Anomaly Detection in Univariate Time Series: HOT SAX vesus LSTM-Based Method

Duong Tuan Anh[1,2(✉)] and Tran Long Hoai[2]

[1] Department of Information Technology, HCMC University of Foreign Languages and Information Technology, Ho Chi Minh City, Vietnam
hdt@huflit.edu.vn

[2] Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam
tlhoai.sdh19@hcmut.edu.vn

Abstract. Anomaly detection in time series has been an important and challenging research topic. There have been several methods proposed for detecting anomaly subsequences in a time series. The majority of these methods is classified into the window-based category, which applies a sliding window with a fixed length to extract subsequences before finding out anomaly subsequences. A well-known algorithm in this category is HOT SAX algorithm. Recently, deep neural network models, especially Long Short Term Memory (LSTM) network, are also applied for time series anomaly discovery. LSTM-based methods for time series anomaly detection belong to prediction-based category. So far, there has been no research work to compare the performance of LSTM-based method to that of any traditional window-based method in time series anomaly detection. The research question investigated in this paper is that whether the newly developed LSTM-based method for time series anomaly detection is superior to the traditional algorithms, such as HOT SAX or not. In this study, we give an empirical comparison between LSTM-based method and HOT SAX in time series anomaly detection. Extensive experiments on seven benchmark time series indicate that LSTM-based method is not superior to HOT SAX since it brings out the same detection accuracy as HOT SAX while it incurs much higher computational overhead.

Keywords: Time series · Anomaly detection · Prediction-based approach · Window-based approach · Long Short Term Memory · HOT SAX

1 Introduction

Time series anomaly detection is important in several application areas such as fault detection, disease diagnosis, event detection and data cleaning. For univariate time series there are two commonly used categories of anomaly detection methods: window-based and prediction-based. In a method of window-based category, a sliding window with fixed length w slides through a time series. For each subsequence extracted under the sliding window, the Euclidean distance from it to the closest subsequence in the time series

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2024
Published by Springer Nature Switzerland AG 2024. All Rights Reserved
P. Cong Vinh and N. Thanh Tung (Eds.): ICCASA 2023, LNICST 579, pp. 70–84, 2024.
https://doi.org/10.1007/978-3-031-58878-5_5

is computed and used as the anomaly score of the subsequences. Empirical evaluations have shown that this simple method is effective for many different kinds of time series. In a method of prediction-based category, a prediction model is trained to learn the normal behavior of a time series segment (training part of the time series), and prediction errors are used to discover anomaly patterns on the test part of the time series with the rule that any points very different to their predicted values are considered as anomaly points.

Some typical algorithms of the first category are Brute-Force, a naïve method proposed by Keogh et al. (2005) [1]; HOT SAX, devised by Keogh et al. (2005) [1]; WAT algorithm by Bu et al. (2007) [2] and BitClusterDiscord algorithm by Li et al. (2013) [3].

Some typical algorithms of the second category can be listed as follows. Oliveira and Meira, in 2006, proposed a method which can detect anomaly patterns through neural network forecasting with robust confidence intervals [4]. Pena et al. in 2013, proposed a method for time series anomaly detection which is based on ARIMA model and Holt-Winters model [5]. Yu et al. in 2014, proposed a method for time series anomaly detection which is based on AR (Auto Regression) model and sliding window prediction [6].

Recently, deep neural network-based anomaly detection algorithms have become increasingly popular and have been applied for a variety of practical areas. Most of deep neural network-based anomaly detection algorithms belong to prediction-based category. Since LSTM network outperforms many other models in time series forecasting ([7–10]), LSTM-based approach has been used in deep learning-based anomaly detection algorithms which belongs to prediction-based category [11]. Some typical LSTM-based methods in time series anomaly detection can be listed as follows. Malhotra et al., in 2015, proposed a time series anomaly discovery method, called LSTM-AD, which utilizes stacked LSTM for prediction and uses prediction errors to detect anomalies [12]. Chauhan and Vig in 2015 devised an LSTM-based method to detect anomalies in electrocardiography (ECG) time series [13]. Buda et al. in 2018 proposed a time series anomaly discovery method, called DeepAD which combines stacked LSTM with traditional forecasting methods such as ARIMA and Holt-Winters in a prediction-based anomaly detection algorithm [14]. Zhang et al. in 2020 proposed an LSTM-based algorithm which can forecast electrical load along with detecting anomalies and adjusting them in order to improve forecasting quality at real time [15].

An interesting and important research question is whether the newly developed LSTM-based method for time series anomaly detection is superior to the traditional algorithms, such as HOT SAX algorithm or not. To the best of our knowledge, there is no specific empirical research work to assess the performance of LSTM-based method in time series anomaly discovery in comparison with traditional methods such as HOT SAX.

