

Time Series Anomaly Detection with Variational Autoencoders

Chunkai Zhang

Department of Computer Science and Technology
Harbin Institute of Technology, Shenzhen
Shenzhen, China
ckzhang812@gmail.com

Yingyang Chen

Department of Computer Science and Technology
Harbin Institute of Technology, Shenzhen
Shenzhen, China
yingyang_chen@163.com

Abstract—Anomaly detection is a very worthwhile question. However, developing effective anomaly detection methods for complex and large data remains a challenge which needs to compare the similarity of each time series based on the idea that anomaly is less and different. There are already some deep learning models based on GAN for anomaly detection that demonstrate validity and accuracy on time series data sets. In this paper, we propose an unsupervised model-based anomaly detection named LVEAD, which assumes that the anomalies are objects that do not fit perfectly with the model. For better handling the time series, we use the LSTM model as the encoder and decoder part of the VAE model. Considering to better distinguish the normal and anomaly data, we train a re-encoder model to the latent space with generated data. Experimental results of several benchmarks show that our method outperforms state-of-the-art anomaly detection techniques and achieves, on average, 5% improvements in AUC.

Index Terms—Anomaly detection, Time series, Deep neural networks

I. INTRODUCTION

Anomaly detection is widely used in many fields, such as network communication to find abnormal information flow [1], financial field [2] like credit card fraud, industrial field for sensor anomaly [3], medical imaging like optical coherence tomography (OCT) [4] and time series where a rich body of literature proposed [5]–[8]. Anomaly detection is to find different patterns in the data which often contain important information, and these patterns are not caused by random deviations. In time series, anomaly sequences are defined as sub-sequences that exists for a period of time in one long time series, which are different from other sub-sequences. It also compensates for the limitations of point anomalies. An example of the anomaly sequence is shown in Fig.1.

Compared with the classification problem [9] has a certain number of categories, the abnormal data represent different from the normal and are difficult to collect and label all the anomalies. In addition, the complexity of the anomalies causes great difficulty in classification. Generally, most of anomaly detection methods are based on the similarity to determine the degree of abnormal data, and the time complexity is $O(N^2)$ [10], [11].

In the face of high-dimensional data and relatively large data volume, the traditional anomaly detection method has high time complexity. However, in reality, anomaly detection is not

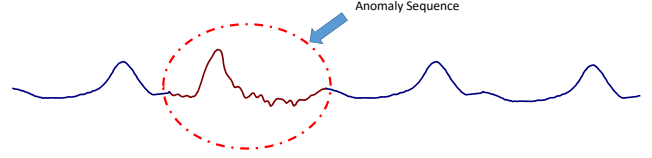


Fig. 1. An anomaly sequence in one time series.

a simple two-category classification, which the anomalies are more unknown and not always minority. We don't know the percentage of anomalies in advance. So the methods found by similarity comparison are not necessarily true anomalies. If the normal and abnormal are distinguished from the feature extraction, the imbalance of the training sample will result in an unsatisfactory algorithm.

In this paper, we propose an unsupervised model-based deep learning anomaly detection method based on the assumption that those do not fit perfectly with the model are anomalies. Our proposed method uses the VAE-reEncoder architecture. We use long-term and short-term memory (LSTM) to model the normal time series under the variational auto-encoder model. After generating new data, we encode it to obtain new potential vectors, and optimize the reconstruction error and potential vector error. During the test, mixed data of abnormal and normal are input, and the new data of abnormal data under normal circumstances are generated for comparison. Meanwhile, the features of latent spatial vector representation are used to compare the newly generated data. We named our approach LSTM-VAE-reEncoder Anomaly Detection(LVEAD).

The contribution of this paper can be summarized as follows.

- (1) We design an unsupervised Variational Autoencoder re-encoder with LSTM encoder and decoder that can perform anomaly detection effectively on high dimensional time series;
- (2) A simple and effective algorithmic method that can be reproduced at any time.
- (3) We design a generic framework for anomaly detection in time series data and the experimental results on several benchmark data show that the proposed method outperforms baseline models in AUC;

The rest of this paper is organized as follows. Section II reviews the related work. Section III introduces the LVEAD framework and describes the training and architecture of the network. Section IV shows the experimental results that our method outperforms other state-of-the-art method on several benchmarks.

