An Effective Adversarial Training and Prediction Network for Time Series Anomaly Detection

Yuxuan Liu\textsuperscript{1,2*}, Jiangyong Duan\textsuperscript{1} and Juan Meng\textsuperscript{1}

\textsuperscript{1} Key Laboratory of Space Utilization, Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences, Beijing, China, 100094
\textsuperscript{2} University of Chinese Academy of Sciences, Beijing, China, 100049
*Corresponding author’s e-mail: liuyuxuan17@csu.ac.cn

Abstract. Due to the scarcity and uncertainty of anomalies, anomaly detection becomes a challenging problem in communities. This paper proposes an effective time series anomaly detection network based on prediction and adversarial training. First, we employ LSTM model based on differential attention mechanism to fully extract the inner characters of time series. To prevent the over-fitting due to only normal samples used to train our LSTM model, we introduce an adversarial training strategy by adding adversarial samples to enhance the generalization of the model on normal data. The effectiveness of our model is demonstrated on several public datasets.

1. Introduction
Time series anomaly detection has been widely applied in different fields [1], e.g., equipment fault diagnosis [2] in aerospace industry and malicious attacks and intrusion detection in network traffic [3]. In many situations, abnormal data is difficult and time-consuming to be obtained [4] [5]. Due to the scarcity and uncertainty of anomalies, anomaly detection becomes a challenging problem.

Generally, anomaly detection methods can be divided into two categories: 1) supervised anomaly detection; 2) unsupervised anomaly detection. Among them, the supervised anomaly detection methods [6] require a large number of labeled abnormal samples to learn abnormal characters, which limits their applications. In addition, new outlier patterns may be different from the known outliers [7], which also limits their applications. Unsupervised anomaly detection approaches learn the normal modes and detect the abnormal modes which are different from normal modes. However, these methods usually produce high false alarm rates [8].

With the successfulness of deep learning methods in computer vision and speech recognition [9], many deep learning models are applied for time series anomaly detection. At present, the widely used deep learning anomaly detection methods are depth Autoencoder and depth anomaly detection model based on regression [10]. In the case of enough normal training samples, the reconstruction error of normal samples using Autoencoder reconstruction method is small, while that of abnormal samples with different distribution is large, thus to detect anomalies. Malhotra et al. [11] use LSTM based Encoder-Decoder reconstruction method to detect time series anomalies, which find the low-dimensional embedded data and reconstructed it back to the original data space, and separate the outliers by reconstruction errors. In addition, Marchi et al. propose a bidirectional LSTM based Autoencoder model for acoustic signal detection [12], which is superior to traditional GMM and HMM methods. In [13], An et al. use the reconstruction probability model of variational Autoencoder
to provide a reconstruction probability measure for finding the exception and analyzing the root cause of the exception.

Another class of depth anomaly detection method is based on regression prediction to detect anomalies by comparing the error between the predicted value and the observed value of prediction model. In [14], Runtian et al. use the depth time series prediction model based on LSTM for anomaly detection of light curve, and achieve satisfying results. In [15], Marchi et al. also employ LSTM based nonlinear prediction for anomaly detection of acoustic signals by predicting the spectrum value of the short-term sequence from the preceding sequence. Besides, many other imposed LSTM models, such as DA-LSTM model [16], are proposed to produce better prediction accuracy.

For depth Autoencoder and anomaly detection models based on depth regression prediction model, they do not use anomaly samples and thus undesired under-fitting and over-fitting may be produced. To this end, this paper introduces a method based on adversarial training on LSTM model to improve the generalization performance of anomaly detection model. While LSTM model can learn complex and long-time dependence on time series, adversarial samples and adversarial training strategies can compensate for the lack of abnormal data, thus to produce better accuracy and generalization for normal and abnormal data.

2. Our Method
In anomaly detection, with only normal data available, we adopt the improved LSTM model and adversarial training strategy to improve the accuracy and generalization to the normal data. In order to fully extract changing rules of time series, we first introduce the time series difference feature to improve the prediction performance of the model. However, we only use normal samples during the training process, which may lead to under-fitting or over-fitting for the learning of normal model. To improve the generalization, we introduce adversarial training strategy to alleviate the deviation caused by the absence of abnormal samples. Our model is shown in figure 1.