This work aims to compare LSTM-based method and HOT SAX in time series anomaly detection on seven benchmark datasets in two perspectives detection accuracy and time efficiency. Extensive experiments on seven benchmark time series indicate that LSTM-based method is not superior to HOT SAX since it brings out the same detection accuracy as HOT SAX while it incurs much higher computational overhead.

The rest of the paper is structured as follows. Section 2 presents some basic definitions about time series anomalies, taxonomy of time series anomaly detection and

LSTM neural networks. In Sect. 3, we describe the LSTM-based approach and HOT SAX algorithm for time series anomaly discovery. Section 4 reports the experiments to compare the performance of the two comparative methods on seven time series datasets in two aspects: accuracy and time efficiency. Finally, Sect. 5 gives some conclusions and notes for future studies.

2 Background

2.1 Some Definitions

A time series is a sequence of real numbers measured at equal time intervals. A time series can be a sequence of observations collected from one source, for example, one sensor. In this case, the series is univariate. If we collect information from more than one source, we have a multivariate time series. In this paper, we consider only univariate time series.

Definition 2.1. *Non-self match*: Given a time series T including a subsequence C of length n starting from position p and a matching subsequence M starting from the position q. If $|p - q| \geq n$, M is called as a non-self match to C.

Definition 2.2. Time series 1-discord: Given a time series T, containing the subsequence C of length n beginning at position p. if C has the largest distance to its nearest non-self match, C is called as the top discord (1-discord) of T.

The 1-discord in a time series is also called the top anomaly subsequence in that time series.

2.2 Taxonomy of Time Series Anomaly Detection Methods

The methods for time series anomaly detection are grouped into four categories: window-based methods, segmentation-based methods, prediction-based methods and classification-based methods.

Window-based method extracts fixed length windows (subsequences) from the time series and computes the distances between the current subsequence with all other subsequences in the time series (using some distance measure). The subsequence which has the largest distance to its closest matching subsequence is considered as the top anomaly pattern. HOT SAX is a typical algorithm for finding the top anomaly in time series which belongs to window-based category. The top anomaly detection algorithm can be extended to become the top-k anomaly detection algorithm.

Prediction-based method uses a prediction model to fit the time series and obtains the predicted value using on the past data. Points (subsequences) that deviate remarkably from their predicted values are determined as anomaly points (subsequences). The predictor used in this anomaly detection approach can be a statistical models such as Auto Regression (AR), ARMA, ARIMA or a machine learning model such as artificial neural network, Support Vector Regression (SVR).

Classification-based method uses some technique to extract fixed length subsequences from the training part of a time series and labels each of them as normal pattern or anomaly pattern. Using the set of these class labeled patterns, a classifier is trained and it can be used to classify a new subsequence as normal or anomaly pattern.

Segmentation-based method uses some segmentation technique to split a time series into segments (subsequences). Then a clustering algorithm is used to group the subsequences into clusters. Using the results of clustering, anomaly scores of all the subsequences will be determined and the subsequences of which the anomaly scores are higher than a given threshold will be considered as anomaly patterns.

Prediction-based methods and classification-based methods can be viewed as supervised or semi-supervised anomaly discovery methods while window-based methods and segmentation-based methods can be viewed as unsupervised anomaly discovery methods.

The experimental results in the survey on time series anomaly detection by Chandola et al., in 2009 [16], revealed that generally, window-based methods tend to outperform prediction-based methods.

2.3 Long Short Term Memory

Long Short-Term Memory (LSTM) [17] is an improved variant of Recurrent Neural Network (RNN) designed specifically for sequential data. Each LSTM unit is a generalization of RNN unit, such that part of information about previous time series data points is stored into the network.

Each LSTM unit has three gates:

- Forget gate, which is responsible for deciding which part of information from the previous state should be saved or thrown away.
- Output gate, which is responsible for selecting how much information should be output.
- Input gate, which is responsible for obtaining new information.

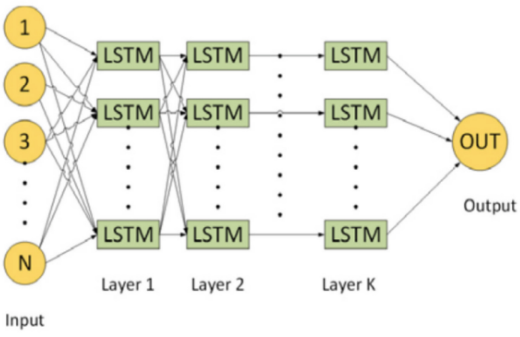

Fig. 1. Stacked Long Short Term Memory network

The deep LSTM neural network includes more than one hidden layer. It consists of multiple layers of LSTM units in which the outputs of the previous layer become the

inputs of the next layer. Figure 1 describes the structure of a stacked Long Short Term Memory network which can be used in time series prediction.

