

Backpropagation Neural Network with Combination of Activation Functions for Inbound Traffic Prediction

Purnawansyah^{a,1,*}, Havaluddin^{b,2}, Herdianti Darwis^{a,3}, Huzain Azis^{a,4}, Yulita Salim^{a,5}

^a Department of informatics, Universitas Muslim Indonesia
Jl. Urip Sumoharjo KM 5, Makassar, 90231 Indonesia

^b Department of informatics, Mulawarman University
Jl. Kuaro No.1, Samarinda, 75123 Indonesia

¹ purnawansyah@umi.ac.id *; ² havaluddin@gmail.com; ³ herdianti.darwis@umi.ac.id;

⁴ huzain.azis@umi.ac.id; ⁵ yulita.salim@umi.ac.id

* corresponding author

ARTICLE INFO

Article history:

Received 26 March 2021

Revised 12 April 2021

Accepted 09 June 2021

Published online 17 August 2021

Keywords:

Backpropagation

Combination of activation function

Forecasting

Inbound Traffic

ReLu-Sigmoid

ReLu-tanh

ABSTRACT

Predicting network traffic is crucial for preventing congestion and gaining superior quality of network services. This research aims to use backpropagation to predict the inbound level to understand and determine internet usage. The architecture consists of one input layer, two hidden layers, and one output layer. The study compares three activation functions: sigmoid, rectified linear unit (ReLU), and hyperbolic Tangent (tanh). Three learning rates: 0.1, 0.5, and 0.9 represent low, moderate, and high rates, respectively. Based on the result, in terms of a single form of activation function, although sigmoid provides the least RMSE and MSE values, the ReLU function is more superior in learning the high traffic pattern with a learning rate of 0.9. In addition, ReLU is more powerful to be used in the first order in terms of combination. Hence, combining a high learning rate and pure ReLU, ReLU-sigmoid, or ReLU-Tanh is more suitable and recommended to predict upper traffic utilization.

This is an open access article under the CC BY-SA license
(<https://creativecommons.org/licenses/by-sa/4.0/>).

I. Introduction

Numerous research has been conducted regarding traffic measurements, whether in terms of traffic patterns, volumes, applications, and user activity characteristics [1][2]. Predicting network traffic is one of the crucial works to perform when it comes to network management as a consideration for admission and congestion control, anomaly detection, and bandwidth allocation to gain superior quality of service and cost reduction. Traffic itself could be formed in two dimensions, what time people actively engage to the internet and how much the work capacity the users engage where the two matters could be presented in one series of traffic data. Time series is data focusing on the recorded values for a given measurement at several points when the data is expressed as y_1, y_2, \dots, y_n y_i denotes value measured at a time i [3].

In dealing with time series forecasting, conventional statistical methods are popularly used, such as ARIMA and its various models, decomposition, and Winter's exponential smoothing [4], Hidden Markov Model [5], and threshold autoregressive (TAR) [6]. Besides, in machine learning, the neural network has been widely developed to deal with network data prediction [7][8][9][10]. Furthermore, a hybrid method has been extensively conducted, particularly for time series forecasting, such as a hybrid of neural network and ARIMA by [11][12], and a hybrid of HMM and multilayer perceptron [13].

Backpropagation neural networks itself is one of multilayer perceptron algorithm that has been widely studied for forecasting and classifying various cases, including an analysis was successfully performed by [14] in modeling ten risky factors of traffic accidents to elderly female and male drivers in West Midlands of the UK, and in the estimation of nuclear accident source [15].

The architecture of backpropagation neural networks still becomes the preferable topic in neural networks, such as optimization in the number of hidden layers [16][17], and another research in sensors, modeling backpropagation based on GA by specifying the number of hidden layer neurons [18]. Despite its slow training, a backpropagation neural network is easy to use and design depending on the input characteristics, whether univariate or multivariate inputs [19]. Thus, backpropagation is proposed to predict inbound traffic to understand and determine the internet usage through the network. Three different activation functions are implemented, i.e., sigmoid, ReLU, and tanh function. The implementation is both in a single form and combination, making up nine permutation models to optimize the weights between layers.

II. Methodology

A. Backpropagation Neural Network

The neural network is a reliable nonlinear technique for modeling a wide range of applications due to the flexibility in terms of architecture. The neural network architecture could be two or more layers. Neural network applied in this study is backward propagation of errors or backpropagation, a supervised learning algorithm in neural networks, which is a multilayer perceptron. Backpropagation in the networks is simply a gradient descent method aiming to optimize the weights connecting the adjacent layers among the input layer, hidden layer(s), and output layer. By the optimized weights, the errors between the observed data and the prediction can be minimized.

Figure 1 shows a neural network 2-hidden-layer. This study uses this architecture since two hidden layers are more superior to those with one hidden layer [20]. This architecture was applied with dense networks, which means that each unit (node) in a layer is densely connected with all other units in the neighboring layers. Each connection is associated with a weight (w_{ij}) reflecting the strength of the connection between the units. The inputs of x_1, x_2, \dots, x_n the hidden unit value (h_j) is determined by applying a weighted sum of all inputs plus a bias as written in (1), while the output unit (y_i) is defined by (2) [21].

$$net(h_j|x) = f(\sum_{i=1} w_{ij} \cdot x_i + b) \quad (1)$$

$$net(y_i|h) = f(\sum_{j=1} w_{ij} \cdot h_j + b) \quad (2)$$

In a backpropagation neural network, firstly, the signal propagates forward from the input layer to the output layer through the hidden layer. After that, the error is calculated, moving vice versa from the output layer to the input layer through the hidden layer. After the iterative training process, the neural network achieves the optimal weight and threshold to reduce the error to the desired level [15]. The weight parameter is updated with the rate change as in (3), where y_i denotes the observed data while \hat{y}_i is the predicted values.

