THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 22, Number 3/2021, pp. 291–298

DEEP NEURAL NETWORKS FOR DETECTION OF ABNORMAL TREND IN ELECTRICITY DATA

Jian ZHENG, Jianfeng WANG, Jiang LI, Shuping CHEN, Lei SHU, Yike PENG

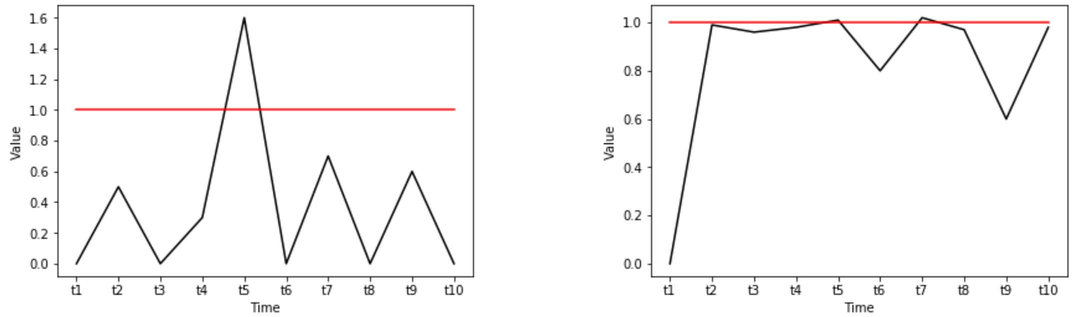
Chongqing Aerospace Polytechnic, Chongqing 40021, P.R. China
E-mail: zhengjian.002@163.com

Abstract. Electricity data not only demonstrates electricity consumption of different time in different region, but also reflects the trend of electricity consumption in different time. Thus, the detection of anomalous electricity data is of great significance. Currently, anomaly detection methods focus on mining the anomaly in the time series data, and few of them study the trend of time series data. Hence, it is a tough task for exploring an advanced approach to predict anomaly trend targeted to electricity data. To address this issue, here proposed a deep neural network to mine anomaly and to predict the anomaly trend in electricity data by focusing on the degree of change to electricity consumption. To predict anomaly trend more accurately, the three trend respecting the change of electricity consumption between time points, i.e., rising, falling and constancy, are considered during calculating the cost function. Experimental results on the real electricity data show the proposed method is higher accuracy compared to the state-of-the-art anomaly detection methods as for predicting the anomaly trend and detecting anomalies for electricity consumption. We find that the ability that neural networks predict the anomaly trend in regard to electricity data can be promoted through this manner of using the cost function to calculate the degree of change to electricity consumption. Our findings also indicate that the prediction to the change of anomaly trend in regard to the time series data using multiple-layers neural network approach outperforms that of using hybrid methods consisting of the state-of-the-art anomaly detection approaches.

Key words*:* anomaly trend, neural networks, time series data.

1. INTRODUCTION

Electricity data changes along with time and power consumption, showing features of time series data. Electricity data is much valuable information, because of reflecting the trend of the electricity consumption in different time, it is of great significance to mine anomalies in electricity data. Fig. 1 displays examples of anomalies to time series data. This easily mines that point at T_5 is an anomaly as shown in Fig. 1a. Unfortunately, it is difficult to find the anomaly trend caused by the point at T_5. For another case, when the anomaly features are not obvious, anomalies are also difficult to mine, e.g., in Fig. 1b, anomaly features are similar to normal features.



a) anomaly features are significant b) anomaly features are not significant

Fig. 1 – Anomalies in time series data.

Anomaly mining refers to discover objects of being different from most data [1, 2]. Using traditional approaches to mine anomalies indicates that the loss has occurred. While for the time series data, it is an ordered set of column observations recorded in chronological order, according to the trend of data over time, this is more values to predict the occurrence of anomalies in time.