3 The Selected Comparative Methods for Time Series Anomaly Detection

In this section, we describe the LSTM-based anomaly detection method and HOT SAX algorithm.

3.1 A Prediction-Based Method: LSTM-Based Anomaly Detection

In this study, we use LSTM-based method for time series anomaly detection which belongs to the category of prediction-based methods. Previous study shows that LSTM-based method is a strong baseline for time series prediction, which outperforms several other prediction methods. Due to this reason, stacked LSTM is a good choice to be predictor in a prediction-based method for anomaly detection. As for time series anomaly detection, inspired by the LSTM_AD method (Malhotra et al., 2015 [12]), we model the normal behavior of a time series through a stacked LSTM network and detect deviations from normal behavior as anomaly patterns. For the prediction task in anomaly detection, we apply multistep-ahead prediction strategy. The workflow of LSTM-based method for time series anomaly discovery is described as the two following steps.

Prediction task
We first train a prediction model using stacked LSTM network and then compute the prediction error distribution using which we detect anomalies. The prediction model accepts the input consisting of p data points in the most recent past and predicts the output consisting of q future values. We stack LSTM layers such that each unit in a lower LSTM hidden layer is fully connected to each unit in the LSTM hidden layer above it through feedforward connections (see Fig. 1). There is one unit at the output layer (also called dense layer). Here we apply multistep-ahead prediction with *recursive strategy* [18]. We use linear transfer function at the input units and MSE (mean squared error) as error function. The model is trained on normal part of the time series in order that it can learn the normal behavior of the time series.

Through multistep-ahead prediction with *lookahead* parameter q at the time point t, the model will predict q values in the future (i.e. at $t + 1, t + 2, \ldots, t + q$).

Detecting anomaly patterns
The anomaly detection can be done by using prediction errors. Prediction error means the deviation between the predicted value and the actual value at time point t. The prediction errors can be modeled by using Gaussian distribution function. The parameters of Gaussian function: the mean μ and the standard deviation ρ can be estimated from the prediction errors using Maximum Likelihood Estimation (MLE). Probability density (PD) function is defined by:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{1}$$

As for prediction errors, to avoid vanishing problem when the values becoming so small, the logarithms of probability density values (logPD) of prediction errors should be computed and used as a measure to detect anomalies. A test part of dataset which contains normal data and abnormal data is used to determine the *threshold* logPD *value*. This threshold value will be used to detect anomaly values. In the test part, any data point of which the logPD value is smaller that the threshold logPD will be considered as anomaly data point.

The algorithm

From the two above-mentioned steps, we come to the LSTM-based algorithm for anomaly detection in time series as follows.

Each given time series is divided into four parts. N: the training set which contains only normal data points will be used to train the LSTM model. V_N: the validation set which contains only normal data points will be used to evaluate the model in training stage. V_A: the validation set which contains both normal data points and abnormal data points will be used to compute the threshold for logPD. T: the test set which contains both normal data points and abnormal data points will be used to detect anomaly patterns in the time series. The steps of the algorithm are described as follows:

Step 1: The training set N is used to train the LSTM model. All the hyper-parameters of this model is determined through experiments.
Step 2: The validation set V_A is used in the model training stage to early stop the training process and to alleviate overfitting.
Step 3: The validation set V_N is used to collect the differences between the predicted values and the corresponding actual values and use MLE to determine the mean μ and the standard deviation ρ for the Gaussian distribution function.
Step 4: The logPD values for all prediction errors in the validation set V_A are computed and based on these values, the threshold value for logPD in detecting anomalies is determined.
Step 5: Based on the threshold logPD value, the process of finding anomalies will be performed on the test set T.

3.2 A Window-Based Detection Method: HOT SAX

HOT SAX, proposed by Keogh et al., 2005 [1], is a time series anomaly detection algorithm which belongs to window-based category. HOT SAX applies PAA (Piecewise Aggregate Approximation) for dimensionality reduction [19], uses Symbolic Aggregate Approximation (SAX) (Lin et al., 2003 [20]) as a discretization transform and utilizes two ordering heuristics for the inner and outer loops to improve the top anomaly search process.

One interesting property of HOT SAX is that there have been several improved variants of HOT SAX, which brings out remarkably higher time efficiency. Some improved variants of HOT SAX can be reviewed as follows. Buu and Anh in 2011 devised HOTiSAX which brings out the same accuracy as HOT SAX and executes about 4 times faster than HOT SAX [21]. Thuy et al. in 2016, proposed Hash_DD, an improved version of HOT SAX which yields the same accuracy as HOT SAX and executes about 8.4 times faster than HOT SAX [22].