$$\Delta w_{ij} = \epsilon((y_i - \hat{y}_i)) \quad (3)$$

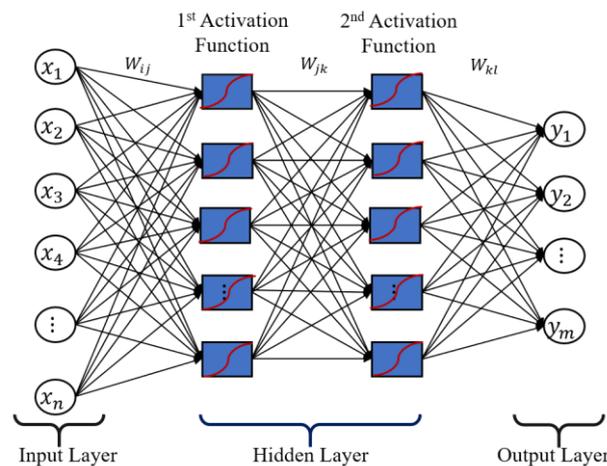


Fig. 1. Proposed Backpropagation Neural network architecture

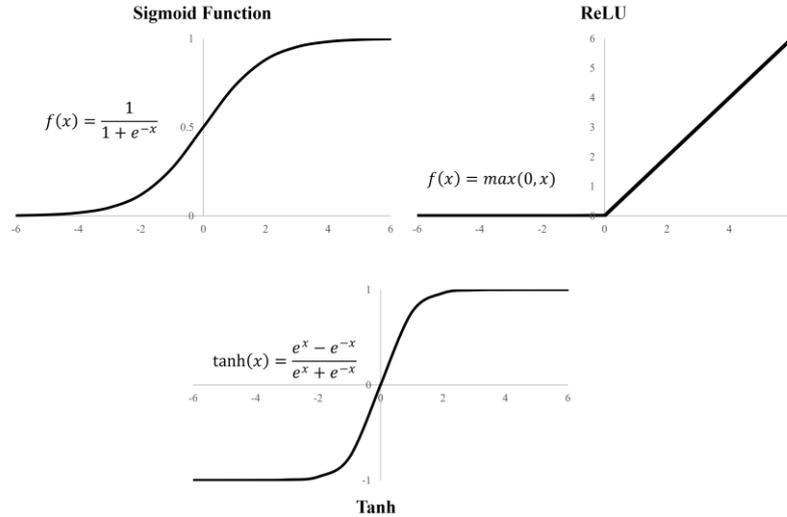


Fig. 2. Activation function; Sigmoid, ReLU, and Tanh

Furthermore, f itself is an activation function to map the values to a nonlinear (Figure 2). Sigmoid function is one of commonly used mainly for forecasting probability-based-output as expressed in (4), and ReLU is another function widely used representing a nearly linear function and preserving the properties of linear models that made them easy to optimize, with gradient-descent method [22] as given by (5). Besides, hyperbolic tangent, known as tanh function, is a zero-centered function that provides better training performance for multilayer neural networks formulated as in (6).

$$f(x) = \text{Sigmoid}(x) = \frac{1}{1+e^{-x}} \quad (4)$$

$$f(x) = \max(0, x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (5)$$

$$f(x) = \text{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (6)$$

B. Experimental Design

This paper collected a time series network inbound traffic data from a backbone network using CACTI and a traffic controller applied in Mulawarman University in Indonesia. The series was week-days inbound traffic accounted daily ranging from 27 August 2019 to 17 February 2021.

Traffic data measured in bits/second were then normalized on a scale of 0 to 1 to prevent huge numbers in the process of BPNN. The study uses the first 80% as training data, and the rest is testing data. Figure 3 illustrates the research flow implemented in this paper, whereas each hidden layer is an applied activation function. There were three various activation functions used, i.e., sigmoid, ReLU, and tanh function designed in a single form and combination, making up nine permutation models in terms of order, i.e., the usage of pure sigmoid function; ReLU function; tanh function; sigmoid-ReLU; sigmoid-tanh; ReLU-Sigmoid; ReLU-tanh; tanh-sigmoid; and tanh-ReLU function.

Furthermore, Table 1 depicts the preferable setting of BPNN utilized. In order to conduct a comparative analysis in the usage of learning rate, this paper was designed with three kinds of learning rate, i.e., 0.1; 0.5; and 0.9 reflecting a low, middle, and high rate, respectively.

C. Accuracy Metrics

In terms of accuracy comparison, mean square error (MSE) and root mean square error (RMSE) were used as expressed in (7) and (8) consecutively, where y_i denotes the observed data while \hat{y}_i is the predicted values. The smaller the value, the less the error is.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (7)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (8)$$

Table 1. Parameter setting of Backpropagation Neural Network

Parameters	Value
Maximum Iteration	3000
Momentum	0.9
Epoch	3000
Hidden Layer	2
Learning rate	0.1; 0.5; 0.9

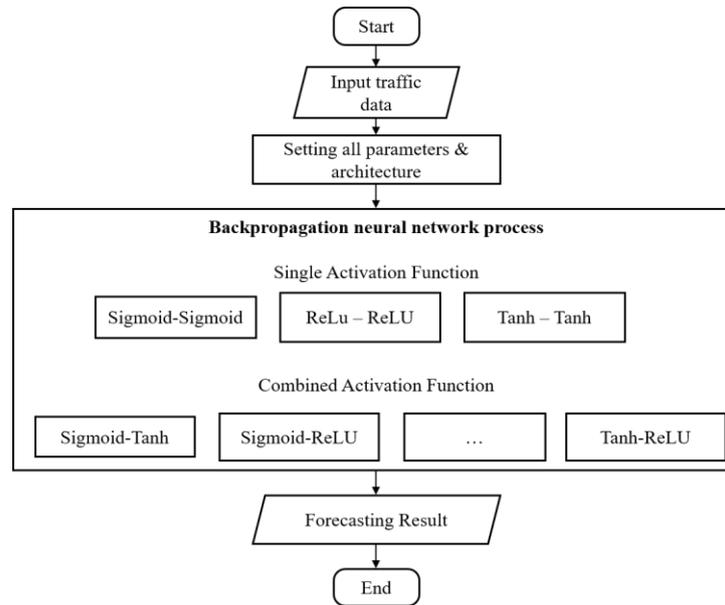


Fig. 3. Research flow

III. Results and Discussions

Nine permutation models of activation function were used in this paper with three various learning rates. Table 2 and Table 3 show the MSE and RMSE values for each model and learning rate used based on the simulation performed. Overall, it can be clearly seen that in terms of the usage of a single activation function, although the usage of pure sigmoid function provided the smallest RMSE, the RMSE gained from the three models were not significantly different.