The anomalies of time series data refer to the point where the pattern in the sequence is inconsistent, such as sudden rise or fall, trend change, exceeding the historical, maximum or minimum value. The detection of abnormal trends respecting the time series data aims to quickly and accurately find existing abnormal points, and predict which points are abnormal based on the anomaly detection results [3, 4, 5]. Hence, Anomaly mining and trend analysis are essential for the study of the time series data. Recently, many studies have presented important contributions in regard to anomaly detection to the time series data. Existing methods of anomaly detection about the time series data can be classified the following categories [6, 7]. (I) Classification-based, such as, Support Vector Machine (SVM) [8, 9], and One Class-Support Vector Machine (OC-SVM) [10]. The complexity and high-dimensionality of data cause the limitation of mining abnormal features using SVM [11]. Moreover, these methods are greatly influenced by data distribution of the sample. Similarly, in [12], the nonlinear modeling approach is used for canonical correlation analysis, through analyzing these canonical correlation between data, this help guide detection algorithms. (II) Clustering-based, e.g., the method in [13], in [14] and in [15] use the manner of clustering to detect anomalies. Such method (unsupervised manner) does not rely on data label, but it depends too much on the parametric setting of models. While for method (i) and (ii), they are prone to analysis the anomalies of the time series data, and few of them study the trend of time series data. (III) Prediction-based, such method adopts machine learning manners for anomaly detection [16]. For example, in [17], a two-step anomaly detection method is used for time series data. S.Ahmad et al. [18] propose a hierarchical temporal memory method to mining anomalies in time series data. As well as, the methods in [19] and in [20] detect anomalies in multivariate time series data using deep architectures approaches. In addition, the generative adversarial networks (GANs) [21] are also widely applied for anomaly detection in time series data, similarly, in [22] and in [23]. These methods judge anomalies via assessing the difference between the predicted value of the data at the time point in the normal mode and the actual value. Indeed, it is difficulty for the estimation of models [24]. Beyond that, the Multivariate Bayesian Structural Time Series (MBSTS) model in [25] is used to predict the prices of a portfolio of stocks. Similar to this hybrid manner, e.g., the hybrid method in [26]. Through the above literature analyzing, for the time series data, it is a challenge for proposing an advanced method to predict anomaly trend.

In this work, our motivation is to predict the anomaly trend in electricity data. However, we aim at proposing an advanced method to explore the anomaly trend targeted to the time series data. To achieve our studied goals, here designs a deep neural network to predict the change of anomaly trend in electricity data. To more accurately predict anomaly trends, the three trends, i.e., rising, falling and constancy, in regard to the electricity consumption change between time points are considered when calculating the cost function in the proposed deep neural network. In addition, the 2-norm is also used to optimize the cost function. Finally, the proposed model is tested and validated comprehensively on real electricity datasets.

We summarize the main contributions of this work as follows:

(1) The multiple-hidden layers neural network is designed, which is used to predict the anomaly trend targeted to electricity data via focusing on the degree of change to electricity consumption.

(2) The prediction ability of neural networks to the anomaly trend respecting electricity data is promoted through this manner of using the cost function to calculate the degree of change in electricity consumption.

(3) For the prediction to time series data, the desired prediction results is easier to obtain using the multiple-layers neural networks method instead of using hybrid methods consisting of the state-of-the-art anomaly detection approaches.

2. METHOD

In this section, we firstly analyze the three trends about the change of electricity consumption. Through calculating the three trends, this provides a theoretical basis for our model to predict anomaly trend. Then, the proposed neural network is designed in subsection 2.2, including description of the model architecture, selection of hyper parameters, and the model training.

2.1. Analysis of three trends

The change of electricity consumption can be divided into three trends between time t_i and time t_{i+1}, including rising, falling and constancy. Whereas, the three trends can hardly reflect the degree of change to electricity consumption, as such, the degree of change to electricity consumption needs to be quantified.

