Univariate Time Series Data Correction Method using LSTM Autoencoder with Temporal Distance

Sohyeon Yun
Department of Electrical and Computer Engineering
University of Seoul
Seoul, Korea
jasmine2161@naver.com

Han-joon Kim
Department of Electrical and Computer Engineering
University of Seoul
Seoul, Korea
khj@uos.ac.kr

***Abstract* — High-quality data is essential to increase the reliability of machine learning-based prediction models. For time series data, anomalies significantly reduce the accuracy of prediction models. In this paper, we propose a novel time series data correction method that converts abnormal values of univariate time series data into normal ones. For anomaly detection and correction, we utilize the LSTM Autoencoder model, where we propose a new weight function that considers temporal distance. Through experiments using the open NAB data, we show that our proposed method is superior to the recent conventional methods.**

Keywords* — *time series data, LSTM Autoencoder, anomaly detection, data correction, deep learning, data quality

I. Introduction

Recently, manufacturing industries have used sensors to collect data and generate AI-based models using that data to realize particular prediction services. These prediction models are used to analyze data from factory equipment to prevent emergencies such as fires and malfunctions [1]. However, in the process of collecting data through sensors, data anomalies may occur in the data for a variety of reasons [2]. Detecting and correcting these anomalies is a top priority to increase the reliability of the AI-based prediction models [3].

In our work, we have used the LSTM Autoencoder model to detect and correct anomalies in univariate time series data. In terms of correcting the detected anomalies, we propose a new correction method that considers the temporal relation of the time series data. Basically, anomaly correction is achieved by replacing detected anomalies with normal values. The key idea is to find a *normal* data window containing the data values best suited for replacement. For this, we propose a new similarity formula that has a weighting function that reflects temporal distance. To prove the effectiveness of the proposed method, we have performed experiments using the open Numenta Anomaly Benchmark (NAB) dataset [4].

II. Methods

After detecting anomalies in the given data, corrections must be made. The time series data correction process consists of two parts; anomaly detection and data correction; following the process, data correction is performed to replace the abnormal values with normal values [5].

1) A normal window is located to replace the anomaly window. Here, to find a normal window, we utilize a similarity formula with a weight function.

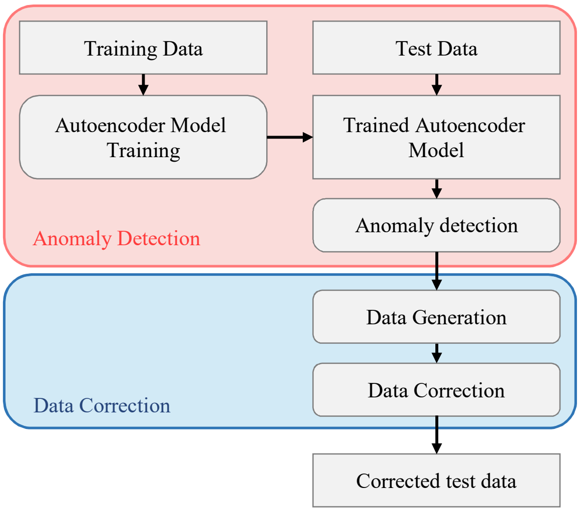

Fig. 1. A process of anomaly detection and correction for time series data

2) Using the LSTM Autoencoder model, a window similar to located one is generated.

3) The generated normal window replaces the anomaly window.

Fig. 1 illustrates the overall process of the proposed correction method.

A. Anomaly Detection

For anomaly detection on time series data, we have used the LSTM Autoencoder model which consists of an encoder and a decoder. Basically, the encoder compresses inherent features within the sequential data, and the decoder generates data similar to real data by using those features. Here, the encoder and decoder can contain LSTM networks to learn temporal characteristics within time series data. Fig. 2 shows the architecture of the LSTM Autoencoder for anomaly detection on time series data.