The results of single form activation function are shown in Figure 4, Figure 5, and Figure 6 for Sigmoid, ReLU, and Tanh respectively. The usage of Sigmoid-Sigmoid and Tanh-Tanh function could not recognize higher traffic pattern. On the contrary, ReLU-ReLU worked more superior in terms of pattern recognition although the RMSE was not the smallest one, but it was still more superior than Tanh function.

Table 2. Parameter setting of Backpropagation Neural Network with single form activation function

Activation Function Form	Activation Function's Order	Learning Rate	MSE	RMSE
Single form Activation Function	Sigmoid – Sigmoid	0.1	0.01519463	0.12326648
		0.5	0.01474695	0.12143701
		0.9	0.01314681	0.11465952
	ReLU – ReLU	0.1	0.01679769	0.12960591
		0.5	0.01536174	0.12394247
		0.9	0.01373752	0.11720719
	Tanh – Tanh	0.1	0.01555421	0.12471651
		0.5	0.01402152	0.11841251
		0.9	0.01653229	0.12857797

Table 3. Parameter setting of Backpropagation Neural Network with combined activation function

Activation Function Form	Activation Function's Order	Learning Rate	MSE	RMSE
Combined Activation Function	Sigmoid – ReLU	0.1	0.01608174	0.12681380
		0.5	0.01583461	0.12583564
		0.9	0.01518564	0.12323002
	Sigmoid – Tanh	0.1	0.01629544	0.12765358
		0.5	0.01575635	0.12552430
		0.9	0.01553580	0.12464268
	ReLU – Sigmoid	0.1	0.01380587	0.11749840
		0.5	0.01430379	0.11959846
		0.9	0.01541954	0.12417544
	ReLU – Tanh	0.1	0.01376507	0.11732464
		0.5	0.01501619	0.12254058
		0.9	0.01799144	0.13413217
	Tanh – Sigmoid	0.1	0.01602616	0.12659446
		0.5	0.01476072	0.12149372
		0.9	0.01651042	0.12849286
	Tanh – ReLU	0.1	0.01481934	0.12173470
		0.5	0.01708461	0.13070810
		0.9	0.01541536	0.12415862

On the other hand, when it comes to combination form between two different activation functions in the architecture, sigmoid could not recognize the high pattern properly whether in a single form or combination as presented in [Figure 4](#), [Figure 7](#), and [Figure 8](#), unless it was mixed with ReLU, with ReLU placed in the first order as shown in [Figure 9](#).

ReLU is remained powerful whether combined with sigmoid or tanh function in the architecture by implementing it in the first order. Combined ReLU in the first order can be seen in [Figure 9](#) and [Figure 10](#).

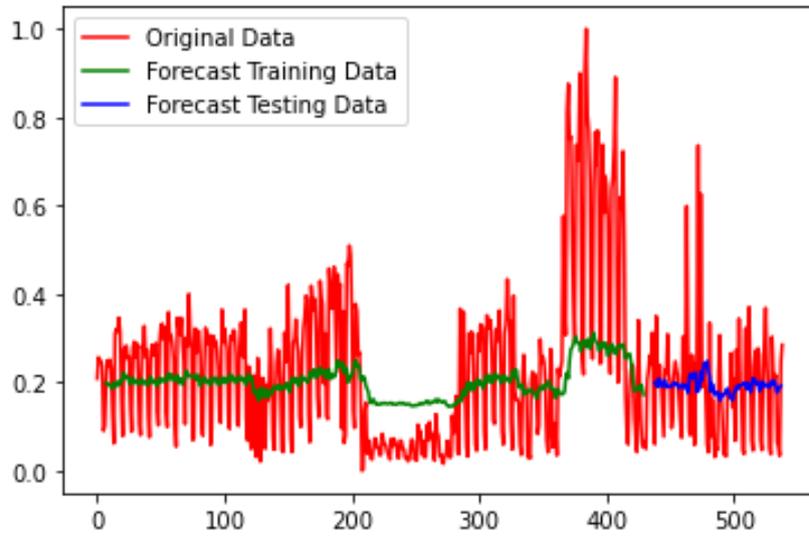
In terms of order, the performance of tanh function seems similar with ReLU that the accuracy result was not significantly changed whether in a single form usage or in combination as long as it was put in the first place then followed by other activation functions as illustrated in [Figure 11](#) and [Figure 12](#).

IV. Conclusion

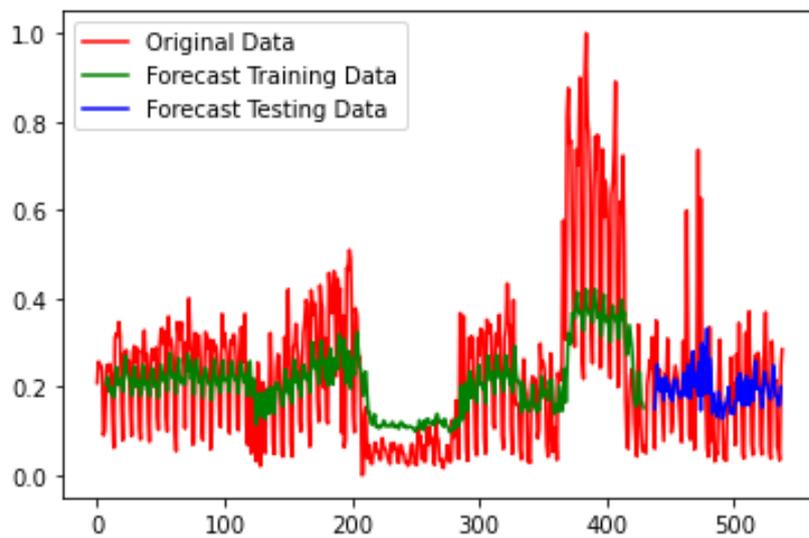
Backpropagation neural network is applied to predict the inbound traffic designed in one input layer, two hidden layers, and one output layer with three kinds of activation functions, i.e., sigmoid function, rectified linear unit (ReLU), and hyperbolic Tangent (tanh) function. The design is used in single and combination forms obtaining nine permutations with three kinds of learning rates, i.e., 0.1; 0.5; and 0.9 representing a low, middle, high rate.