II. RELATED WORK

The traditional anomaly detection [10], [12]–[14] uses similarity method to reduce dimensionality and distinguish between normal and anomaly data, which based on the assumption that anomalies are minority class and different. The result obtained by this method in the real data is not necessarily a true anomaly.

More recent attention in the literature has been focused on model-based anomaly detection [15]–[17]. Joshi et al. [18] studied the Hidden Markov Model (HMM) for anomaly detection, which built a Markov model after extracting features and calculated the anomaly probability from the state sequence generated by the model. Ahmad et al. [19] proposed Hierarchical Temporal Memory (HTM) that derived from neuroscience that simulates spatial and temporal patterns for model in streaming data. As for Autoregressive Integrated Moving Average model (ARIMA) [20] creates a model by the correlation among data for non-stationary time series and the prediction result of the model is judged anomaly by the threshold.

In more recent years, one of the most influential accounts of anomaly detection is using deep learning. Malhotra et al. [21] proposed a prediction-based model based on LSTMs and used the distribution of the prediction errors to compute anomaly scores. However, this approach is not suitable for time series affected by external factors not captured by sensors, making them unpredictable. Bayer and Osendorfer [22] used variational inference and RNNs to model time series data and introduced stochastic recurrent networks (STORNs), which were subsequently applied to anomaly detection in robot time series data [23]. An and Cho [24] proposed a method based on a VAE and introduced a novel probabilistic anomaly score that takes into account the variability of the data with the reconstruction probability. Seebock et al. [25] trained an Autoencoder and utilized a one-class SVM [26] on the compressed latent space to distinguish between normal and anomaly patches. Schlegl et al. [4] presented the AnoGAN framework, in which they create a rich generative model of normal sample using a GAN. Assuming that the model cannot properly reconstruct abnormal samples, they classify query samples as either anomalous or normal by trying to optimize the latent code based on a novel mapping score, effectively also leading to a delineation of the anomalous region in the input data. Houssam Zenati et al. [27] proposed Adversarial Learned Anomaly Detection (ALAD) based on bi-directional GANs, that derives adversarial learned features for the anomaly detection task based on AnoGAN and uses reconstruction errors based on these adversarial learned features to determine if a data sample is anomalous.

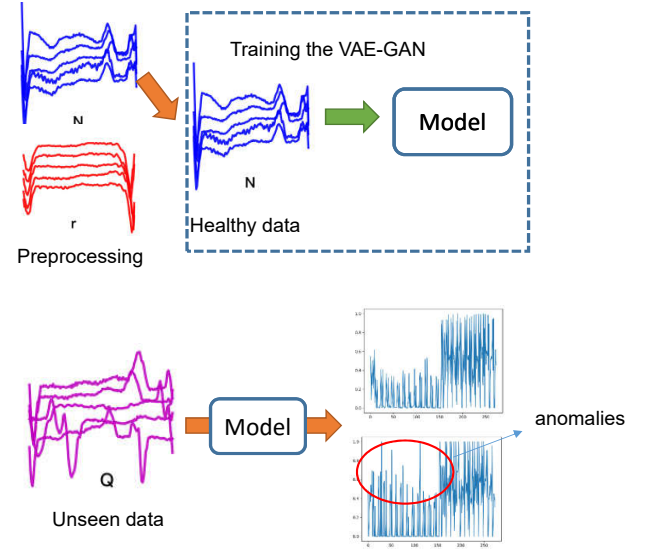


Fig. 2. Anomaly detection flow chart. The upper block is the training phase and the below blow is the testing phase.

