to the original input sequence and provides latent feature vectors for each event sequence in the dataset.

Incorporating human domain knowledge through interaction can benefit the anomaly detection process, especially when the boundary between normality and abnormality is not precisely defined. To this end, researchers have developed many visual anomaly detection tools [13], [14]. This includes methods for the detection of anomalous user behaviors from sequence data [15]. Chae et al. [16] applied traditional control chart methods together with seasonal trend decomposition to extract outliers. Thom et al. [14] introduced a visual analysis system to monitor for anomalous bursts of keywords. More recently, FluxFlow [17] was developed to reveal and analyze anomalous information processes in social media. Although these systems are often designed to help detect anomalous points, few approaches focus on identifying anomalous sequences or on the comparison between the detected outliers and "normal" sequences. To enhance the interpretability of the analyzed results, in ET³, we provide an interactive oneto-many comparison between the anomalous sequence and normal progressions.

III. VAE-BASED ANOMALY DETECTION

A. LSTM-Based Variational AutoEncoder

We first introduce the structure of the Sequence-to-Sequence VAE model. The model contains two modules: the VAE encoder and the VAE decoder. Both modules are designed using Recurrent Neural Networks 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 $\{X = x_i\}_{i=1}^n$ into a low-dimensional latent feature vector that describes a sequential distribution of events occurring in the sequence(as shown in Fig. 1(1)). In this input, n is the length of the sequence and $x_i \in \{0,1\}^{|E|}$ is the multi-hot encoding of the events in event set E occurring in the *i*-th time step. Each coordinate represents an event type, which is marked 1 if the corresponding event occurs in the *i*-th time step, or 0 otherwise. After feeding the multihot vectors into the corresponding layer of RNN, the state of the entire sequence is extracted and represented in the hidden state vector h_{enc} of the last layer, which is denoted as $h_{enc} = encoder(X)$. The hidden state vector h_{enc} is projected into vector μ and δ 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 z by randomly sampling from the distribution and use 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 z. Specifically, 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 X' = decoder(z), where X' is a sequence of probability distributions denoted as $X' = \{x'_i\}_{i=1}^n$, and the element $x'_{i,j}$ in $x'_i \in R^{|E|}$ represents the occurrence probability of j-th event at the i-th time step.

Training Process. We train the model with a goal of narrowing the gap between the original input sequence and its reconstruction, which can be formally defined as minimizing the following loss function:

$$L = L_r + w_{kl} \cdot L_{kl} \tag{1}$$

$$L_r = \frac{\sum_{i=1}^{n} \sum_{j=1}^{|E|} (w_{e_j} x_{ij} log(x'_{ij}) + (1 - x_{ij}) log(1 - x'_{ij}))}{-n} \quad (2)$$

$$L_{kl} = -\frac{1}{M_z} \sum_{i=1}^{M_z} (1 + \log(\sigma_i^2) - \mu_i^2 - \sigma_i^2)$$
(3)

The first term L_r is the reconstruction loss which calculates the weighted cross entropy between $x_{i,j}$ and $x'_{i,j}$, indicating an event-level difference between the reconstruction and the original input with respect to the *j*-th event at the *i*-th time step. In particular, a parameter $w_{e_j} = 1/log(n_j)$ is introduced to reduce the marginal importance of high-frequency events so as to address the issue of skewed dataset, where n_j is the number of occurrences for event e_j . The second term L_{kl} is the Kullback-Leibler Divergence Loss which estimates a distribution-level difference between the distribution of the latent vector z and a normal distribution N(0, 1), where M_z is the dimension of the latent vector z. These two terms are balanced with a parameter w_{kl} .

Parameter Settings. Both the encoder and decoder employ LSTM units [18] 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. Moreover, we optimize the loss function with the Adam optimizer [19] with training data batch size of 80 for each training step. We train the model on an Nvidia Tesla K80 graphics card. Each training epoch takes approximately 10.5 seconds on average.

Anomalous Sequence Detection. After training the model, we employ the latent vector z of each input sequence to detect anomalous sequences in the dataset to calculate the degree of anomaly for each sequence in the latent space using the Local Outlier Factor (LOF) [20](as shown in Fig. 1(2)). 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.