Let us assume that given the time series data of electricity consumption $TD = \{<t_1, D_1>, <t_2, D_2>, \dots, <t_i, D_i>, \dots, <t_n, D_n>\}$, and $1 \le i \le n$, the item $<t_i, D_i>$ represents that electricity consumption is D_i at t_i time. $T_{i,i+1}$ expresses the degree of the trend about electricity consumption from time t_i to time t_{i+1}. Hence, $T_{i,i+1}$ can be defined as following:

$$T_{i,i+1} = \frac{D_{i+1} - D_i}{t_{i+1} - t_i}, \quad 1 \le i \le n. \tag{1}$$

Equation (1) indicates that electricity consumption increases if $T_{i,i+1}$ is greater than zero. If $T_{i,i+1}$ is lower than zero, electricity consumption decreases. Obviously, $T_{i,i+1}$ is equal to zero , which means is constant for electricity consumption. Using these series $\{T_{1,2}, T_{2,3}, \dots, T_{i,i+1}, \dots, T_{n,n+1}\}$ can describe the trend of electricity consumption between any two adjacent time points.

2.2. Model description

(1) *Model architecture.* A multiple-hidden layer neural network is proposed, denoted as m-NN, $m \ge 1$, where m is the m-th hidden layer of m-NN. The proposed m-NN is formulated in detail as follows.

– Input layer. In the layer, the input $x_{in}(t)$ is the original time series data, i.e., $x_{in}(t) = TD$.

– Multiple-hidden layers. Hidden layers include the encoding hidden-layers of capturing the useful representations and the decoding hidden-layers of the reconstructed input. The two types of architectures are designed as following, respectively.

For the encoding hidden-layers, the input $H_m^{in}(t,l)$ and the output $h_m^{out}(t,l)$ of the m-th hidden layer in the l-th iteration are calculated using Eq. (2) and Eq. (3), respectively, having

$$h_m^{out} = act_m^{en}(\mathbf{w_m}(l) H_m^{in}(t,l) + \mathbf{b_m}(l)) \tag{2}$$

$$H_m^{in}(t,l) = h_{m-1}^{out}(t,l). \tag{3}$$

As regards the decoding hidden-layer, correspondingly, the corresponding output of the k-th hidden layer in the l-th iteration are calculated, as following

$$h_k^{out} = act_k^{de}(\mathbf{w_k}(l) H_k^{in}(t,l) + \mathbf{b_k}(l)) \tag{4}$$

$$H_k^{in}(t,l) = h_{k-1}^{out}(t,l), \tag{5}$$

where the items act_m^{en} and act_k^{de} are activation functions in the encoding hidden-layer and the decoding hidden-layer. **w** and **b** are the weights and bias of each hidden layer.

– Output layer. The reconstructed $\hat{x}_{in}(t,l)$ is used as final output.

– Cost function. The cost function $J(\mathbf{w},\mathbf{b})$ of m-NN consists of the mean square error (mse), the item Θ and the L_2 regulation term $\Omega_{weights}$ [27], as following

$$J(\mathbf{w},\mathbf{b}) = \mathrm{mse} + \alpha * \Theta + \lambda * \Omega_{weights} \tag{6}$$

$$\mathrm{mse} = \frac{1}{n} \sum_{t=1}^{n} \| x_{in}(t) - \hat{x}_{in}(t,j) \|^2 \tag{7}$$

$$\Theta = \frac{\sum_{i,j=1}^{n} |T_{i,i+1} - T_{j,j+1}|}{n}, \quad 1 \le i, j \le n \tag{8}$$

where α and λ are trade-off parameters, and $0<\alpha<1$, $0<\lambda<1$. The item Θ is used to assess the degree of change to electricity consumption at time t_i and time t_j, thereby predicting the anomaly trend. $T_{i,i+1}$ and $T_{j,j+1}$ are calculated using Eq. (1). The item $\Omega_{weights}$ is used to optimize the weight parameter of *m*-NN.

(2) *Hyper parameters.* Some hyper parameters of *m*-NN need to be considered, including the number of neurons, neuron weight and activation function. The number of neurons in *m*-th hidden layer, denoted as e_m, is determined by cross-validation by varying from $e_m = e_1$ to e_2 with a step of Δe. Let $e_1=10$, $e_2=30$, and $\Delta e=2$. Given those data volume and data dimensionality, the number of neurons is sufficiently large, but not too large. While for neuron weight, a relatively small initial value is given. Using this manner to consider the number and weight of neurons is to prevent from causing over-fitting due to their excessive values. In addition, the function Leaky ReLu, i.e., $\mathrm{g}(x)=\max(0,x)+leak*\min(0,x)$, is used as the activation function of *m*-NN, where the item *leak* is a constant. Compared with other activation functions, e.g., ReLu, tanh and Sigmoid, etc, Leaky ReLu can obtain a very small gradient when the input *x* is not greater than zero, which is beneficial to prevent from gradient vanishing. Certainly, for the other parameters in *m*-NN, they have no substantial effects on the results, so their default values are adopted.