Our proposed LVEAD method is more suitable to time series data for we use bidirectional bow-tie LSTM to encode and decode the input to get better representation under dimensionality reduction. In contrast to variety GANs, which simply lets the data generated by the generator fool the discriminator, so unreasonable data situation may occur when generating data. Whereas, the variation generated by VAE is to use existing data to generate potential vector under the encoder, which is subject to Gaussian distribution and can well retain the characteristics of the original data, so that the generated data will be more reasonable and accurate. So we combine the advantages of both, using the VAE-reEncoder model.

III. PROPOSED METHOD

The pipeline of the model is shown in Fig. 2. Our model is based on the principle of training the model with normal data sets and using mixed data sets when testing. When there is abnormal data, the model generates data for that data under normal circumstances. Moreover, we encode the newly generated data to get the new latent vector again. For normal data, the potential space generated by data coding is similar to the potential space generated by the first coding, while the potential space generated by abnormal data changes greatly. Therefore, by comparing the similarity between the generated data and the original data and the difference between the latent layer vectors, we can conclude that our model will only produce data similar to the normal time series, and the data generated by the abnormal data have a large difference, while the data generated by the normal data have a small difference.

A. Model

The structure of LSTM-VAE-reEncoder is shown in the Fig.3, including two encoders and one decoder. The model consists of three parts. The first part is the generation network.

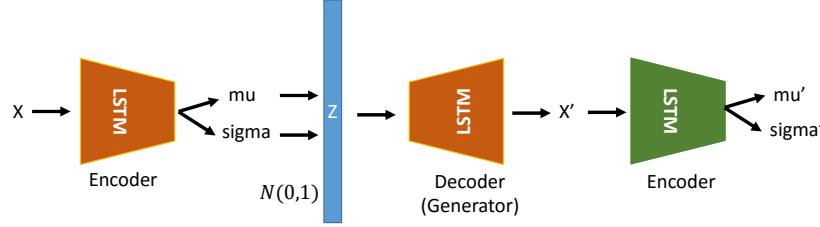


Fig. 3. The network structure of LVEAD. Two orange blocks are the encoder and decoder layer of VAE, and the green block is the re-Encoder layer.

We use bidirectional LSTM-VAE. Because the LSTM model is more suitable for processing time series data, we use the bow-tie model to remove noise to some extent when encoding. And for time series data, we can't guarantee that the importance weights between time points are different, so using bidirectional LSTM can make each time point equal, he does LSTM from left to right, then LSTM from right to left. And then combine the two results into the following formula, where o_t is the output, \vec{S}_t^1 and \vec{S}_t^2 represent the hidden layer from the hidden layer to the back and the back to the front. We used a bow-tie model to remove noise to some extent when encoding. For the input data, the reconstructed data x can be obtained by the first part passing through the decoder.

$$o_t = \text{softmax} \left(V^* \left[\vec{S}_t^1; \vec{S}_t^2 \right] \right) \quad (1)$$

$$\vec{S}_t^1 = f \left(\vec{U}^1 * X_t + \vec{W}^1 * S_{t-1} + \vec{b}^1 \right) \quad (2)$$

$$\vec{S}_t^2 = f \left(\vec{U}^2 * X_t + \vec{W}^2 * S_{t-1} + \vec{b}^2 \right) \quad (3)$$

The second part is the discriminator D. For the original time series to be true, the reconstructed data is judged to be false, so that the difference between the reconstructed data and the original data is continuously optimized. The third part is to encode the reconstructed time series to obtain the potential vector \hat{Z} of the reconstructed data, and compare the difference between the two potential vectors. There have been papers demonstrating that similar potential spaces can produce visually similar high-dimensional data. By inverse mapping the high-dimensional image to an additional encoder network of lower dimensional potential space and learning through the model. So we explore the reasoning of the model by using the latent vector representation. In the training process, the normal data distribution can generate a unique representation under ideal conditions, so that unknown abnormal data samples can be found at the data level.

B. Training Objective

In the training process, we train the model with normal samples, so the encoder, decoder and reconstruction encoder are suitable for normal samples. According to the model, we define the loss function as three parts, so that the generator is optimized based on the context information of the input data.