B. Anomalous Event Analysis

To facilitate the interpretation of sequence anomalies, we further identify anomalous events that contribute to sequence abnormality by analyzing the reconstruction probabilities(as shown in Fig. 1(3)). As we assumed that the majority of the sequences are normal, the reconstruction probabilities shall be similar to the normal progression of sequences, and the training objective ensures that the reconstruction probabilities are also similar to the original input sequence. Thus, the



Fig. 1. 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, and (3) anomalous event detection based on the reconstruction of the input sequence.

reconstruction probabilities of the anomalous sequences can be used to infer a *mean sequence* that represents an expected "normal" progression for the anomalous sequence. From this, we can identify the anomaly events within the anomalous sequence that deviate from the expected normal progression.

We categorize the anomalous events into missing events (noted as x^{mis}) and redundant events (noted as x^{red}), which represent the cases where events show high occurrence probabilities in the reconstruction but do not appear in the sequence, and events that exist in the anomalous sequence but are not expected to occur, respectively. Based on this intuition, we calculate the anomaly scores for missing events and redundant events with $Pr(X = x^{mis})$ and $1 - Pr((X = x^{red}))$, respectively, where Pr(X = x) indicates the occurrence probability of the corresponding event derived from the reconstruction. Consequently, events with an anomaly level higher than a userdefined threshold are identified as anomalous. The threshold is by default set as 0.6, which can be adjusted by users during an analysis via the visualization module.

IV. VISUALIZATION

A. Design Tasks

We formulated a set of design tasks to solve the key challenges in visually analyzing anomalies in event sequence.

- **T1 Provide an overview of the analysis scope.** To help users find anomalous sequences of interest within a collection of anomalous sequences, the system should provide an overview of all sequences detected as anomalies and illustrate their level of abnormality.
- **T2 Emphasize anomalous events within the sequence.** To help quickly explore complex event sequences and uncover the reason behind an abnormality, the visualization should be designed to highlight key events that are suspicious of being anomalous.
- **T3** Facilitate result interpretation in context. The designed visualization should help users effectively analyze the detected anomalies within the context of the entire training set, to uncover the difference between abnormal and normal sequence progressions and facilitate reasoning about the analyzed result.
- **T4** Support sequence exploration at multiple levels of granularity. Applying different levels of aggregation for a group of sequences can result in distinct interpretations of the result. To support more accurate findings, the system should support the exploration of normal sequences at different levels of granularity.

Guided by the tasks above, ET³ incorporates seven key views to visually analyze the anomalous sequences (Fig. 2), which includes (1) an anomaly overview providing an overview of all detected anomalies from which users can choose for subsequent analysis (T1); (2) a similarity view, showing the distribution of normal sequences as regard to their similarities and the selected anomaly (T1); (3) a reconstruction view presenting the occurrence probabilities of the events in each time slot (T3); (4) a flow overview that aggregates the flow of normal sequences with the evolution of the selected anomaly overlaid at the top to show differences (T3); (5) a comparison view with three variants: (5a) sequence comparison view, (5b) flow comparison view, and (5c) summarization view that separates the flow of the anomalous sequence from the normal sequences with *comparison glyphs* (Fig. 2(a)) emphazing the event difference (T2) and displays the summarization of normal sequences in multi-level granularity (T4); (6) an anomalous record view and (7) a similar record list displaying low-level details of the selected anomaly and normal sequences, respectively. The system is incorporated with rich interactions to support exploratory analysis.

B. Interactive Anomalous Event Analysis

Our system is designed to facilitate interpretation of the selected anomaly in the context of the progression of normal sequences (T3) via interactive one-to-many visual comparison. The comparison view is vertically divided into three regions: an *anomalous sequence* at the top, a group of *comparison glyphs* in the middle, and a summarization of normal sequences at the bottom.

1) Anomalous Sequence: The selected anomalous sequence is displayed using a line of rectangular nodes ordered by time of occurrence. To deal with the issue of event co-occurrence and avoid event overlap, we display the sequence with a visual technique introduced in [21]. Specifically, concurrent events are grouped into treemaps at each time slot, and all event nodes are color-coded according to the type of anomaly. Event nodes are spaced with equal distance and connected with duration bars to reveal the span of time. The time span between events is proportional to the duration bar.