Based on the result, it can be seen that ReLU works more superior in recognizing the inbound traffic pattern than sigmoid and tanh function in the similar architectures and parameters used in the analysis. Hence, an interesting result could be concluded that in regards to the usage of two different activation functions in BPNN architecture, the selection of first-order activation function is crucial in order to gain superior prediction result and ReLU function is recommended to be used in the initial order to catch the high pattern in the data. In addition, in terms of predicting upper traffic utilization, the combination of a high learning rate and pure ReLU or a combination of ReLU-sigmoid or ReLU-Tanh is more suitable and recommended.

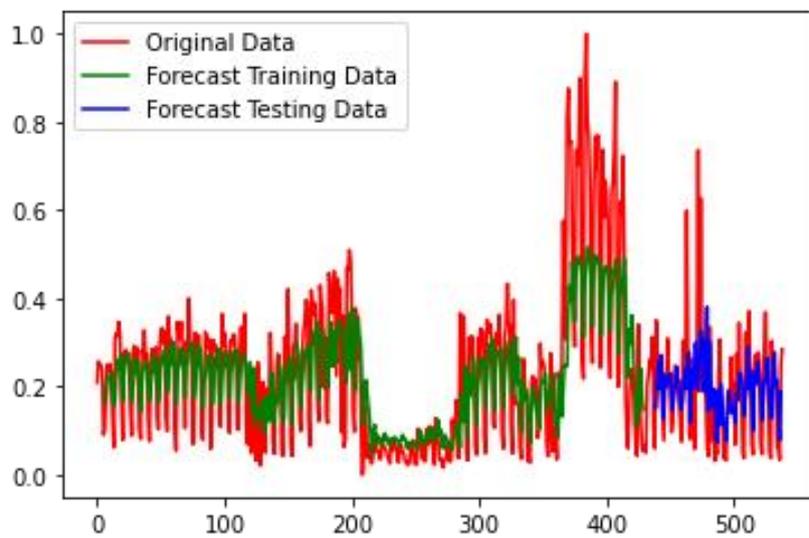
As for future work, it is recommended to optimize the architecture and parameters, particularly in the number of neurons in the hidden layers and learning rate, respectively. Nevertheless, overfitting and convergence could be problems encountered in the process so that a proper architecture, activation function's order, and parameter determination should be carefully performed.



(a)

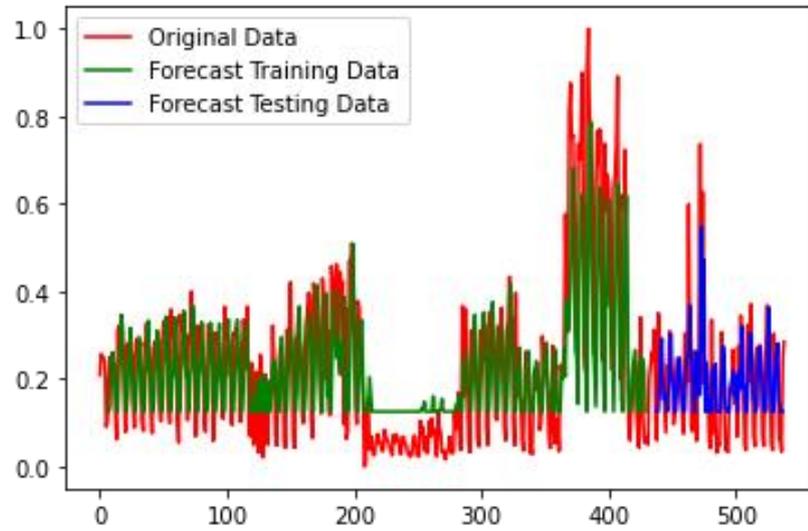


(b)

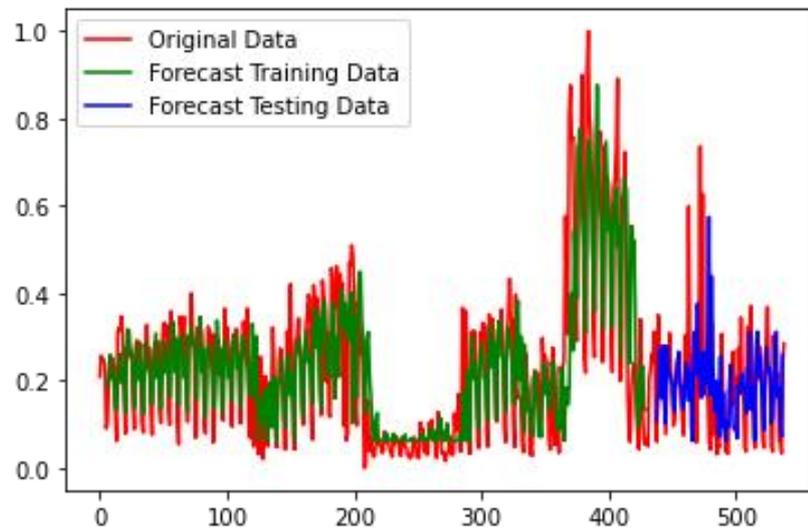


(c)