The first part is the reconstruction error function, which is used to narrow the difference between the original time series and the reconstructed data, so that the generator optimizes according to the context information of the input data.

$$L_{rec} = E_{x \sim p_X} \|x - G(x)\|_1 \quad (4)$$

The second part is the feature matching error of the discriminator for the antagonistic learning. The feature matching is proved to reduce the instability of the training. We update G based on the internal representation of D, and calculate the L2 distance between the original feature representation and the generated image, respectively.

$$L_{adv} = E_{x \sim p_X} |f(x) - E_{x \sim p_X} f(G(x))|_2 \quad (5)$$

The third part is the potential vector error. The above two loss functions can force the generator to learn reliable context-correlated data. We remap the generated time series to the potential space and compare the loss of the two coding networks.

$$L_{enc} = E_{x \sim p_X} \|G_E(x) - E(G(x))\|_1 \quad (6)$$

For the model, the entire loss function can be expressed as

$$L = w_{adv} L_{adv} + w_{rec} L_{rec} + w_{enc} L_{enc} \quad (7)$$

Where w_{adv} , w_{rec} , w_{enc} are weighting parameters that adjust the effect of the individual loss function on the overall objective function. For anomalous inputs, it will not minimize the distance between the input and the image generated in the feature space, as the G and E networks are optimized only for normal samples.

In the test phase, when the model inputs an exception sample, the difference between the generated reconstruction sequence and the regenerated potential vector and the original data is huge, so we can set the overall anomaly score by this $S = \{s_i : A(\hat{x}_i), \hat{x}_i \in \hat{D}\}$, $A(\hat{x}_i)$ is the abnormal score of each sample, and the calculation formula is as follows.

$$A(x) = \alpha \|G_E(x) - E(G(x))\|_1 + \beta \|G_E(x) - E(G(x))\|_1 \quad (8)$$

where α and β are the parameters to constrain the penalty term, and $\alpha + \beta = 1, \alpha > 0, \beta > 0$.

TABLE I
THE DETAILS OF BENCHMARK DATA SETS. THE AR REPRESENTS FOR THE ANOMALY RATIO.

Dataset	Data type	AR	Length	Size
KDD99	Network intrusion	0.15	121	494021
Arrhythmia	Sensor	0.20	274	452
ItalyPowerDemand	Sensor	0.49	24	1096
TwoLeadECG	Sensor	0.49	82	1162
GunPointAgeSpan	Motion	0.49	150	450
MoteStrain	Sensor	0.46	84	1452
ToeSegmentation2	Motion	0.25	343	166
Herring	Image	0.46	512	128
Wafer	Sensor	0.11	152	7164
ECGFiveDays	Sensor	0.20	136	884

Finally, the anomaly score needs to be normalized to the interval of $[0, 1]$ as the formula. The more abnormal the data, the larger the abnormal score is close to 1. By setting the threshold φ , as long as $A(x) > \varphi$, we consider the sample to be abnormal.

$$s'_i = \frac{s_i - \min(S)}{\max(S) - \min(S)} \quad (9)$$

IV. EXPERIMENTAL RESULT

This section introduces the different types of benchmark data sets we used and the baseline methods we compared to demonstrate the effectiveness of our proposed method in unsupervised anomaly detection.

A. Data sets

To illustrate the effectiveness of our proposed method, four types of time series data, Sensor, Motion, Image and Network intrusion, is got from UCR public data set [28] and UCI public data set [29]. The details can be seen in Table I. The data we select is relatively large regardless of the length of the sequence or the size of the data set. Due to the high proportion of outliers in the KDD99 dataset, "normal" data is considered abnormal. For other data sets, we choose the minority class as anomaly class. We split 20% of the data as test data.

To validate our method, we consider AnoGAN [4], ALAD [27], MLP-VAE [24] and Isolation Forest [30] as the baseline algorithms. Isolation Forest is a state-of-the-art traditional ensemble anomaly detection method which uses random hyper-plane to cut the data space. In contrast, AnoGAN uses GAN's model to generate data that compares pixel-level differences between raw and generated data. And ALAD is an improvement based on above method with additional discriminators to improve the encoder. As for MLP-VAE, it uses reconstruction probability from the variational Autoencoder. We set parameter $w = 1$ for training phase. Rather than precision or recall, AUC (Area Under the ROC Curve) [31] is the common metrics to measure performance in anomaly detection.