2) One-to-Many Sequence Comparison: Our system incorporates a one-to-many sequence comparison mechanism, which allows users to validate the anomalies detected by the model by comparing the anomalous sequence with a collection of similar sequences from the normal group. This aims to help users establish confidence in the analysis result based on evidence in the dataset. The comparative analysis consists of two steps: sequence alignment and support rate calculation. In the first step, we employ a sequence alignment technique introduced in [22] to semantically map each normal sequence to the focal anomaly based on Dynamic Time Warping (DTW) [23] to address the issues of variable sequence length and progression rate for a more precise comparison of events. After sequence alignment, we compare events occurring in each time slot to calculate a support rate for each anomalous event identified in Sec. III-B. Intuitively, the support rate represents the proportion of normal sequences that "support" the corresponding event to be abnormal. More specifically,



Fig. 2. The user interface of ET³ consists of seven key views with a comparison glyph designed to support comparison-based visual anomaly detection.

the support rate of a missing event x^{mis} is the proportions of sequences that include x^{mis} in the corresponding time slot, while the support rate for x^{red} is the contrary.

3) Comparison Glyph: To facilitate visual comparison, we design a comparison glyph (Fig. 2(a)) that highlights the anomalous events in each time slot. We encode four critical variables to help quickly identify problematic time slots and events that need further inspection: the overall abnormality of the time slot, the abnormality of each event, the type of anomaly, and the support rate for each anomalous event. Specifically, each circle inside the glyph represents an anomalous event. The size of each internal circle(Fig. 2(a1)) indicates the anomaly score of the corresponding event derived from Sec. III-B, and the size of outer circle (Fig. 2(a2)) represents the overall abnormality at the corresponding time slot. The type of abnormality (e.g., missing event or redundant event) is distinguished with different colors, consistent with other views. The support rate of each anomalous event (Sec. IV-B2) is encoded with color saturation.

Updating with user feedback. To leverage analyst domain knowledge, the system allows users to interactively tweak the anomalous events displayed in the comparison glyphs. As shown in Fig. 2(b), users can tune the thresholds for the anomaly score and support rate that determine the conditions at which an event is identified as anomalous. Moreover, when users select a subgroup of normal sequences during the analysis, the comparison glyphs will also be updated simultaneously to reflect the support rate within the subgroup.

4) Multi-granular Sequence Aggregation: To support more comprehensive one-to-many sequence comparisons, the design provides three coordinated comparison views (Fig. 2(5a-c)). The views support comparison at different levels of aggregation (**T4**), and transitions allow users to move smoothly from one view to another.

The *sequence comparison view* displays the sequences of normal records individually, which aims to support sequence-to-sequence level comparison and efficient access to the raw data. As shown in Fig. 2(5a), the normal sequences are displayed in a scrollable list with a consistent encoding

schema as the anomalous sequence, ranked from top to bottom according to the degree of similarity. Users can select any individual sequence to update the comparison glyphs with their differences during the analysis.

The flow comparison view (Fig. 2(5b)) provides a progression-level summarization on all normal sequences by aggregating them into a flow-based visualization. This view aims to incorporate confidence of abnormality for anomalous events by comparing the anomalous sequence with subgroups of sequences having particular progression patterns. Specifically, identical events in each time slot are grouped into nodes, and the transition paths among events in adjacent time slots are merged into links. The height of each node represents the population (weighted by event co-occurrence) having the event at the corresponding time slot, with the exact number displayed in a label to the left side of each node. Event nodes are connected with links to represent a sequence path from one event to another. Each link is consist of a duration bar and a connection line. The height of the duration bar shows the proportion of the population corresponding to the link, while the width indicates the average time gap between events. Users can select any node or link to highlight the progression pattern and narrow the comparison of a specific subgroup.

In the summarization view (Fig. 2(5c)), nodes in each time slot are further aggregated into a more compact form, illustrating the highest-level summarization of the distribution of events. This view aims to support a comparison of the anomalous sequence against the overall progression of the entire set of similar records. We encode the summarized sequences in a way similar to the anomalous sequence, with the only difference that the size of each inner rectangle represents the size of the population. To allow for the analysis and exploration on a higher-level summarization of progression stages, we also leverage a recently proposed progression analysis technique [22] to segment the anomalous sequence into different stages. Stages are marked with line segments under the identifier of the time slots (Fig. 2(c)). Users can click on a stage identifier to merge or expand all visual elements in the main panel that align to the corresponding time slots.



Fig. 3. Performance evaluation results of our VAE-based algorithm (VA) in comparison with two baseline methods (kNN, HMM).