First, we put the time series of step 1 into the DA-LSTM prediction model, we get an output prediction value. From the error between the output prediction value and the real value, we get a loss of step 1. Through the FGSM, we generate the adversarial disturbance of the input time series, and add it to the time series of step 1. Then a new input time series of step 2 is obtained. The final loss is composed by the two steps.
2.1 Difference Attention LSTM Prediction Model

Our LSTM prediction model is based on differential attention LSTM (DA-LSTM) [16]. DA-LSTM introduce attention mechanism into standard LSTM model, makes the model pay more attention to the change characteristics of time series itself. LSTM is an improved recurrent neural network. By using memory cells, the gradient vanishing problem in the long-time data training process is overcome, which makes LSTM suitable for processing time series with long-time dependence. The DA-LSTM method uses the added magnitude change feature, which can further learn the change rule and the implied pattern of complex sequence.

Standard LSTM [17] is sequentially stacked by LSTM cells as shown in figure 2. The typical LSTM cell updates the cell state $C_t$ and the output hidden state $h_t$ through three gates, which are input gate $i_t$, forget gate $f_t$ and output gate $O_t$. Let us denote the input as $X = (x_1, ..., x_n)$, output time series as $\hat{X} = (\hat{x}_1, ..., \hat{x}_n)$, hidden state of memory cells as $H = (h_1, ..., h_n)$.

![Figure 2. LSTM cell.](image)

The standard LSTM model only uses the last time step’s hidden state to make prediction. While our DA-LSTM model takes all the information at each time step into attention mechanism and employs additional difference features to focus on the time series changes. The detail structure of DA-LSTM is shown as figure 3.

![Figure 3. DA-LSTM model.](image)

Let us denote the input of our DA-LSTM model as $X_{t:t+n} = (x_t, ..., x_{t+n})$, output as $\hat{x}_{t+n}$, hidden state as $H_{t-1:t+n} = (h_{t-1}, ..., h_{t+n})$, the cell unit state as $C_{t-1:t+n} = (C_{t-1}, ..., C_{t+n})$. In DA-LSTM, we extract...
the difference feature \( \text{diff}_t = [(x_{t-1} - x_{t-2})^2, (x_t - x_{t-n})^2] \) to measure the input time series changes. By concatenating the difference feature and the hidden state into a new hidden feature \( \tilde{h}_t = [\text{diff}_t, h_t] \), the attentional weight \( \alpha_t \) indicating the importance of each time step is calculated by:

\[
\alpha_t = \frac{\exp \left( f_{\text{attention}} (W_s[\text{diff}_t, h_t]) \right)}{\sum_{k=1}^{t} \exp \left( f_{\text{attention}} (W_s[\text{diff}_k, h_k]) \right)}
\]  

Given \( \alpha_t \) and the hidden state \( \tilde{h}_t \), we replace the output \( \hat{x}_{t+n} = f_{\tilde{h}}(h_{t+n}) \) computed by the full connection with the hidden state of the last cell in standard LSTM by the new formulation \( \hat{x}_{t+n} = f_{\tilde{h}}(\theta_{t+n}) \) where \( \theta_{t+n} = \sum_{i=t}^{t+n} \alpha_i \tilde{h}_i \) is the weighted sum of the concatenated hidden layer \( \tilde{h}_i \).

### 2.2 Adversarial Training

For anomaly detection by DA-LSTM prediction, with only training on normal data, it only learns the patterns of normal data and doesn’t learn the patterns of different abnormal data. Thus the model may be under-fitting or over-fitting on normal data. Some generalization strategies are needed to reduce the prediction error for both training and testing normal data.

Specifically, we adopt the method of adding adversarial samples [18] [19] in the training process to adversarially train the DA-LSTM prediction model. When the normal samples and their adversarial samples are included when training, the model learns more knowledge of the normal samples.

We use Fast Gradient Sign Method (FGSM) [20] to obtain the adversarial disturbance and generate the adversarial sample. In DA-LSTM, let \( x_{t+n} \) as input, \( \theta \) as model parameter, \( \text{sign} \) as sign function, \( \varepsilon \) as disturbance magnitude, and \( J(\theta, x_{t+n}) \) as forecast cost. If a disturbance \( r \) is added to the original input, the corresponding adversarial sample \( x_{t+n}^{\text{adv}} \) is represented as follows.

\[
r = \varepsilon \text{sign}(\nabla_{x_{t+n}} J(\theta, x_{t+n}))
\]

\[
x_{t+n}^{\text{adv}} = x_{t+n} + r
\]

After we get the adversarial sample, we use it for training. This is equivalent to optimizing the following objective function:

\[
J^{\text{adv}}(\theta, x_{t+n}) = (1 - \alpha)J(\theta, x_{t+n}) + \alpha J(\theta, x_{t+n} + \varepsilon \text{sign}(\nabla_{x_{t+n}} J(\theta, x_{t+n})))
\]

We set \( \alpha = 0.5 \), which means all the examples are equally treated. We call this architecture adversarial DA-LSTM (Adv-DA-LSTM).