Besides, Anh and Hien, in 2021 [23] proposed a time series anomaly detection algorithm, called DPDD, which applies dynamic programming into Brute-Force algorithm [1] to bring out the same accuracy as HOT SAX and the computational efficiency about 25.2 times faster than HOT SAX. Thuy et al. in 2018 [24] devised a segmentation-based approach for time series anomaly detection, called EP-ILeader, which yields the same accuracy as HOT SAX and executes about 244 times faster than HOT SAX.

4 Experimental Evaluation

In this section, we describe the experiments for comparing the performance of two methods: HOT SAX and LSTM-based anomaly detection method (abbreviated as LSTM_DD). The two methods were implemented with Python. To implement LSTM_DD we also utilize the open-source framework Keras 2.3.1 with the core TensorFlow [25]. The experiments were conducted on Macbook Pro 2020 - MWP42; 2.0 GHz Quad-Core Intel Core i5 gen-10th; RAM: 16 GB.

4.1 Data Description and Parameter Setting

We conducted the experiments over seven time series datasets. Most of the datasets are from the UCR Time Series Data Mining Archive for anomaly detection [26], except the Numenta dataset which is from [27]. The datasets are from different application domains (finance, medicine, industry, image processing). These seven benchmark time series are commonly used by the research community on time series anomaly detection.

Table 1 summarizes the domain and the length of each tested datasets. Figure 2, 3, 4, 5, 6, 7, 8 illustrate the plots of seven tested time series. Notice that Ann_gun dataset is from image processing and therefore is a two-dimensional time series (see Fig. 8).

The anomaly patterns in these benchmark datasets have been *annotated* by experts. For example, TEK16 is a sensor time series representing normal Space Shuttle Marotta Valve Time Series annotated by a NASA engineer [1]. In the plot of TEK16 shown in Fig. 5, NASA engineer annotates the 1-discord (bold in red) of this time series occuring at the fifth cycle of the time series. Another example is the Power-demand dataset which measured the power consumption for a Dutch research facility for the entire year of 1997. In the plot shown in Fig. 4, the three top discords (which are bold in green, red and purple) in Power-demand dataset are annotated by experts. The annotations given by experts in each time series are very useful in checking the accuracy of discord detection by a given method.

The parameters for HOT SAX are the discord length n, the length of PAA frame w and the alphabet size a. To estimate the discord length n and the length of PAA frame w, we applied the segmentation-based techniques which were proposed in the work [28] by Thuy et al. to facilitate parameter determination in HOT SAX. All the parameters for HOT SAX are reported in Table 2. Interested readers on how to determine parameters in HOT SAX can refer to the work [28].

As for the architecture of stacked LSTM in LSTM_DD, for most of time series datasets, it consists of two hidden layers and one dense layer. Paricularly for Power-demand dataset, it is made of one hidden layer and one dense layer. We determine

Table 1. Details of the tested datasets

Name	Description	Length
ECG	Medicine	18000
Numenta	Industry	22695
Power-demand	Industry	35040
TEK 16	Industry	5000
Stock	Finance	5000
Memory	Industry	6875
Ann-Gun	Image processing	11250