The object function of the LSTM Autoencoder model is defined as follows [6]:

$$\operatorname{argmin}_{x'} \sum_{X \in s_N} \sum_{i=1}^{L} \|x_i - x_i'\|^2 \quad (1)$$

In Eq (1), s_N denotes the total values of time series data and X is a set of some values defined as a 'window'. Also, L is the number entire value that are in each window, x_i is the real value and x_i' is the value generated by the Autoencoder. The model is trained to minimize the mean squared error(MSE)

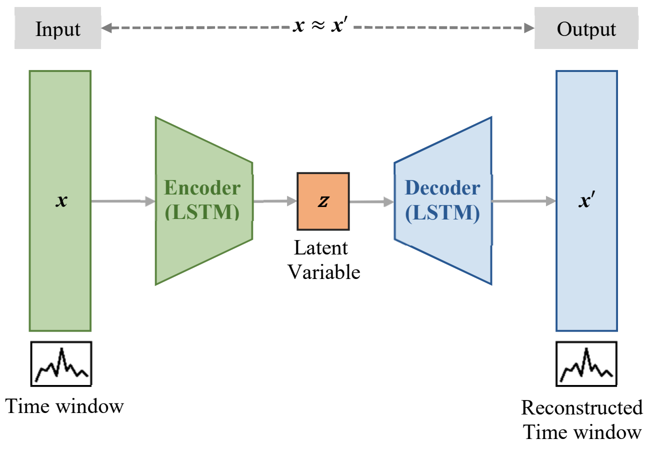

Fig. 2. LSTM Autoencoder model for anomaly detection on time series data

of the difference between the input data and the output data.

To detect anomalies within given time series data, we define an anomaly score function using the error vector. The error vector $e_i = |x_i - x_i'|$ is assumed to follow a normal distribution, and we need to estimate its mean μ and covariance Σ. Then the resulting anomaly score a_i is defined as Eq. (2):

$$\text{Anomaly score}(a_i) = (e_i - \mu)^{\mathrm{T}} \Sigma^{-1}(e_i - \mu) \qquad (2)$$

If the distance between the error vector and the assumed data distribution increases, we can consider that there is an anomaly; that is, if the anomaly score of a certain window is higher than a threshold (τ), it is detected as an anomaly.

Fig. 3 shows an example of anomaly detection using the anomaly score in Eq (2); two peaks where the anomaly score is above the threshold are considered as anomalies. Fig. 4 shows the result of mapping the anomalies detected by the anomaly score to the real data. In the shaded areas including anomalies, it can be seen that the anomalies have been well detected.

B. *Data window search using a similarity formula*

As mentioned earlier, we try to convert detected anomalies (i.e., abnormal values) to appropriate values on a window basis. For this, we intend to utilize the information (such as *periodicity* and *time continuity*) of the normal window. The overall process of anomaly correction is as follows.

1) Find the target window that is earlier than the anomaly window (in terms of *time continuity*).

2) Find the normal window that is most similar to the target window (in term of *periodicity*).

3) Replace the detected abnormal window with the normal window by using the information in the window next to the located normal window.

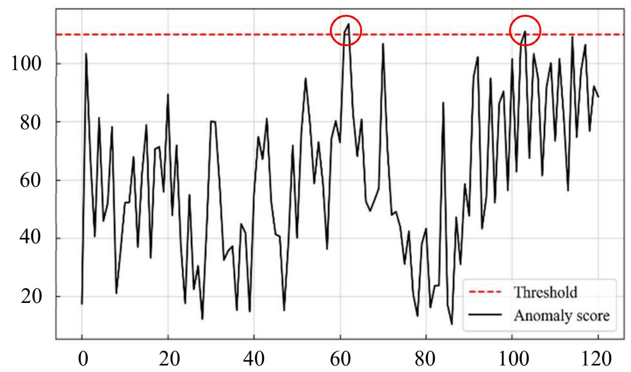

Fig. 3. An example of anomaly detection with anomaly score. The black and red lines denotes the anomaly score and the threshold, respectively. The red circles represents the points that are identified as anomalies.