We put the input \( x_{t+n} \) into the DA-LSTM prediction model, and calculate the prediction loss \( J(\theta, x_{t+n}) \) from the predicted value and the true value. The loss is as shown in equation (5).

\[
J(\theta, x_{t+n}) = \frac{1}{N} \left\| x_{t+n+1} - \hat{x}_{t+n+1} \right\|^2
\]

where \( N \) is the dimension of input, \( x_{t+n+1} \) is true value, and \( \hat{x}_{t+n+1} \) is prediction value. Then a disturbance \( r \) is sent back to the input to obtain the adversarial sample \( x_{t+n}^{\text{adv}} \), which is sent into the DA-LSTM again. We get another loss \( J(\theta, x_{t+n} + \varepsilon \text{sign}(\nabla_{x_{t+n}} J(\theta, x_{t+n}))) \), which is shown as equation (6). Finally, both of them are optimized as a whole.

\[
J(\theta, x_{t+n} + \varepsilon \text{sign}(\nabla_{x_{t+n}} J(\theta, x_{t+n}))) = \frac{1}{N} \left\| x_{t+n+1}^{\text{adv}} - \hat{x}_{t+n+1}^{\text{adv}} \right\|^2
\]

We employ Adam [21] method to optimize the loss. In each iteration, some adversarial examples are generated and added to the training data to improve the robustness of the model. In addition to the
traditional regularization methods (dropout, pre-training, model averaging, etc.), adversarial training provides another regularization method.

2.3 Anomaly Detection
In anomaly detection, for each input sequence \( x_{tr,n} \), we obtain a prediction value by our Adv-DA-LSTM model. By comparing the error between the predicted value and the true value with the threshold, we can judge whether there is any abnormality. If the error is larger than threshold, the sample is abnormal. Otherwise, the sample is normal.

3. Experiments
We compare our Adv-DA-LSTM model with the representative method of Autoencoder model, i.e., LSTM Encoder-Decoder [11], and the regression prediction model, i.e., DA-LSTM prediction model [16]. In all methods, the input sequence length and the hidden layer size are set to 40 and 200, respectively. We used the AUC (area under the ROC curve) [22] as the evaluation criterion with larger AUC indicating better results. Five public evaluation datasets [23] and one telemetry dataset are tested and listed in table 1.

Table 1. Table of Data properties, where \( n \) is the number of instances, and \( d \) is the number of dimensions, and the percentage in bracket indicates the percentage of anomalies.

| Dataset | \( n \) | \( d \) | Anomaly |
|---------|--------|-------|---------|
| Shuttle | 15000  | 1     | 741 (5\%) |
| Power   | 35040  | 1     | 2254 (6.4\%) |
| Respiration | 42177  | 1     | 1802 (4.3\%) |
| Gesture | 11251  | 2     | 739 (6.6\%) |
| ECG     | 91202  | 2     | 1871 (2.1\%) |
| Gamma   | 96662  | 66    | 3103 (0.03\%) |

The anomaly detection results are shown in table 2. Our Adv-DA-LSTM and DA-LSTM are generally better than LSTM Encoder-Decoder on the tested dataset. This may be due to encoder-decoder reconstruction strategy in LSTM may produce over-complete identity mapping from encoder to decoder.

Our Adv-DA-LSTM produces best results on all the six datasets. Our method introduces adversarial data, which are added during the training for normal data. This makes our model learn more patterns and rules of normal samples, and thus increases the robustness of our model and suppresses the performance of model generalization to abnormal data.