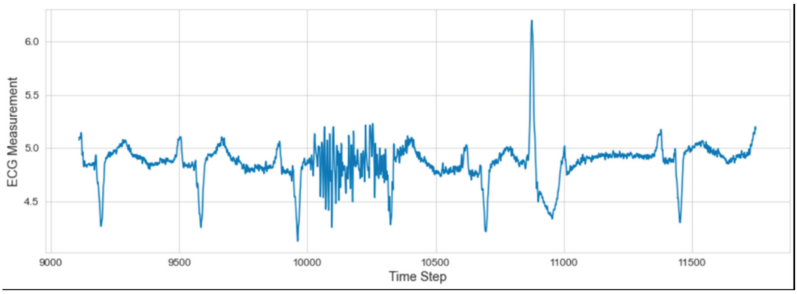

Fig. 2. The ECG time series

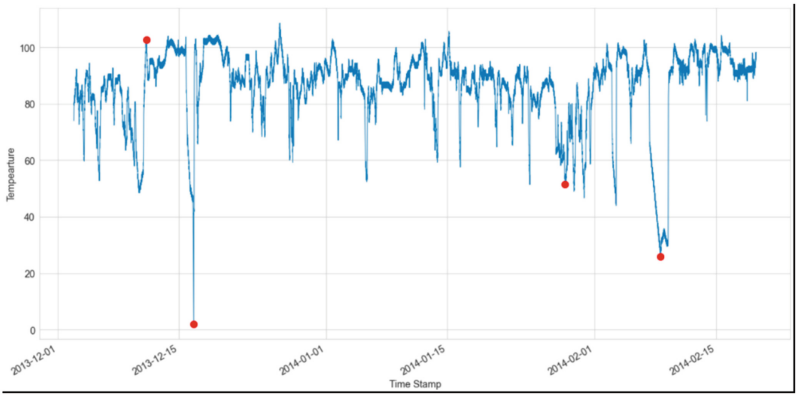

Fig. 3. The Numenta time series

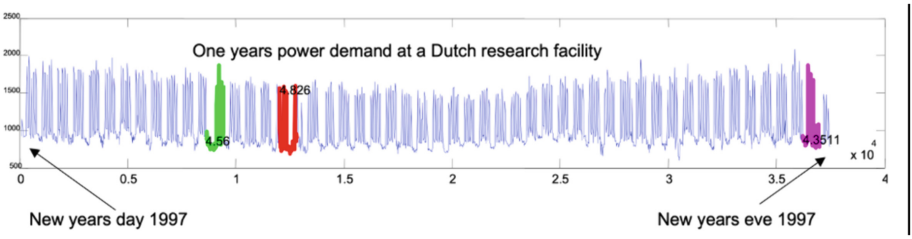

Fig. 4. The Power-demand time series

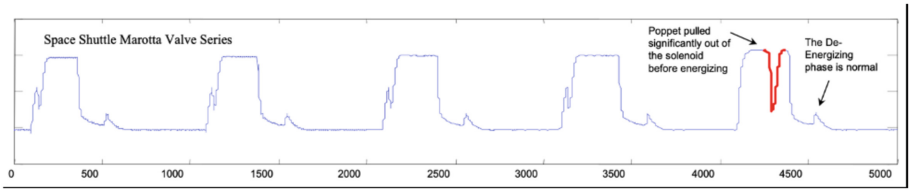

Fig. 5. The TEK16 time series

Fig. 6. The Stock time series

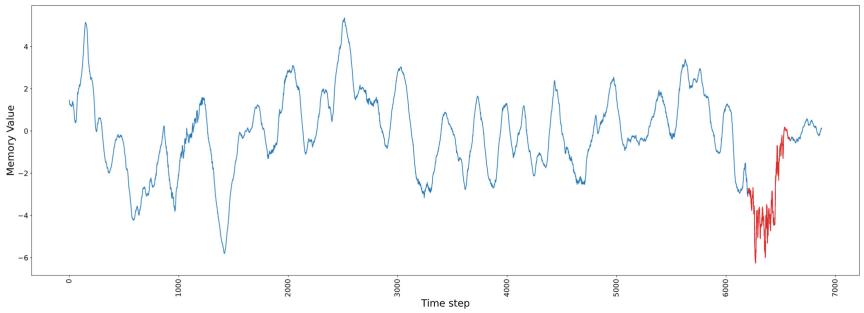

Fig. 7. The Memory time series

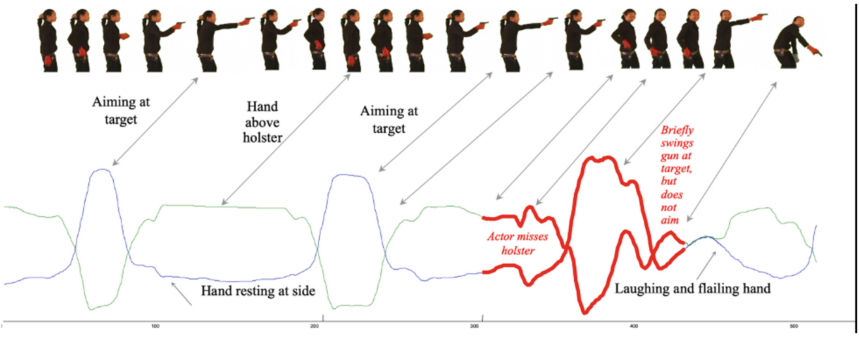

Fig.8. The plots of 2-D Ann_gun dataset

Table 2. Parameters in HOT SAX for seven datasets

Dataset	Discord length	PAA frame size	Alphabet size
ECG	n: 128	w: 8	a: 3
Numenta	n: 128	w: 8	a: 3
Power-demand	n: 1000	w: 8	a: 3
TEK 16	n: 128	w: 8	a: 3
Stock	n: 128	w: 8	a: 3
Memory	n: 128	w: 8	a: 3
Ann-Gun	n: 128	w: 8	a: 3

the hyperparametes of stacked LSTM in LSTM_DD through experiments. The stacked LSTM networks are trained with Adam optimizer [29] in which learning rate (η) and decay factor must be determined. All the parameters for LSTM_DD are given in Table 3. As for the logPD threshold in LSTM_DD for each time series dataset, we determined the suitable values as follows:

ECG: −5; Numenta: −25; Power-demand: −28;
TEK 16: –20, Stock: −5; Memory: −13;
Ann-Gun_X: −8; Ann-Gun_Y: −9.