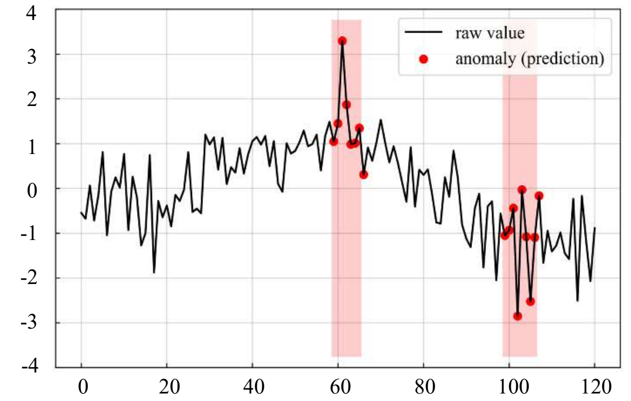

Fig. 4. Result of mapping the detected anomalies to the test data. The black line denotes the real data, and the red dots denotes the anomalies detected with on the anomaly score. There are real anomalies (i.e., ground truths) in the shaded area.

Since time series data is time continuous, the data values at a particular point in time are strongly influenced by the data values at the nearest point in time. Therefore, we define a window adjacent to the window containing the abnormal value as the *target* window. Then, considering the periodicity, we can find the *normal* window with the most similar pattern to the target window for the entire time series data.

In this paper, we define a new similarity formula for detecting similar windows, as shown in Eq (3).

$$\text{Weighted cosine similarity SIM } (T, S_i) \qquad (3)$$
$$= \frac{\sum_{i=1}^{n} (T \cdot S_i)}{\|T\| \cdot \|S_i\|} + \alpha\left(\frac{\sum_{i=1}^{n} (T \cdot S_i) * \mathrm{W}(i)}{\|T\| \cdot \|S_i\|}\right)$$

$$\mathrm{W}(i) = \frac{1}{1 + d(T, S_i)^2} \qquad (4)$$

In Eq (3), T is the target window and S_i is the search window i. The goal is to find the window S_i that is most similar to T throughout the time series data. $W(i)$ is defined as in Eq (4), which is a function that gives a larger weight windows closer to the target window. The $d(T, S_i)$ in Eq (4) is the distance function between the target window T and the search window S_i. This distance function is defined by taking into account the periodicity of time series data, and operates in a way that the closer it is to the reference window, the greater the importance of the window is emphasized, and the farther away it is, the greater the window's influence is reduced. The α is used to control the rate at which periodicity and temporal continuity are considered.

ALGORITHM FOR TIME SERIES DATA CORRECTION

```
Function Correction(X)
Input: A test set of time series windows X
Output: A set of corrected time series windows X
    AE ← autoencoder model trained with training dataset
    N ← number of windows
    for i in 1 to N do
        Idx ← AnomalyDetection (X[i])
        if Idx == Anomaly
            X[i] ← DataGeneration(X[i])
        end if
    end for
    return X
end Function
Function AnomalyDetection (x)
    A ← anomalyScore(x, AE[x])
    if A > α then
        Idx ← Anomaly
    else A < α
        Idx ← Normal
    end if
    Return Idx
end Function
Function DataGeneration(x)
    n ← number of consecutive anomaly window
    j ← SimilarIndexDetection(x) + n
    x′ ← AE.encoder(X[j])
    return x′
end Function
Function SimilarIndexDetection(x)
    for i in 1 to N do // refer to Eq. (3) & (4)
        $S \leftarrow \mathrm{argmax}_{X[i]}\,(\cos(X[i],x)+\alpha(\cos(X[i],x)\cdot W(i)))$
    end for
    return S
end Function
```

* For the *anomalyScore* function, refer to [6].

Fig. 5. The algorithm for time series data correction

C. Autoencoder-based time series data correction

After finding the window that most closely resembles the target window for anomaly correction in time series data, the next step is to generate and correct the data using the window information at the next point in time. The following two examples show the correction process; one is when there is a single anomaly window, and the other is when there are consecutive anomalies.

As in the example in Fig. 6, assume that window 16 is detected as an abnormal window. First, window 15 at the previous point is set as the target window (T). Next, search window (S) number 7, which has the most similar pattern to the target window (T), is found. Then, by using the information within window 8 at the next time point, similar data is generated. The newly generated normal value replaces window 16 which contains the anomaly. Moreover, as in the example in Fig. 7, cases where abnormal windows are continuous may occur. As shown in this figure, if windows 16, 17, and 18 are detected as abnormal windows, search window number 7 (S), which has the most similar pattern to window number 15 (T) at the previous time, is found. Then, similar data is generated by using the information within windows 8, 9, and 10, and the generated normal value replaces the anomaly value, as in the case of Fig. 6. In our work, to generate similar data, we used a pre-trained LSTM autoencoder model for anomaly detection.