B. Performance Evaluation

We performed experiments on accuracy on seven data sets and used AUC as the criterion. From the Table II we can see that our method has been greatly improved on 8 out of 10 data

sets, increased by at least 0.8% on most data sets, and the most improved data set is TwoLeadECG, which is improved by at least 13%. It shows that our model has good anomaly detection ability for sequence data. And when compared with traditional anomaly detection algorithms, our deep learning algorithm is not weaker than traditional machine learning.

TABLE II
AUC COMPARISONS BETWEEN THE BASELINES AND LVEAD. THE BEST RESULTS ARE TYPESET IN BOLD.

Name	OUR*	ANOGAN	ALAD	MLP-VAE	IF
kdd99	0.958	0.887	0.950	0.622	0.929
Arrhythmia	0.758	0.576	0.515	0.747	0.530
ItalyPowerDemand	0.761	0.516	0.538	0.768	0.763
TwoLeadECG	0.891	0.554	0.515	0.731	0.760
GunPointAgeSpan	0.881	0.515	0.547	0.821	0.612
MoteStrain	0.840	0.746	0.504	0.750	0.762
ToeSegmentation2	0.846	0.547	0.544	0.816	0.787
Herring	0.659	0.488	0.569	0.627	0.698
Wafer	0.965	0.558	0.587	0.790	0.847
ECGFiveDays	0.970	0.970	0.694	0.910	0.678

C. The distribution of Anomaly Score

In order to verify that the anomaly score metric is effective in distinguishing anomalies, we choose ECGFiveDays as the experimental data, and the AUC value of our algorithm can be close to 1 under this experimental distribution. Fig. 4 shows the distribution of anomaly scores after 16 cross-validations of 26 anomalous samples and 20 normal samples. The x-coordinate is the time point and the y-coordinate is the anomaly score. The blue region is the abnormal score distribution of the normal sample, the red region is the abnormal score distribution of the abnormal sample, the blue poly-line is the mean of the normal sample score, and the red poly-line is the mean of the abnormal sample score. The scores for non-anomalous executions show a specific pattern of change, and the maximum value does not exceed 0.15, and the mean and variance are smaller than the abnormal execution; the range of abnormal scores is very large, up to about 0.35. Among them, the most severe change in the abnormal score is the time interval $[49, 90]$, indicating that the most common anomaly occurs during this time range, resulting in an abnormal sub-sequence. Analysis from the graph makes it easy to distinguish anomaly from normal.

D. Parameters Analysis

Because the anomaly score we design Equation 8 is constrained by parameters α and β , in order to get better results, we choose different value of parameters α and β varying from $(\alpha = 0.2, \beta = 0.8)$ to $(\alpha = 0.8, \beta = 0.2)$. The results can be seen in Fig.5 that when the parameters is $(\alpha = 0.6, \beta = 0.4)$, the result is better than others in average.

E. Visualization of latent representation and comparison of generated data

In order to verify the difference between the data generated by our model and the original data, we selected the KDD99