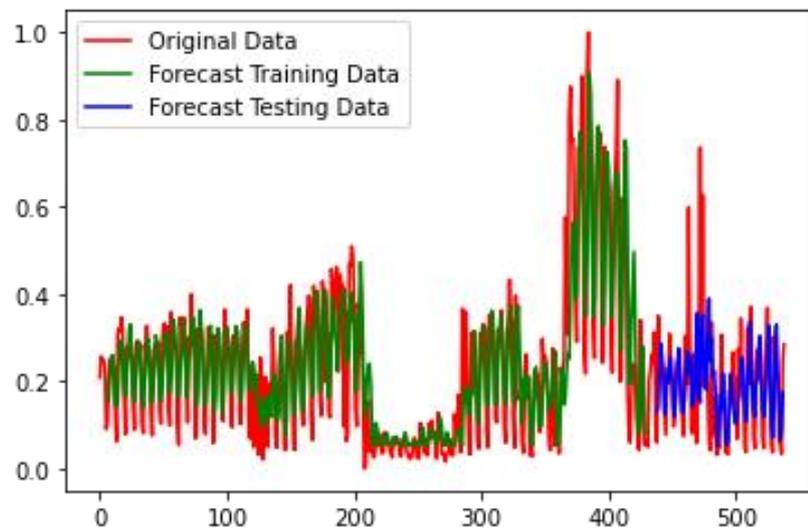
Fig. 4. Prediction result of single form activation function of Sigmoid function for each learning rate in BPNN: (a) learning rate 0.1, (b) learning rate 0.5, and (c) learning rate 0.9



(a)

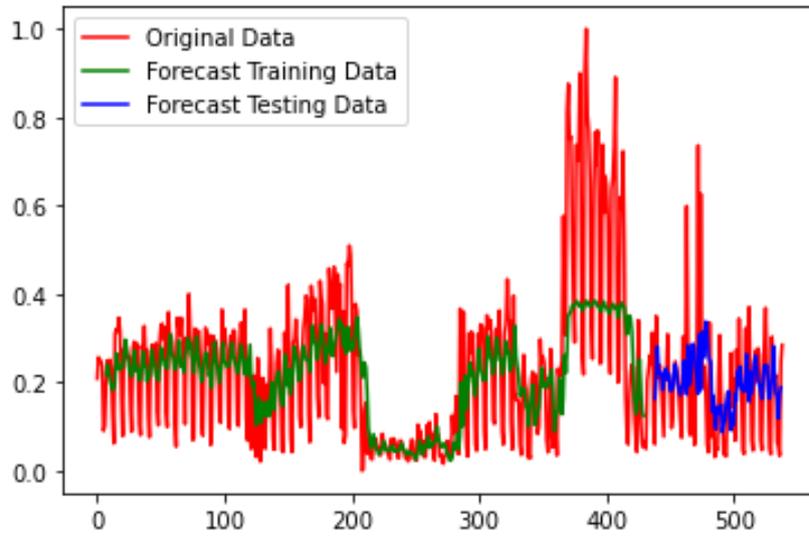


(b)

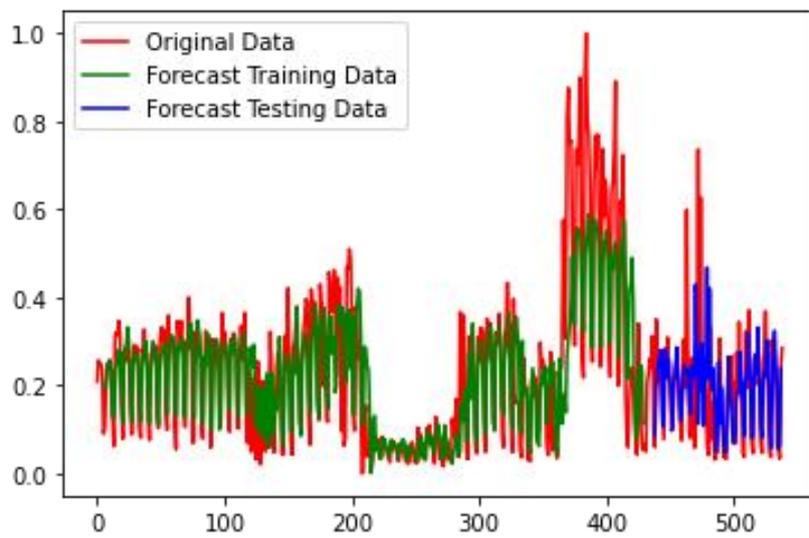


(c)

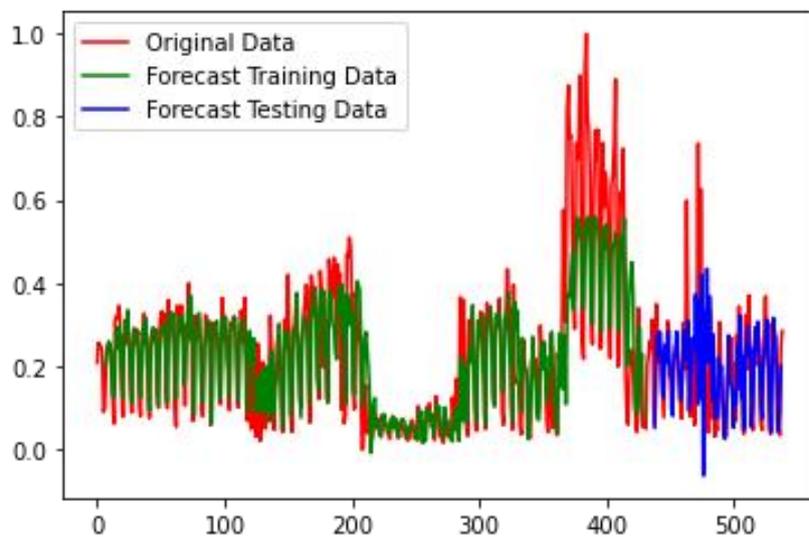
Fig. 5. Prediction result of single form activation function of ReLU function for each learning rate in BPNN: (a) learning rate 0.1, (b) learning rate 0.5, and (c) learning rate 0.9



(a)