Fig. 5 shows the pseudocode that performs the anomaly detection and data correction process based on the LSTM Autoencoder model. The *Correction* function is the main function that performs the correction process. First, it builds an LSTM Autoencoder model with prepared training dataset and then identify any abnormalities on a window-by-window basis through the *AnomalyDetection* function. At this time, if a particular window have an anomaly score higher than α through the *anomalyScore* function, then it is regarded as 'Anomaly'. When a window with 'Anomaly' data is detected, the *DataGeneration* function is called, which replaces an abnormal 'Anomaly' windows with automatically generated normal windows. Here, to find the optimal window that can replace the abnormal window, the *SimilarIndexDetection* function is called. Then, a new window similar to the window of the next time of the optimal window is generated by the LSTM autoencoder; that is, the *Correction* function replaces the window containing the normal values with an abnormal window. Fig. 8 shows the results of anomaly data being corrected by the data correction algorithm.

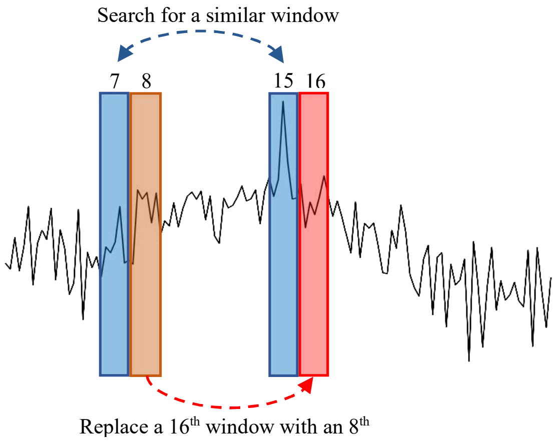
Search for a similar window
7
8
15
16
Replace a 16th window with an 8th

Fig. 6. Time series data correction within an anomaly window

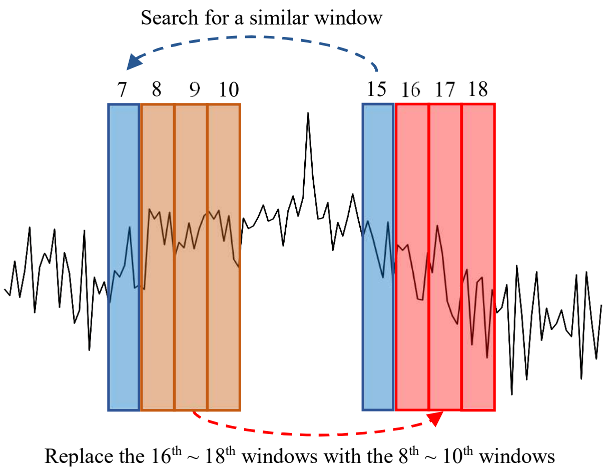
Search for a similar window
7
8
9
10
15
16
17
18
Replace the 16th ~ 18th windows with the 8th ~ 10th windows

Fig. 7. Data correction for consecutive anomaly windows

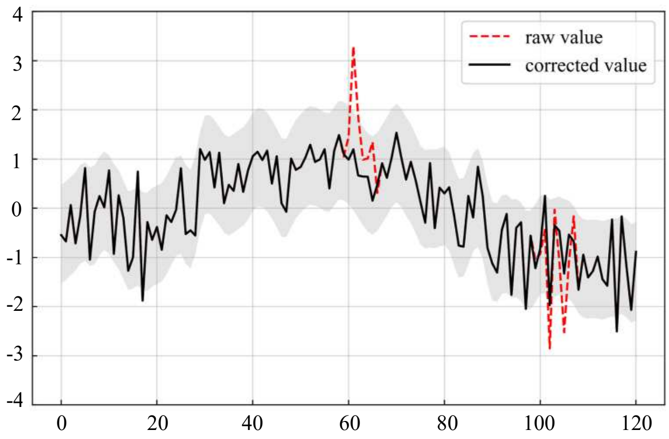

Fig. 8. Comparison of the raw data and corrected data. The red dotted line denotes the raw values that include anomalies, and the black line denotes the corrected values.