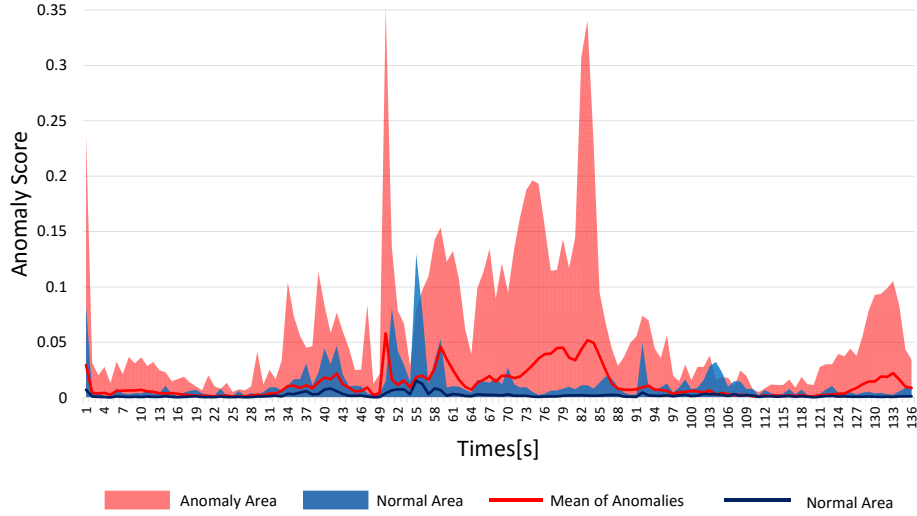


Fig. 4. Distribution of ECG abnormalities and normal sample abnormal scores. The red region is the distribution of anomaly score, and the red line is the mean anomaly value for each time point. The blue region is the distribution of normal score, and the blue line is the mean normal value for each time point.

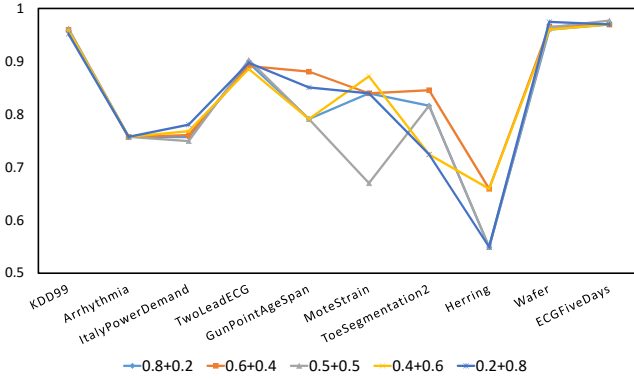


Fig. 5. Overall performance of the model based on varying parameters.

dataset and compared it with the normal sequence and the reconstructed sequence on the pixels of the image.

As shown in the Fig. 6 and Fig. 7, The blue box is the normal sample and the new sample generated by the normal sample. The red box is the abnormal sample and the new sample generated by the abnormal sample. The upper and lower rows are the original time series and the reconstructed sequence, respectively. It can be seen from the experimental results that the samples generated by our normal samples are basically similar and the normal time series is relatively stable, while the abnormal sample time series fluctuations are relatively large, and the generated new samples are significantly different from the original abnormal samples. It indicates that the abnormal sample has a great difference from the original under reconstruction, so the latent variables obtained by the encoder will naturally generate different samples, thereby judging the abnormality.

V. CONCLUSION

In our proposed method, we design an unsupervised deep learning anomaly detection method LVEAD based on the assumption that the anomalies are objects that cannot fit perfectly with the model. The model uses Encoder-Decoder-reEncoder to model the normal time series. VAE is a potential vector generated by encoding the existing data under the encoder, which is Gaussian-distributed and can retain the characteristics of the original data well. The generated data will be more reasonable and accurate. To better represent sequence, we use LSTM for encoder and decoder part. Our results show that the model is able to detect anomalous sequence by using latent vector error and reconstruction error, and performs better than other baseline models. A future line of work can add additional feature like trend for training the model so that the model can better generate normal samples.

REFERENCES

- [1] I. Onat and A. Miri, "An intrusion detection system for wireless sensor networks," in *International Conference on Telecommunications*, 2017.
- [2] A. D. Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy," *IEEE Transactions on Neural Networks & Learning Systems*, vol. 29, no. 8, pp. 3784–3797, 2018.
- [3] W. Du, F. Lei, and N. Peng, "Lad: Localization anomaly detection for wireless sensor networks," *Journal of Parallel & Distributed Computing*, vol. 66, no. 7, pp. 874–886, 2005.
- [4] T. Schlegl, P. Seebeck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," 2017.
- [5] S. Sathe and C. C. Aggarwal, "Subspace histograms for outlier detection in linear time," *Knowledge & Information Systems*, pp. 1–25, 2018.
- [6] H. Ren, M. Liu, Z. Li, and W. Pedrycz, "A piecewise aggregate pattern representation approach for anomaly detection in time series," *Knowledge-Based Systems*, 2017.
- [7] S. Salvador and P. Chan, "Learning states and rules for detecting anomalies in time series," *Applied Intelligence*, vol. 23, no. 3, pp. 241–255, 2005.