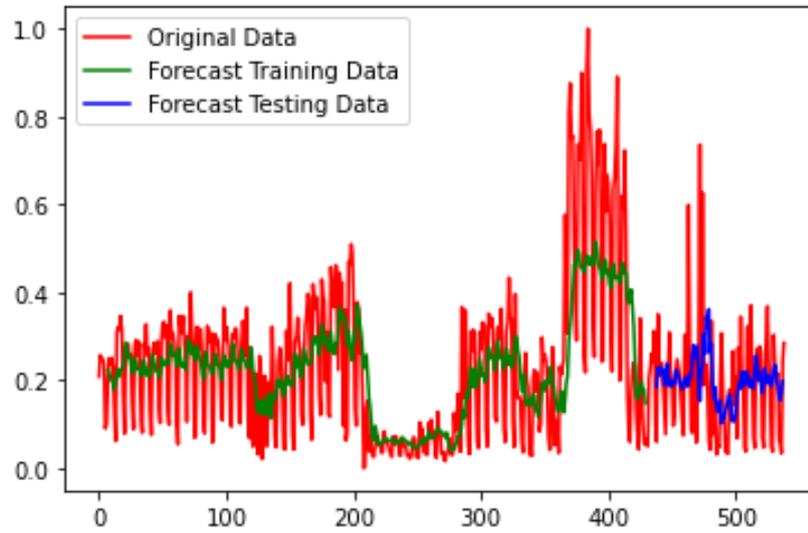


(b)

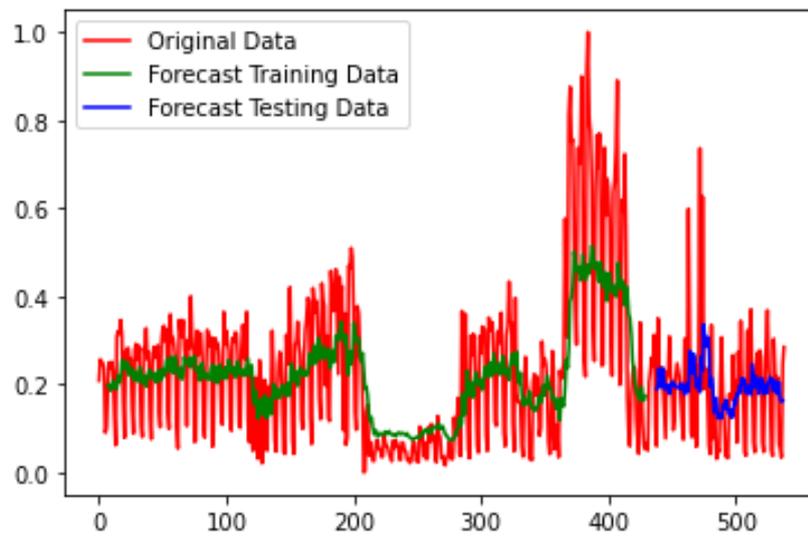


(c)

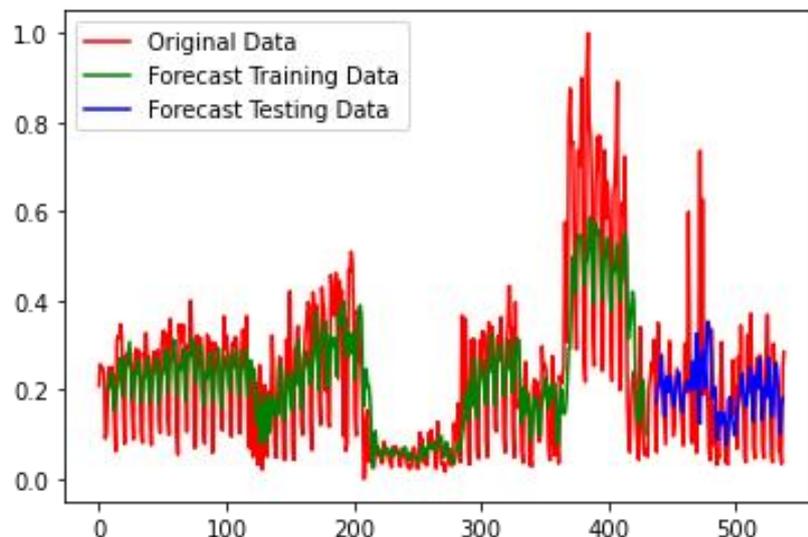
Fig. 6. Prediction result of single form activation function of Tanh function for each learning rate in BPNN: (a) learning rate 0.1, (b) learning rate 0.5, and (c) learning rate 0.9



(a)

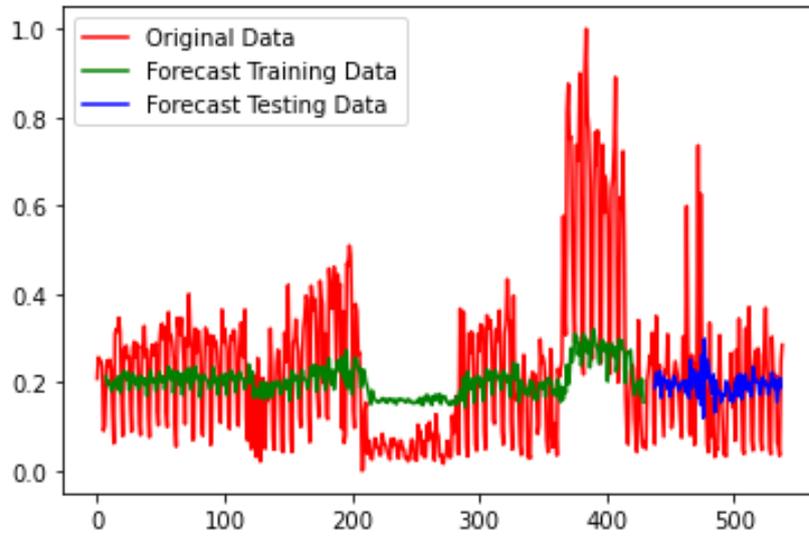


(b)