III. Results

In our experiments, we intend to evaluate the effectiveness of our proposed correction method indirectly through performance evaluation of prediction models that are developed with time series data corrected by our proposed method.

A. Data setup

For our performance evaluation, we have used the "Ambient temperature system failure" dataset (including anomaly labels) from the NAB (Numenta Anomaly Benchmark) collection at *Kaggle.com*. This dataset consists of 7,267 hourly data points collected over 328 days from a single sensor. When composing training data and test data, we organized the data in window units, and one window consists of 60 time steps. In addition, to create a more accurate anomaly detection and correction model, we performed min-max normalization on the data.

Table I shows the performance of anomaly detection in terms of precision, recall, F1 score, and accuracy with the LSTM Autoencoder model, before correcting the detected anomalies to normal values.

TABLE I. Anomaly Detection Performance

Evaluation Metric	**Value**
Precision	0.882
Recall	1.0
F$_1$-measure	0.937
Accuracy	0.983

TABLE II. Performance of Predictive Models for Raw Data and Corrected Data

Method	**LSTM**		**GRU**	
	MAE	***MSE***	***MAE***	***MSE***
Raw data	0.551	0.065	0.625	0.055
Mean-based correction	0.902	0.074	0.890	0.084
SVR-based correction	0.489	0.057	0.477	0.056
GAN-based correction	0.486	0.055	0.468	0.056
Proposed correction	**0.457**	**0.047**	**0.460**	**0.050**

B. Performance evaluationof data correction

To evaluate the effectiveness of the proposed time series data correction method, we evaluated prediction models using the corrected data; the prediction models were built up by using LSTM and GRU algorithms., and their training data are prepared from both raw data and corrected data. The proposed correction method was compared with mean-based, Support Vector Regression (SVR) [7], and Generative Adversarial Neural Network (GAN) [8].

As performance metrics for time series data correction, we adopted the mean absolute error (MAE) and mean squared error (MSE); the smaller these values are, the better the performance of the prediction model. Table II shows the performance of the LSTM and GRU prediction models on the respective data. As seen in the table, the prediction model for the data corrected by the proposed method has the best performance with a 20% improvement over the original data.

IV. Conclusion

In this paper, we proposed a new anomaly correction method for univariate time series data using an LSTM Autoencoder model and proved its effectiveness through experiments using NAB dataset. As future work, we plan to develop a deep learning architecture that can learn the correlation and dependence among variables for multivariate time series data correction.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea

(NRF) funded by the Ministry of Education (NRF-2022R1A2C1011937).

References

[1] W. Mao, W. Wang, Z. Dou, and Y. Li, “Fire recognition based on multi-channel convolutional neural network”, Fire technology, Vol. 54, No. 3, pp. 531-554, 2018.

[2] S. Zhanwei, and Z. Liu, “Abnormal detection method of industrial control system based on behavior model”, Computers & Security, 84, 2019, pp. 166-178, 2019.

[3] M. C. Douglas, “Introduction to statistical quality control”, John Wiley & Sons, 2020.

[4] A. Lavin and S. Ahmad, “Evaluating Real-Time Anomaly Detection Algorithms -- The Numenta Anomaly Benchmark”, Proceedings of 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), pp. 38–44, 2015.

[5] H. Jeong, H. J. Kim, “A Dynamic Correction Technique of Time-Series Data using Anomaly Detection Model based on LSTM-GAN”, The Institute of Internet, Broadcasting and Communication, Vol.23, No.2, pp.103-111, 2023.

[6] P. Malhotra, R. Anusha, A. Gaurangi, V. Lovekesh, and S. Gautam. “LSTM-based encoder-decoder for multi-sensor anomaly detection”, arXiv preprint arXiv:1607.00148, 2016.

[7] M.-K. Lee, S.-H. Moon, Y.-H. Kim, and B.-R. Moon, “Correcting abnormalities in meteorological data by machine learning”, Proceedings of 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 888–893, 2014.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, and Y. Bengio, “Generative adversarial networks”, Communications of the ACM, Vol. 63, No. 11, pp. 139-144. 2020.