(3) *Model Training.* Training a neural network is to train its hyper parameters, so as to get the optimal parameters. Through monitoring some observed indicators, such as loss error, training precision, etc, those parameters can be adjusted in time for scientific training during model training. Consequently, for the *m*-NN training, we dynamically adjust the iteration epoch based on the observed training accuracy. Until *m*-NN converges, the training of the model stops.

Table 1

Algorithm 1

1 Initialization hyper parameters, learning rate, etc, parameters, Q, P, e_1, e_2, weight ;
2 Input sample dataset $X=\{x \mid x_1, \dots, x_N\}$;
3 X_{train} is gotten by random selecting 80% of X ; /* training set */
4 X_{test} is gotten by random selecting 10% of X ; /* testing set */
5 X_{pre} is gotten by random selecting 10% of X ; /* prediction set */
6 Randomly select 80% of X_{train} to obtain *Train_set* ; /* Cross-validated training set */
7 Cross-validation set $Cross\_set = X_{train} - Train\_set$; /* Cross-validated testing set */
8 **for** p=1 to P **do**:
9   **for** $e_m^p = e_1$ to e_2 with step Δe **do**:
10   Use gradient descent method to train a neural network m-NN$_p$ for cross-validation;
11   Calculate training accuracy $TrAcc(e_m^p) = m\text{-NN}_p(Train\_set;\, e_m^p)$;
12   Calculate cross-validation accuracy $CroAcc(e_m^p) = m\text{-NN}_p(Cross\_set;\, e_m^p)$;
13   **end for**
14   **for** $e_m^p = e_1$ to e_2 with step Δe **do**:
15   Calculate average cross-validation accuracy $Avg\_CroAcc(e_m^p) = \left(\sum_{p=1}^{p=P} CroAcc(e_m^p)\right)/P$;
16   **end for**
17 **end for**
18 Get the optimal parameter e_m between e_1 and e_2 $Opt(e_m) = \arg\max_e (Avg\_CroAcc(e_m^p))$;
19 **for** q=1 to Q do:
20   Use gradient descent method and $Opt(e_m)$ to train a neural network m-NN$_q$;
21   Calculate training accuracy $TraAcc(q) = m\text{-NN}_q(X_{train};\, Opt(e_m))$;
22   Test the neural network *m*-NN;
23   Calculate testing accuracy $TeAcc(q) = m\text{-NN}(X_{test})$;
24 **end for**
25 Select q so that $q_{max} = \arg\max_q (TraAcc(q))$;
26 Get the maximum training accuracy $TraAcc = m\text{-NN}_q(X_{train};\, Opt(e_m),\, q_{max})$ in the q_{max}-th iteration;
27 Get the maximum testing accuracy $TeAcc = m\text{-NN}_q(X_{test},\, q_{max})$;
28 Predict anomaly trend using the neural network m-NN$_q$;
29 Calculate prediction accuracy $PreAcc = m\text{-NN}_q(X_{pre})$;
30 Output *TraAcc*, *TeAcc*, *PreAcc*;

Table 1 displays the overall algorithm. In Algorithm 1, some hyper parameters are firstly initialized, e.g., parameters, Q, P, e_1, e_2, and the division of sample set is given in between step1 to step 7. Then the cross-validation is implemented from step8 to step18, obtaining the optimal parameter e_m, denoted as $Opt(e_m)$.

After getting the $Opt(e_m)$, m-NN is trained q times until it converges, as shown from step19 to step24. After m-NN is trained, the testing set and prediction set are inputted, respectively. Finally, the maximum training accuracy, testing accuracy and prediction accuracy are outputted as shown from step25 to step30.

3. EXPERIMENTAL SETTINGS

In this section, experimental datasets, comparison methods and assessment metrics are given for the subsequent experiments. In addition, experimental procedure is also described in detail.