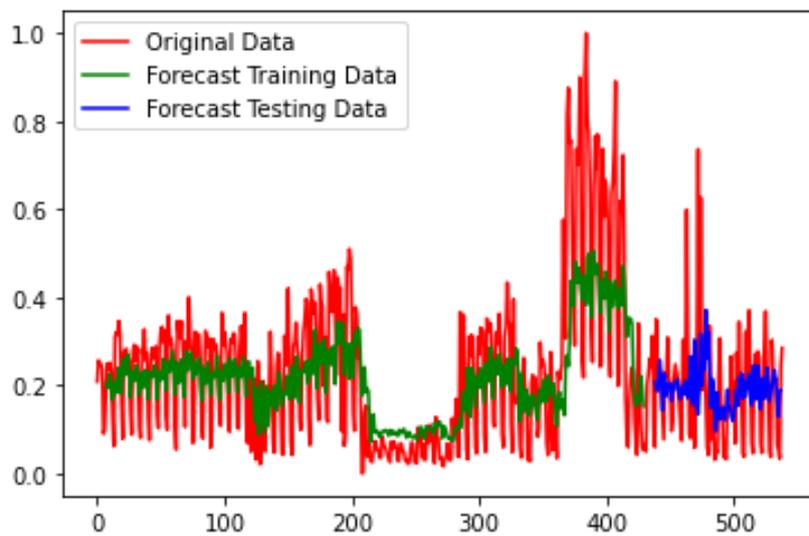


(c)

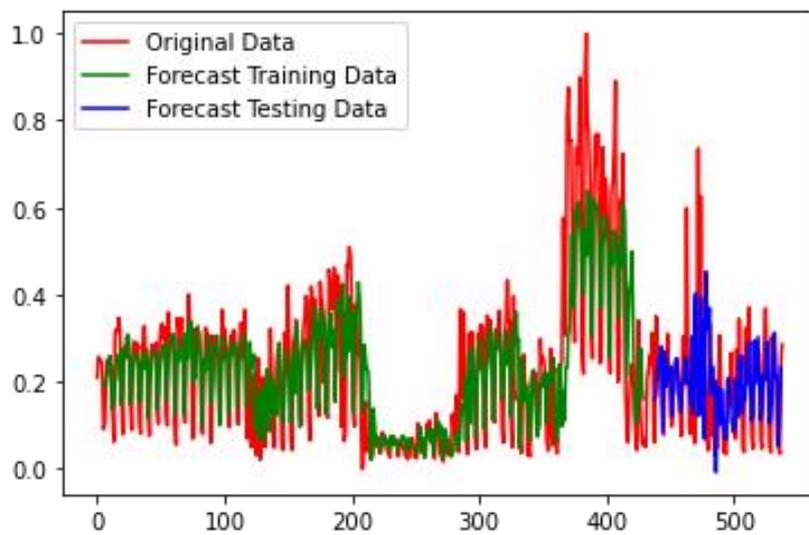
Fig. 7. Prediction result of combined activation function of Sigmoid-ReLU function for each learning rate in BPNN: (a) learning rate 0.1, (b) learning rate 0.5, and (c) learning rate 0.9



(a)



(b)



(c)

Fig. 8. Prediction result of combined activation function of Sigmoid-Tanh function for each learning rate in BPNN: (a) learning rate 0.1, (b) learning rate 0.5, and (c) learning rate 0.9