C. Other Views

The system also includes several contextual views to display auxiliary information and provide access to raw data. The anomaly overview (Fig. 2(1)) shows the multidimensional scaling (MDS) projection of the latent vector z on a colored contour map. Each anomalous sequence is represented as a circle with the size indicating the LOF score, and the color saturation indicating the sequence length, so as to help analysts choose sequences of high anomaly degree for subsequent analysis. The similarity distribution view (Fig. 2(2)) displays the distribution of all normal sequences in the dataset based on their similarity to the selected anomaly, which aims to help users select a proper group of normal sequences for comparative analysis. The reconstruction probabilities (given by Equation III-A) of the selected anomaly are shown in the reconstruction view (Fig. 2(3)) with the intent of providing an overview of the occurrence probabilities of events for each time slot. The reconstruction probabilities are shown as a line of circle packings arranged in time order, with the size of each circle shows the value of probability, and the color indicates different anomaly types (consistent with other views). The anomalous record view (Fig. 2(6)) and the similar record list (Fig. 2(7)) provide access to raw event sequence data of the anomalous sequence and similar sequences. These low-level details provide detailed evidence to support interpretation.

V. EVALUATION

We demonstrate the effectiveness of ET³'s analytical model and the usefulness of visualization system through a quantitative evaluation and a case study.

A. Quantitative Evaluation

We compare the performance of our VAE-based anomaly detection algorithm (denoted as VA) with two baseline methods using an intrusion detection dataset, snd-cert [24]. The dataset consists of sequences of operating system calls that are labeled in terms of the system state (i.e., normal or hacked) when running these operations.

We select two representative baseline methods under the categories of kernel-based and Markovian anomaly detection techniques: Nearest Neighbor (kNN) [25] and Hidden Markov Model (HMM) [26]. Both methods have been shown efficient for detecting anomalies in event sequence data in previous research [27]–[29]. More specifically, the longest common subsequence (LCS) was used as the distance metric in kNN and standard information retrieval metrics (precision, recall, and ROC) to evaluate model performance.

Evaluation Results. Our algorithm outperforms the baseline methods as shown in Fig 3. The ROC plot (Fig 3(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 3(b)) shows that VA had overall higher precision than the baseline methods. The results indicate that our approach can produce a higher quality set of suspicious sequences when compared to the baseline algorithms. Using the designed visualization, the system can further support the interpretation of detected anomalies.

B. Case Study

We applied ET^3 to MIMIC [30], a publicly accessible critical care database with de-identified electronic health records for 46,520 patients with 12,487 event types in total. 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 10,183 patients who were diagnosed with cardiovascular diseases to produce a more homogeneous set of sequence progressions for training. Four cardiologists (E1-E4, 5-8 years of domain experience each) were invited to participate in our study. Prior to the study, we asked the doctors about expected patterns of anomaly and they expressed interests in exploring anomalous medical usage within the follow-up lab test results, based on which we extracted 87 types of prescriptions and lab events, which were further distinguished using different shape in the visualization.

404 anomalous sequences were detected for subsequent analysis. The experts chose a patient who was far away from the main cluster with a relatively high anomaly score from the overview(Fig. 2(1)), and then retrieved 105 similar patients with the distance to the mean sequence under 0.2 for subsequent analysis (Fig. 2(2)). As they quickly scrolled the flow overview back and forth to get a big picture of the major sequence progression paths, they found that the treatment plans for the patients were very similar with regular use of Phenytoin and Insulin (Fig. 4(a)), and speculated that all of these patients were suffering from epilepsy and diabetes. While most events in the progression of anomaly were in agreement with the major trend, several exceptional events appeared in the second and the third time slots of the selected anomaly (Fig. 2(g)). By splitting the sequence of the anomalous patient from others, the comparison glyphs uncovered suspicious events at these time slot. The experts noticed an abnormal lab event with a 0.59 anomaly score and 100% support rate, CK-MB (Fig. 4(i)). "This is a critical indicator for myocardial infarction", E1 said. The experts also found an event that was continuously missing in several time slots throughout the entire progression, Hydralazine (Fig. 4(b)). "This drug is mainly applied to patients with chronic heart failure" explained E2. "This may imply different causes of epilepsy. Both heart diseases can potentially cause epileptic seizure." The experts then explored the redundant medicines highlighted in the comparison glyphs to investigate the differences in the treatment plan and surprisingly found no medicines aimed directly at curing myocardial infarction. "This is unusual," E1 said, "It seems the patient



Fig. 4. The anomaly detection result of MIMIC dataset. The system identified major progression paths (a-b), from which the sequence anomaly deviates in (i) an anomalous lab test result and the (ii) misuse of a prescription drug.

was treated as a regular epileptic patient." They also found a type of medicine, Pheonobarbital(Fig. 4(ii)) used only by the anomalous patient. "I believe Pheonobarbital is mainly used for neonatal and childhood seizures according to guidelines," said E3. "It is rare to see this drug prescribed for a 69-year-old man." E4 found this finding especially useful, as he commented: "It is a potential drug of abuse. Long-time usage can result in physical dependence, thus should be strictly controlled. I feel this system has great potential to be applied to monitor drug misuse."