3.1. Dataset

In order to verify our method, we carry out experiments on real electric consumption data provided by California, USA [28]. Each record indicates the actual electric consumption at a specific moment. Hence, this dataset is considered to be a time series dataset. We opt for the data of 100 days of actual electricity consumption as an experimental dataset. Because the experimental dataset does not exist anomaly change trend, we select randomly the data of 10 days from it, and replace the origin value by using the y ($y > 0$) times of origin value to compose the experimental dataset included anomalies.

3.2. Comparison methods and assessment metrics

To address a fair assessment to our method, the Hierarchical Temporal Memory (HTM) [17] method and the Two-Step Anomaly Detection (MA + SARIMA) [18] method are used for comparison. To have a fair conclusion, for the competing methods, their optimal parameters observed in the corresponding literature are used. Unless otherwise stated, all experiments run on the same GPU, using the same environment.

The Receiver Operating Characteristic curve (ROC) and the corresponding area under the curve (AUC) are used to assess the accuracy of methods. To minimize the random effects, all experiments are run 100 times and then the results are averaged for evaluation.

3.3. Experimental description

Two sets of experiments were implemented to test the ability of our method. The detailed description is as follows.

The dataset is randomly divided into three parts, of which, 80% of the data, i.e., training set, is used for model training, and 10% of the data, i.e., testing set, is used to test the capabilities of model to mine anomalies. The remaining data is used as the prediction set, which verifies the predictive ability to models.

Experiment 1 (see Section 4.1). To test the structure for m-NN. Let m be equal to 1, 2, 3, … , 20, respectively, m-NN runs on the training set and the testing set, observing the impact of different hidden-layer scale on the results.

Experiment 2 (see Section 4.2). To test the capabilities that the three methods (our method and two comparison methods) mine anomalies and predict anomaly trends. The three methods are trained using the training set. And their capabilities to mine anomalies are verified using the testing set. The prediction set is used for verifying the precision to predict anomaly trends. Then, experimental results are observed.

4. RESULTS

All experimental results show that the proposed m-NN outperforms the competing methods in the accuracy to mine anomalies and to predict anomaly trend. Section down below detailed experimental results.

4.1. Experiments on network structure

For the prediction ability of neural networks, network structure may reach their peak performance by different selection hidden-layer scale. Hence, this subsection focuses on the testing of network structure.

Let m set in the range of 1 to 20, and the results on datasets are shown in Fig. 2. As m increases, the performance of increases and remains stable when m reaches a certain size, i.e., $m = 3$. Fig. 3 displays the loss error of the training and testing procedure when m is equal to 3. These results given in Fig. 2 and Fig. 3 indicate that m-NN is stable. Hence, let m be equal to 3 in the m-NN for subsequent experiments.

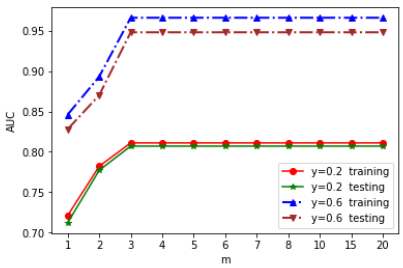


Fig. 2 – Validation structures.