4.2 Detection Accuracy

The key observations for LSTM-DD method from our experimental results are as follows. In Fig. 10, the logPD values in the anomaly regions are significantly lower than those in normal regions for the ECG dataset. Figure 10 (top) illustrates three plots for ECG time series in LSTM_DD (true value: green, predicted value: dashed green, error: red). Figure 10 (bottom) shows the logPD plot for ECG time series (logPD: green, logPD threshold: dashed green). We can see that this time series dataset has only one anomaly subsequence detected at the data points which have logPD values below the logPD

threshold. This anomaly detection technique used in ECG dataset are repeated in all other datasets.

Table 3. Parameters in LSTM_DD for seven datasets

Dataset	LSTM structure	Adam optimizer	Lookback, lookahead	Batch-size
ECG	1st layer: {60} Dropout: 0.1 2nd layer: {30} Dropout: 0.1 Dense layer: {1}	η (learning rate): 0.1 Decay: 0.99	8, 5	256
Numenta	1st layer: {80} Dropout: 0.1 2nd layer: {20} Dropout: 0.1 Dense layer: {1}	η: 0.05 Decay: 0.99	24, 12	1024
Power-demand	1st layer: {300} Dropout: 0.2 Dense layer: {1} Linear Activation	η: 0.01 Decay: 0.99	1, 1	672
TEK 16	1st layer: {80} Dropout: 0.2 2nd layer: {30} Dropout: 0.2 Dense layer: {1}	η: 0.01 Decay: 0.99	1, 1	1000
Stock	1st layer: {60} Dropout: 0.1 2nd layer: {30} Dropout: 0.1 Dense layer: {1}	η: 0.1 Decay: 0.99	5, 3	256
Memory	1st layer: {60} Dropout: 0.1 2nd layer: {30} Dropout: 0.1 Dense layer: {1}	η: 0.1 Decay: 0.99	5, 3	256
Ann-Gun	1st layer: {80} Dropout: 0.2 2nd layer: {30} Dropout: 0.2 Dense layer: {1}	η: 0.01 Decay: 0.99	1, 1	150

As for checking the effectiveness of the two comparative methods, we compare the resulting results given by the two methods with the expected results from the annotations given by experts based on the domain meaning of the dataset.

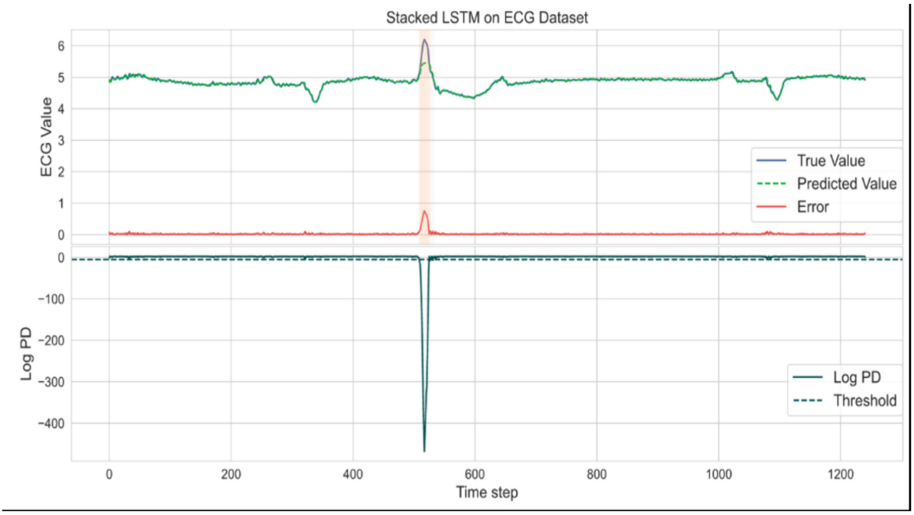

Fig. 10. Illustration of the results found by LSTM-DD on ECG dataset

From our experiments, mainly through inspection by eye, we discovered that the resultant top anomaly subsequence detected by LSTM_DD for each test dataset almost matches with the one discovered by HOT SAX and both of them match with the one annotated by experts for the dataset. Besides, for some datasets, the start location of the top anomaly subsequence detected by LSTM_DD might be slightly different from that of the one found by HOT SAX. However, this kind of differences affects slightly on the detection accuracy.

In sum, it is clear that both LSTM_DD and HOT SAX bring out the same detection accuracy in time series anomaly detection.