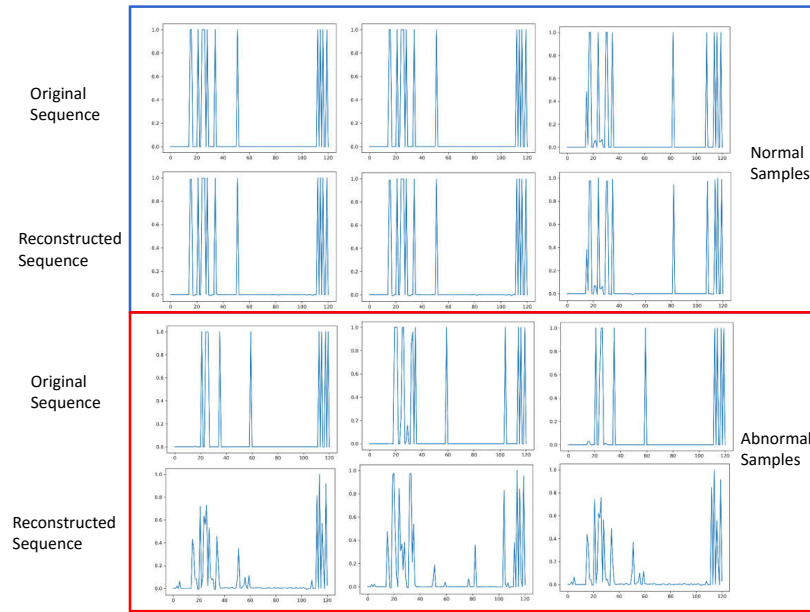


Fig. 6. Comparison of reconstructed and original samples of KDD99.

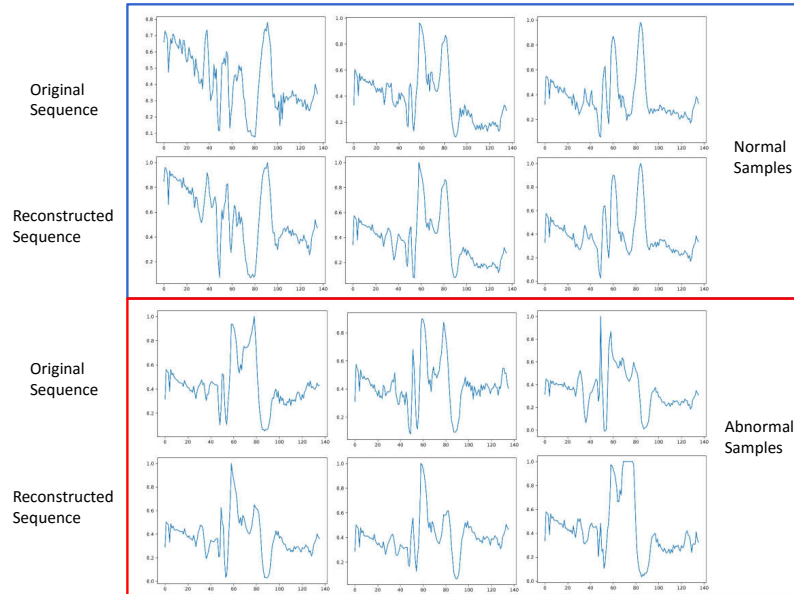


Fig. 7. Comparison of reconstructed and original samples of ECGFiveDays.