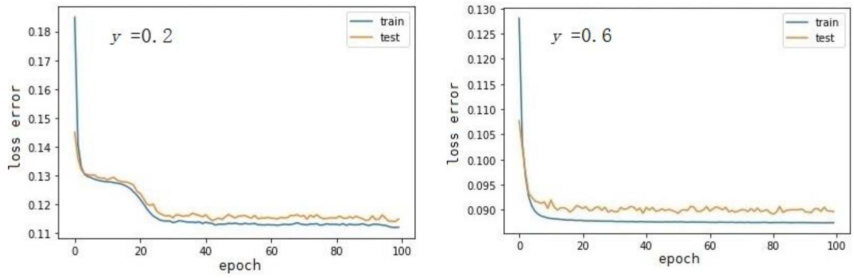
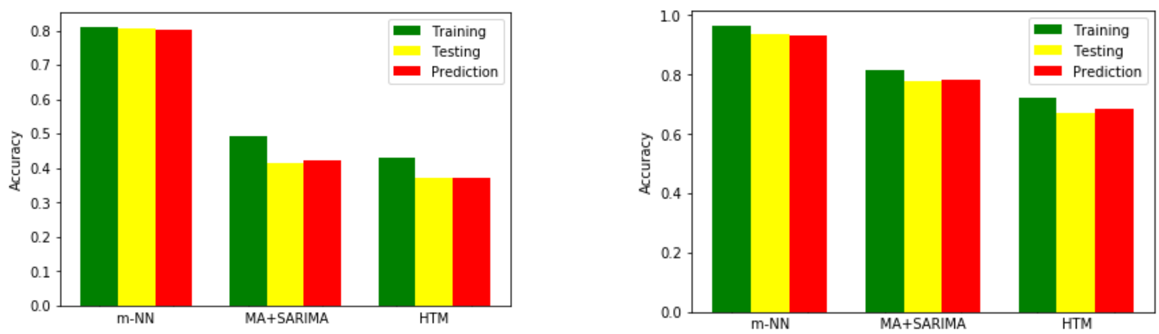


Fig. 3 – Loss error on $y = 0.2$ and $y = 0.6$.

4.2. Precision comparison

Figure 4 displays the mining accuracy to experimental methods using different y value dataset. Results show that our method outperforms the competing methods. It can be seen in Fig. 4a that our method reaches above 0.8 in the precision of training and testing, as well as prediction precision. Even if anomaly features are not obvious, i.e., y is equal to 0.2. While for the competing methods, their precision is less than 0.5. Similarly, in Fig. 4b, when y is equal to 0.6, i.e., anomaly features are more obvious, the precision to all methods augments, while our method obtains desired results, and is superior to these competitors. This indicates that m-NN has advantages to mine the limited number of potential anomalies and to predict anomaly trend targeted to time series data.



(a) $y = 0.2$ (b) $y = 0.6$

Fig. 4 – Results of AUCs using different y value: a) y value is equal to 0.2; b) y value is equal to 0.6.

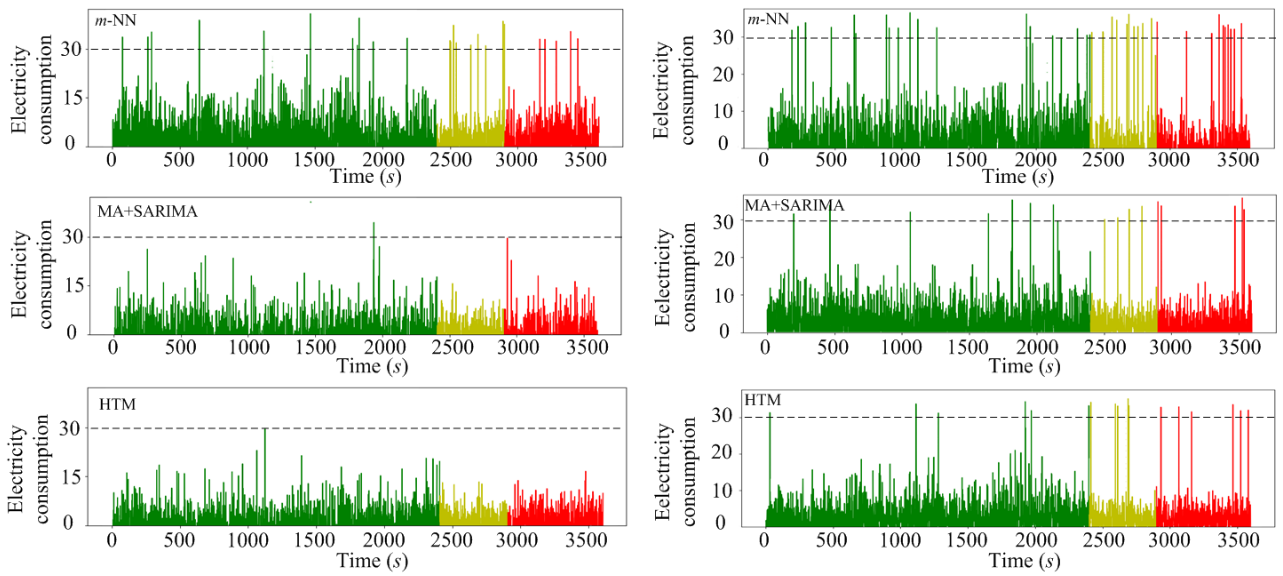
Figure 5 visualizes these results. When the anomaly features are not obvious, our method predict the anomaly trend with high accuracy, as shown in red area in Fig. 5a. Unfortunately, most the competing approaches are almost failure to predict. As anomaly features become obvious, all methods predict the anomaly trends, nevertheless, the prediction precision to our method is still higher than that of the competitors, as shown in red area in Fig. 5b.