4.3 Efficiency

To check the time efficiency of the two comparative methods, we collected the running times (in seconds) of the two methods on seven datasets. Table 4 reports the running times of the two methods. It can be seen that the execution times of LSTM_DD are always higher than that of the HOT SAX in all seven datasets. Noted that here the running time of LSTM_DD on a given time series consists of the time for building the prediction model and the time of using the model to detect anomalies. In average, HOT SAX is about 2.58 times faster than LSTM_DD.

Besides, in comparing LSTM_DD with some improved variants of HOT SAX in time efficiency, we get some following results. As mentioned in Subsect. 3.2, since HOTiSAX is about 4 times faster than HOTSAX, HOTiSAX can run about 10.32 times faster than LSTM_DD. Since Hash_DD is about 8.2 times faster than HOTSAX, Hash_DD can run about 21.156 times faster than LSTM_DD. Since DPDD is about 25.2 times faster than HOTSAX, DPDD can execute about 65.02 times faster than LSTM_DD. As for the segmentation-based method EP-ILeader, since this algorithm can execute about 244 times faster than HOT SAX, it can run about 629.52 times faster than

LSTM_DD. This special finding suggests that efficiency is the key aspect of LSTM_DD method. LSTM_DD for time series anomaly detection should be improved further in computational efficiency to be comparable to window-based approaches.

Table 4. Time comparison of the two methods (in seconds) on seven datasets

Dataset	LSTM_DD	HOT SAX
ECG	399	30
Numenta	246	39
Power-demand	320	318
TEK 16	9	7
Stock	34	9
Memory	27	11
Gun-X	67	17
Gun-Y	51	15

From the experimental results in two performance metrics: detection accuracy and time efficiency, we can find out that LSTM_DD is not superior to HOT SAX since the accuracy of LSTM_DD is almost the same as that of HOT SAX while LSTM_DD incurs a significantly higher computational overhead. The finding from this study on comparing HOT SAX to LSTM-based method in time series anomaly discovery is somewhat consistent with the finding in the important review by Chandola et al. [12] which remarked that generally, window-based methods tend to outperform prediction-based methods.

Furthermore, determining the hyperparameters in LSTM_DD is much more complicated and challenging than estimating the parameters in HOT SAX algorithm.

5 Conclusion

In this study, we give an exploration for the research question whether LSTM-based models are better than HOT SAX in time series anomaly discovery or not. We compared HOT SAX to LSTM-based method on seven benchmark time series datasets in two perspectives: detection accuracy and time efficiency. The experimental results indicate that LSTM-based method does not outperform HOT SAX since the accuracy of LSTM-based method is almost the same as that of HOT SAX while LSTM-based method incurs a significantly higher computational overhead. Therefore, the success of LSTM-based methods in image processing, natural language processing and time series forecasting is not so easy to be replicated in time series anomaly detection. Old-fashion window-based approaches for anomaly detection in time series, such as HOT SAX and its improved variants are still much more efficient than LSTM-based method.

As for future work, we intend to evaluate the two comparative methods on more time series datasets. Besides, we plan to improve further the effectiveness and time efficiency of LSTM-based method in time series anomaly detection by applying a systematic technique in tuning hyperparameters of LSTM-based predictors [30].