VI. CONCLUSION

We have presented ET^3 , a visual analysis technique designed to support visual anomaly detection in event sequence data. ET^3 incorporates an unsupervised VAE-based anomaly detection model to identify anomalous sequences and events in an interpretable manner, and a visualization system with multiple coordinated views and rich interactions is provided to facilitate interpretation via one-to-many sequence comparison. We evaluate the effectiveness and usefulness of ET^3 through a quantitative comparison of the performance of our proposed algorithm and a case study. The study results illustrate the strengths of ET^3 and shed light on several directions for future work, including enabling integrating an associative measurement for event abnormality that considers both anomaly score and support rate, and supporting the analysis of multiple anomalous sequences simultaneously.

ACKNOWLEDGMENT

The authors thank all medical experts for their participation in the case study. This research was supported by NSFC Grant 61602306, Fundamental Research Funds for the Central Universities, the National Grants for the Thousand Young Talents in China, the NSFC Grant 61672231 and NSFC-Zhejiang Joint Fund under Grant U1609220. Nan Cao is the corresponding author.

REFERENCES

- V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Computing surveys, vol. 41, no. 3, p. 15, 2009.
- [2] 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.
- [3] 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, 2005, pp. 270–276.

- [4] 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.
- [5] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *IEEE ICDM*, 2008, pp. 413–422.
- [6] G. Münz, S. Li, and G. Carle, "Traffic anomaly detection using k-means clustering," in *GI/ITG Workshop MMBnet*, 2007, pp. 13–14.
- [7] H. Chen, S. Zhang, W. Chen, H. Mei, J. Zhang, A. Mercer, R. Liang, and H. Qu, "Uncertainty-aware multidimensional ensemble data visualization and exploration," *IEEE TVCG*, vol. 21, no. 9, pp. 1072–1086, 2015.
- [8] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. John wiley & sons, 2005, vol. 589.
- [9] 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, 2000, pp. 93–104.
- [10] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "Highdimensional and large-scale anomaly detection using a linear one-class svm with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016.
- [11] R. Chalapathy, A. K. Menon, and S. Chawla, "Anomaly detection using one-class neural networks," arXiv preprint arXiv:1802.06360, 2018.
- [12] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture on IE*, vol. 2, pp. 1– 18, 2015.
- [13] 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.
- [14] D. Thom, H. Bosch, S. Koch, M. Wörner, and T. Ertl, "Spatiotemporal anomaly detection through visual analysis of geolocated twitter messages," in *IEEE PacificVis*, 2012, pp. 41–48.
- [15] A. Bock, A. Pembroke, M. L. Mays, L. Rastaetter, T. Ropinski, and A. Ynnerman, "Visual verification of space weather ensemble simulations," in *IEEE SciVis*, 2015, pp. 17–24.
- [16] 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*, 2012, pp. 143–152.
- [17] 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.
- [18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
- [20] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1641–1650, 2003.
- [21] Z. Jin, J. Yang, S. Cui, D. Gotz, J. Sun, and N. Cao, "Carepre: An intelligent clinical decision assistance system," arXiv preprint arXiv:1811.02218, 2018.
- [22] 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.
- [23] M. Müller, "Dynamic time warping," Information Retrieval for Music and Motion, pp. 69–84, 2007.
- [24] 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*, 1996, pp. 120–128.
- [25] 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.
- [26] 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.
- [27] S. Budalakoti, A. N. Srivastava, R. Akella, and E. Turkov, "Anomaly detection in large sets of high-dimensional symbol sequences," 2006.
- [28] 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.
- [29] X. Zhang, P. Fan, and Z. Zhu, "A new anomaly detection method based on hierarchical hmm," in *Proceedings of IEEE PDCAT*, 2003, pp. 249– 252.
- [30] 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.