From Fig. 4 and Fig. 5, several observations can be obtained:

(i) For the time series data, our method outperforms the competing approaches in the precision to mine anomalies and to predict the anomaly trend. This is because three trends respecting the degree of change to electricity consumption are fully considered during calculating the cost function of m-NN, thus improving the accuracy.

(ii) The capability that neural networks predict the anomaly trend in electricity data can be increased through this manner of using the cost function to calculate the degree of change in electricity consumption.

(iii) For the prediction to the time series data, using neural network approaches are easier to obtain the desired results instead of using hybrid methods consisting of the state-of-the-art anomaly detection methods.



(a) $y = 0.2$ (b) $y = 0.6$

Fig. 5 – Results of anomaly trend using different *y* value. Training results are marked as green. Testing results are marked as yellow. Prediction results are marked as red.: a) *y* is equal to 0.2; b) *y* is equal to 0.6.

5. CONCLUSION

In this work, in order to predict the anomaly trend in electricity data, a deep neural network was proposed based on time series data. Experiments on the real data of electricity consumption show that the proposed method outperforms the competitors in predict anomaly trend and mining anomaly. In the future works, we will explore the method respecting predict anomaly trend.

ACKNOWLEDGMENTS

This research is supported by the Science and Technology Research Program of Chongqing Municipal Education Commission of China (Grant No KJQN201903003). And the Science and Technology Research Program of Chongqing Municipal Education Commission of China (Grant No KJQN201903002).

REFERENCES

1. T. KUREMOTO, S. KIMURA, K. KOBAYASHI, M. OBAYASHI, *Time series forecasting using a deep belief network with restricted Boltzmann machines*, Neurocomputing, **137**, pp. 47-56, 2014.
2. J.F. WU, W.D. JIN, P. TANG, *Summary of monitoring techniques for data anomalies*, Computer Science, **44**, *b11*, pp. 24-28, 2017.
3. D. CHOUDHARY, A. KEJARIWAL, F. ORSINI, *On the runtime-efficacy trade-off of anomaly detection techniques for real-time streaming data*, arXiv: 1710.04735, 2017.
4. S. GUO, P. FENG, Z. ZHONG, W. LI, C. XU, M. YU, Y. YOU, Y. TANG, W. YAO, *Steady-state power quality anomaly recognition based on time series trend*, 2020 IEEE International Conference on Energy Internet (ICEI), 2020.
5. A.A. COOK, G. MISIRLI, Z. FAN, *Anomaly detection for IoT time-series data: A survey*, IEEE Internet of Things Journal, **7**, *7*, pp. 6481-6494, 2019.
6. H. IZAKIAN, W. PEDRYCZ, *Abnormal detection and characterization in spatial time series data: A cluster-centric approach*, IEEE Transactions on Fuzzy Systems, **22**, *6*, pp. 1612-1624, 2014.
7. T.T. DANG, H.Y.T. NGAN, W. LIU, *Distance-based k-nearest neighbors outlier detection method in large-scale traffic data*, Proceedings of the 2015 IEEE International Conference on Digital Signal Processing, pp. 507-510, 2015.
8. R.K. SEVAKULA, N.K. VERMA, *Clustering based outlier detection in fuzzy SVM*, Proceedings of the 2014 IEEE International Conference on Fuzzy System, pp. 1172-1177, 2014.
9. H. MARTINS, L.B. PALMA, A. CARDOSO, P. GIL, *A support vector machine based technique for online detection of outliers in transient time series*, Proceedings of the 2015 10th Asian Control Conference, pp. 1-6, 2015.