References

1. Keogh, E., Lin, J., Fu, A.: HOT SAX: Efficiently finding the most unusual time series subsequence. In: Proceedings of The Fifth IEEE International Conference on Data mining (ICDM), pp. 226–233, (2005)
2. Bu, Y., Leung, T.W., Fu, A., Keogh, E., Pei, J., Meshkin, S.: WAT: Finding top-K discords in time series database. In: Proceedings of the 2007 SIAM International Conference on Data Mining (SDM' 07), Minneapolis, MN, USA, 26–28 (2007)
3. Li, G., Braysy, O., Jiang, L., Wu, Z., Wang, Y.: Finding time series discord based on bit representation clustering. Knowl. Based- Syst. **54**, 243–254 (2013)
4. Oliveira, A.L.I., Meira, S.R.L.: Detecting novelties in time series through neural networks forecasting with robust confidence intervals. Neurocomputing **70**(1–3), 79–92 (2006)
5. Pena, E.H.M., de Assis, M.V.O.M., Proença Jr., M.L.: Anomaly detection using forecasting methods ARIMA and HWDS, In: Proceedings of 32nd International Conference of Chilean Computer Science Society (SCCC), Temuco, Chile, pp. 11–15 (2013)
6. Yu, Y., Zhu, Y., Li, S., Wan, D.: Time series outlier detection based on sliding window prediction. Math. Problems Eng. **2014**, 879736, (2014)
7. Siami-Namini, S., Tavakoli, N., Sinam-Namin, A.: A comparison of ARIMA and LSTM in forecasting time series, In: Proceedings of 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 17–20, Orlando, FL, USA, pp. 1394–1401 (2018)
8. Sezer, O.B., Gudelek, M.U., Ozbayoglu, A.M.: Financial time series forecasting with deep learning: a systematic literature review: 2005–2019, Appl. Soft Comput. **90**, 106181 (2020)
9. Han, Z., Zhao, J., Leung, H., Ma, K.F., Wang, W.: A review of deep learning models for time series forecasting, IEEE Sens. J. **21**(6), 7833–7848, (2021)
10. Lindermann, B., Muller, T., Vietz, H., Jazdi, N., Weyrich, M.: A survey on long short term memory networks for time series prediction. Procedia CIRP **99**, 650–655 (2021)
11. Lindemann, B., Maschler, B., Sahlab, N. and Weyric, M.: A survey on anomaly detection for technical systems using LSTM networks. Comput. Ind. **131**, 103498 (2021)
12. Malhotra, P., Vig, L., Shroff, G., Agarwal, P.: Long short term memory networks for anomaly detection in time series. In: Proceedings of European Symposium on Artificial Neural Networks (ESANN), Bruges (Belgium), 22–24 April, pp. 89–94 (2015)
13. Chauhan, S. Vig, L.: Anomaly detection in ecg time signals via deep long short-term memory networks. In: Proceedings of 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Paris, France, 19–21 October, pp. 1–7 (2015)
14. Buda, T., Caglayan, B. Assem, H.: Deepad: A generic framework based on deep learning for time series anomaly detection. In: Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2018), LNCS 10937, Springer, Cham, pp. 577–588 (2018)
15. Zhang, L., Yang, L., Gu, C. Li, D.: Lstm-based short-term electrical load forecasting and anomaly correction, E3S Web of Conferences **182** (01004) (2020)
16. Chandola, V., Cheboli, D. K., Kumar, V.: Detecting anomalies in a time series database, Technical Report, Department of Computer Science and Engineering, University of Minnesota, TR-09–004 (2009)
17. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Comput. **9**(8), 1735–1780 (1997)
18. Ben Taieb, S., Bontempi, G., Atiya, A. F. Sorjamaa, A.: A review and comparison of strategies for multi-step ahead time series forecasting based on the {NN5} forecasting competition. Expert Syst. Appl. **39**(8), 7067–7083 (2012)
19. Keogh, E., Chakrabatti, K., Pazzani, M.: Dimensionality reduction for fast similarity search in large time series databases. Knowl. Inf. Syst. **3**, 263–286 (2001)

20. Lin, J., Keogh, E., Lonardi, S. Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms, In: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, DMKD 2003, pp. 2–11 (2003)
21. Buu, H.T.Q., Anh, D.T.: Time series discord discovery based on iSAX symbolic representation. In: Proceedings of the 3rd International Conference on Knowledge and Systems Engineering (KSE), Hanoi, Vietnam, 14–17 October, pp. 11–18 (2011)
22. Thuy, H.T.T., Anh, D.T., Chau, V.T.N.: An effective and efficient hash-based algorithm for time series discord discovery, In: Proceedings of the 2nd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS 2016), September 14–16, Da Nang, Vietnam, pp. 85–90 (2016)
23. Anh, D.T. Hien, N.V.: A dynamic programming approach for time series discord detection, In: Proceedings of 10th EAI International Conference on Context-Aware Systems and Applications (ICCASA 2021), Virtual Event, 28-Oct, pp. 255–266 (2021)
24. Thuy, H.T.T., Anh, D.T., Chau, V.T.N.: Novel method for time series anomaly detection based on segmentation and clustering. In: Proceedings of 10th International Conference on Knowledge and System Engineering (KSE), IEEE, Ho Chi Minh City, Vietnam, 1–3 November, pp. 276–281 (2018)
25. Cholett, F.: Keras. http://keras.io. Accessed 2021
26. The UCR Time Series Dataset Archive for Discord Detection http:/www.cs.ucr.edu/~eamonn/discords/. Accessed 2021
27. Lavin, A., Ahmad, S.: Evaluating real-time anomaly detection algorithms: the numenta anomaly benchmark. In: Proceedings of IEEE 14th International Conference on Machine Learning and Applications (ICMLA), Miami, Florida, USA, 2–11 December (2015)
28. Thuy, H.T.T., Anh, D.T., Chau, V.T.N.: Some segmentation-based techniques to improve time series discord discovery, In: Proceedings of International Conference on Nature of Computation and Communication (ICCTC 2016), March 17–18, Rach Gia, Vietnam, LNICST 128, Springer, pp. 179–188 (2016)
29. Kingma, D. B., Ba, J.: Adam: a method for stochastic optimization, arXiv preprint arXiv :14126 980 (2014)
30. Abbasimehr, H., Shabani, M., Yousefi, M.: An optimized model using LSTM network for demand forecasting. Comput. Ind. Eng. **143**, 106435 (2020)