Table 2. Comparison results.

| Dataset | Adv-DA-LSTM | DA-LSTM | LSTM Encoder-Decoder |
|---------|-------------|---------|----------------------|
| Shuttle | **0.8015**  | 0.7983  | 0.7932               |
| Power   | **0.7678**  | 0.7645  | 0.7623               |
| Respiration | **0.6309** | 0.6275  | 0.6236               |
| Gesture | **0.7751**  | 0.7723  | 0.7691               |
| ECG     | **0.7620**  | 0.7572  | 0.7503               |
| Gamma   | **0.8584**  | 0.8513  | 0.8478               |

4. Conclusion
In this paper, we proposed a novel model Adv-DA-LSTM for time series anomaly detection. The model used DA-LSTM prediction model to fully learn the change information in the time series for effective prediction. The DA-LSTM prediction model then was trained by an adversarial way to make up for the lack of abnormal data. Satisfying results had demonstrated the effectiveness of our model.
Acknowledgments
This work was supported by the National Natural Science Foundation of China (No. 61901454, No. 61971404 and No. 61501434)

References
[1] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3): 15.1-15.58.
[2] Gupta, M., Gao, J., Aggarwal, C.C., & Han, J. (2014). Outlier detection for temporal data: A survey. Knowledge and Data Engineering, IEEE Transactions on, 26(9): 2250-2267.
[3] Kwon, D., Kim, H., Kim, J., Suh, S.C., Kim, I., & Kim, K.J. (2017). A survey of deep learning-based network anomaly detection. Cluster Computing, 22: 949-961.
[4] Liu, F.T., Ting, K.M., & Zhou, Z.H. (2009). Isolation Forest. In: The Eighth IEEE International Conference on Data Mining. Pisa. pp. 413-422.
[5] Shipmon, D.T., Gurevitch, J.M., Piselli, P.M., & Edwards, S.T. (2017). Time Series Anomaly Detection; Detection of anomalous drops with limited features and sparse examples in noisy highly periodic data. arXiv preprint arXiv:1708.03665.
[6] Jumutc, V., & Suykens, J. A. (2014). Multi-class supervised novelty detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 36(12): 2510-2523.
[7] Song, H.C., Jiang, Z.Q., Men, A.D. (2017). A Hybrid Semi-Supervised Anomaly Detection Model for High-Dimensional Data. Computational Intelligence & Neuroscience, 1-9.
[8] Xue, Z., Shang, Y., & Feng, A. (2010). Semi-supervised outlier detection based on fuzzy rough C-means clustering. Mathematics and Computers in Simulation, 80(9): 1911-1921.
[9] Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553): 436-444.
[10] Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1901.03407.
[11] Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., & Shroff, G. (2016). LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection. arXiv preprint arXiv:1607.00148.
[12] Marchi, E., Vesperini, F., Eyben, F., Squartini, S., & Schuller, B. (2015). A Novel Approach for Automatic Acoustic Novelty Detection Using a Denoising Autoencoder with Bidirectional LSTM Neural Networks. In: Proceedings of ICASSP IEEE International Conference on Acoustics Speech & Signal Processing. Brisbane. pp. 1996–2000.
[13] An, J., & Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability. Special Lecture on IE, 2: 1-18.
[14] Runtian, Z., & Qian, Z. (2018). Time Series Prediction and Anomaly Detection of Light Curve Using LSTM Neural Network. Journal of Physics: Conference Series, 1061: 012012.
[15] Marchi, E., Vesperini, F., Weninger, F., Eyben, F., & Schuller, B. (2015). Non-Linear Prediction with LSTM Recurrent Neural Networks for Acoustic Novelty Detection. In: 2015 International Joint Conference on Neural Networks(IJCNN). Killarney. pp. 1–7.
[16] Liu, Y.X., Duan, J.Y., & Meng, J. (2020). Difference Attention Based Error Correction LSTM Model for Time Series Prediction. arXiv preprint arXiv: 2003.13616.
[17] Gers, F. A., Schmidhuber, Jürgen, & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. Neural Computation, 12(10): 2451-2471.
[18] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., & Fergus, R. (2014). Intriguing properties of neural networks. arXiv preprint arXiv: 1312.6199.
[19] Nguyen, A., Yosinski, J., & Clune, J. (2015). Deep neural networks are easily fooled: high confidence predictions for unrecognizable images. In: The IEEE Conference on Computer Vision and Pattern Recognition. Boston. pp. 427-436.
[20] Goodfellow, I.J., Shlens, J., & Szegedy, C. (2014). Explaining and Harnessing Adversarial Examples. arXiv preprint arXiv: 1412.6572.

[21] Kingma, D., & Ba, J. (2014) Adam: A Method for Stochastic Optimization. arXiv preprint arXiv: 1412.6980.

[22] Hanley, J.A., & Mcneil, B.J. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. Radiology, 143(1): 29-36.

[23] Keogh, E., Lin, J., & Fu, A. (2005). HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence. In: The Fifth IEEE International Conference on Data Mining. Houston. pp. 226-233.