10. S.M. ERFANI, S. RAJASEGARAR, S. KARUNASEKERA, C. LECKIE, *High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning*, Pattern Recognition, **58**, pp. 121–134, 2016.
11. T. ERGEN, A.H. MIRZA, S.S. KOZAT, *Unsupervised and semi-supervised anomaly detection with LSTM neural networks*, arXiv: 1710.09207, 2017.
12. R. ZALL, M.R. KANGAVARI, *On the construction of multi-relational classifier based on canonical correlation analysis*, International Journal of Advanced Intelligence (IJAI), **17**, *2*, pp. 23-43, 2019.
13. E. MANZOOR, S.M. MILAJERDI, L. AKOGLU, *Fast memory-efficient anomaly detection in streaming heterogeneous graphs*, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1035-1044, 2016.
14. D. DENG, *Research on anomaly detection method based on DBSCAN clustering algorithm*, 2020 5th International Conference on Information Science, Computer Technology and Transportation (ISCTT), pp. 439-442, 2020.
15. Y. HUANG, Y. XUE, Y. SU, S. HAN, *Hyperspectral anomaly detection based on isolation forest with band clustering*, 2020 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2020.
16. M.U. AHMED, S. BRICKMAN, A. DENGG, N. FASTH, M. MIHAJLOVIC, J. NORMAN, *A machine learning approach to classify pedestrians events based on IMU and GPS*, International Journal of Artificial Intelligence, **17**, pp. 154-167, 2019.
17. R.E. SPERL, S.M. CHUNG, *Two-step anomaly detection for time series data*, International Conference on Data and Software Engineering (ICoDSE), 2019.
18. S. AHMAD, A. LAVIN, S. PURDY, Z. AGHA, *Unsupervised real-time anomaly detection for streaming data*, Neurocomputing, **262**, pp. 137-147, 2017.
19. C. ZHANG, D. SONG, Y. CHEN, X. FENG, C. LUMEZANU, W. CHENG, J. NI, B. ZONG, H. CHEN, N.V. CHAWLA, *A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data*, Proceedings of the AAAI Conference on Artificial Intelligence, 2019.
20. D. SHAH, S. KHADE, S. PAWAR, *Anomaly detection in time series data of Sensex and Nifty50 with Keras*, 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), 2021.
21. D. LI, D. CHEN, J. GOH, S.K. NG, *Anomaly detection with generative adversarial networks for multivariate time series*, arXiv: 1809.04758, 2018.
22. D. LI, D. CHEN, B. JIN, L. SHI, J. GOH, S.K. NG, *MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks*, arXiv: 1901.04997, 2019.
23. T. SCHLEGL, P. SEEBÖCK, S.M. WALDSTEIN, U. SCHMIDT-ERFURTH, G. LANGS, *Unsupervised anomaly detection with generative adversarial networks to guide marker discovery*, International Conference on Information Processing in Medical Imaging, 2017.
24. H. YANG, Y. WANG, S.F. TAGNE, Q. HUANG, *Abnormal pre-detection algorithm based on time series*, 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService), pp. 262-265, 2019.
25. S.R. JAMMALAMADAKA, J. QIU, N. NING, *Predicting a stock portfolio with the multivariate Bayesian structural time series model: Do news or emotions matter?*, International Journal of Artificial Intelligence, **17**, *2*, pp. 81-104, 2019.
26. U.L. YUHANA, N.Z. FANANI, E.M. YUNIARNO, S. ROCHIMAH, L.T. KOCZY, M.H. PURNOMO, *Combining fuzzy signature and rough sets approach for predicting the minimum passing level of competency achievement*, International Journal of Advanced Intelligence(IJAI), 2020.
27. B.A. OLSHAUSEN, D.J. FIELD, *Sparse coding with an over complete basis set: A strategy employed by V1*, Vision Research, **37**, pp. 3311-3325, 1997.
28. *Electric consumption data in California, USA (2019-2020)*, http://www.engieresources.com/

Received July 10, 2021