- [8] C. Kim, J. Lee, R. Kim, Y. Park, and J. Kang, "Deepnap: Deep neural anomaly pre-detection in a semiconductor fab," *Information Sciences*, vol. 457, p. S002002551830375X, 2018.
- [9] R. Duda, "Pattern classification and scene analysis," *IEEE Transactions on Automatic Control*, vol. 19, no. 4, pp. 462–463, 2003.
- [10] V. B. S. Prasath, H. A. A. Alfeilat, O. Lasassmeh, A. B. A. Hassanat, and A. S. Tarawneh, "Distance and Similarity Measures Effect on the Performance of K-Nearest Neighbor Classifier – A Review." [Online]. Available: <http://arxiv.org/abs/1708.04321>
- [11] K. Tamura and T. Ichimura, "Clustering of time series using hybrid symbolic aggregate approximation," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, pp. 1–8. [Online]. Available: <http://ieeexplore.ieee.org/document/8280846/>
- [12] C. Zhang, Y. Chen, A. Yin, Z. Qin, X. Zhang, K. Zhang, and Z. L. Jiang, "An Improvement of PAA on Trend-Based Approximation for Time Series," in *Algorithms and Architectures for Parallel Processing*, J. Vaidya and J. Li, Eds., vol. 11335. Springer International Publishing, pp. 248–262. [Online]. Available: http://link.springer.com/10.1007/978-3-030-05054-2_19
- [13] C. C. Aggarwal and P. S. Yu, "Outlier Detection for High Dimensional Data," p. 10.
- [14] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality reduction for fast similarity search in large time series databases," *Knowledge & Information Systems*, vol. 3, no. 3, pp. 263–286, 2001.

- [15] D. Park, Y. Hoshi, and C. C. Kemp, "A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-based Variational Autoencoder." [Online]. Available: <http://arxiv.org/abs/1711.00614>
- [16] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection." [Online]. Available: <http://link.springer.com/10.1007/s10586-017-1117-8>
- [17] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sanchez, "A survey on deep learning in medical image analysis," vol. 42, pp. 60–88. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1361841517301135>
- [18] S. S. Joshi, "Investigating hidden markov models capabilities in anomaly detection," in *Southeast Regional Conference*, 2005.
- [19] S. Ahmad and S. Purdy, "Real-time anomaly detection for streaming analytics," 2016.
- [20] T. Feng, Z. Du, Y. Sun, J. Wei, B. Jing, and J. Liu, "Real-time anomaly detection of short-time-scale gwac survey light curves," in *IEEE International Congress on Big Data*, 2017.
- [21] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long Short Term Memory Networks for Anomaly Detection in Time Series," p. 6.
- [22] J. Bayer and C. Osendorfer, "Learning stochastic recurrent networks," *Eprint Arxiv*, 2015.
- [23] M. Slch, J. Bayer, M. Lundersdorfer, and P. V. D. Smagt, "Variational inference for on-line anomaly detection in high-dimensional time series," 2016.
- [24] "2015 - Variational Autoencoder based Anomaly Detection using Reconstruction Probability.pdf."
- [25] P. Seebck, S. Waldstein, S. Klimscha, B. S. Gerendas, R. Donner, T. Schlegl, U. Schmidt-Erfurth, and G. Langs, "Identifying and categorizing anomalies in retinal imaging data," 2016.
- [26] Y. Chen, X. S. Zhou, and T. S. Huang, "One-class svm for learning in image retrieval," in *International Conference on Image Processing*, 2001.
- [27] H. Zenati, M. Romain, C. S. Foo, B. Lecouat, and V. R. Chandrasekhar, "Adversarially learned anomaly detection," 2018.
- [28] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, "The ucr time series classification archive," October 2018, https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.
- [29] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [30] F. T. Liu, M. T. Kai, and Z. H. Zhou, "Isolation forest," in *Eighth IEEE International Conference on Data Mining*, 2009.
- [31] C. X. Ling, J. Huang, and H. Zhang, "Auc: a statistically consistent and more discriminating measure than accuracy," in *International Joint Conference on Artificial Intelligence*, 2003.