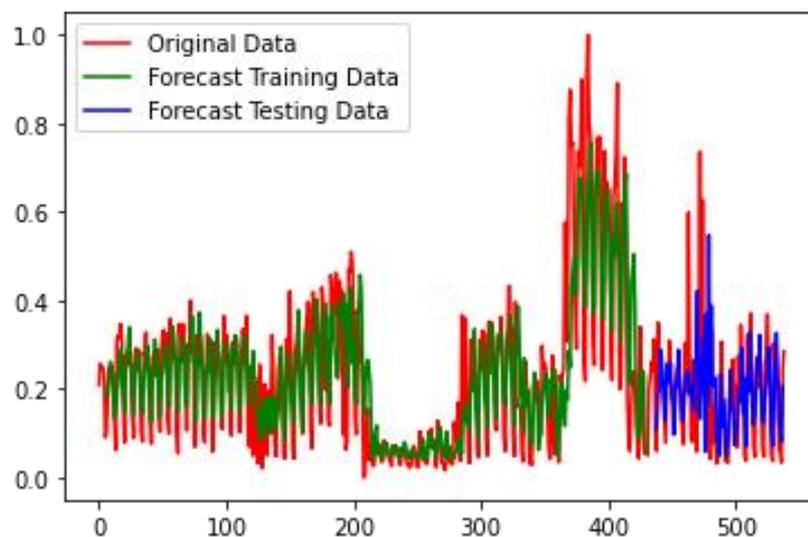
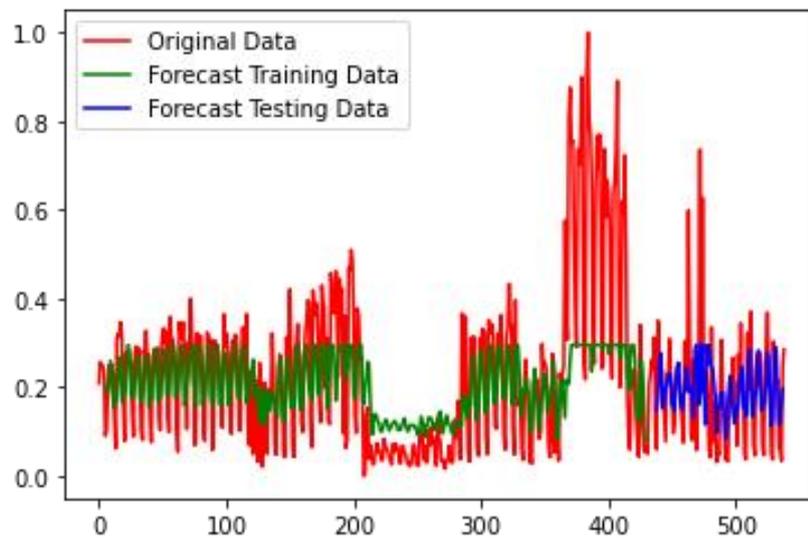
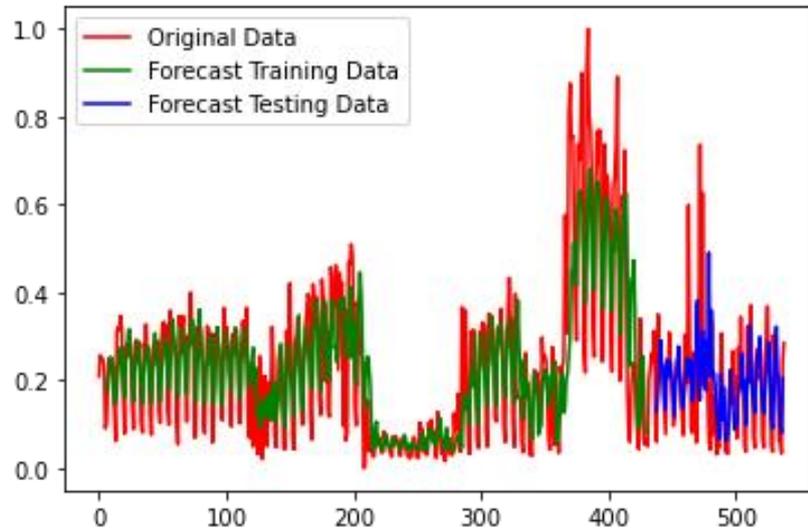
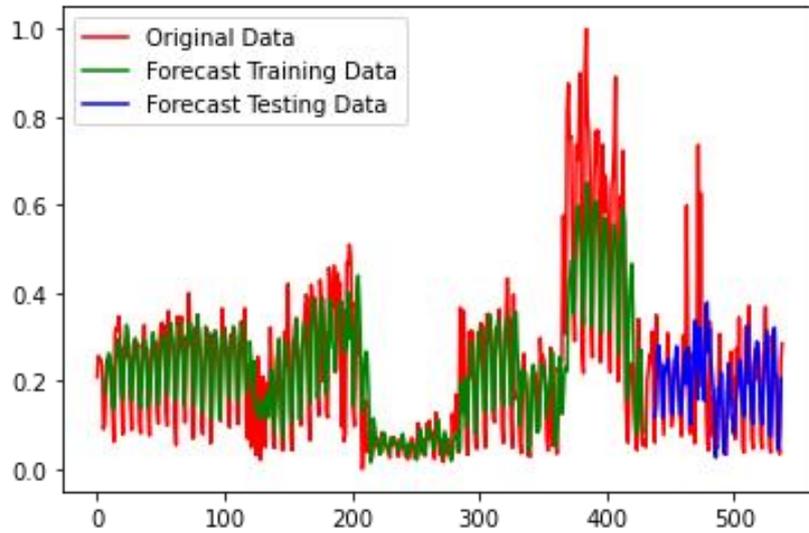
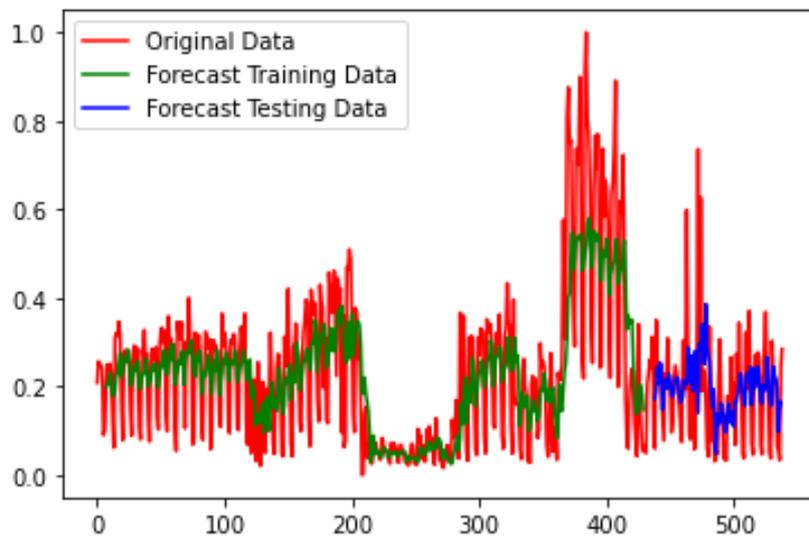


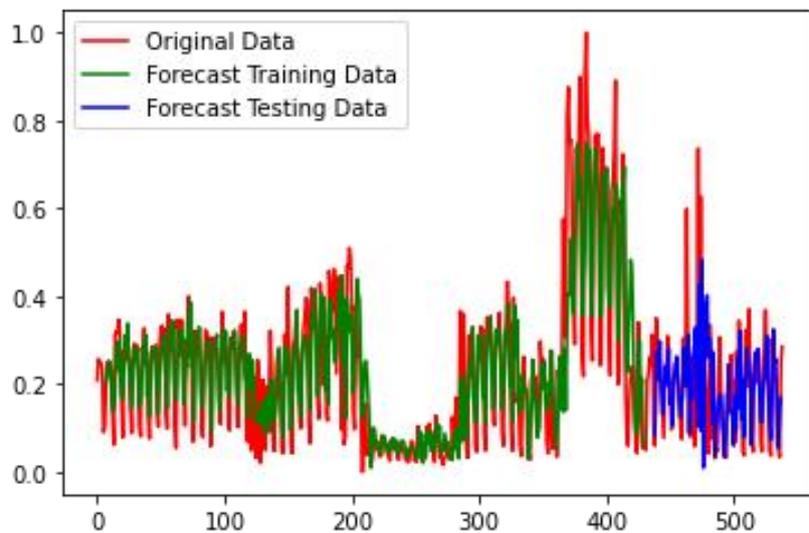
Fig. 9. Prediction result of combined activation function of ReLU-Sigmoid function for each learning rate in BPNN: (a) learning rate 0.1, (b) learning rate 0.5, and (c) learning rate 0.9



(a)



(b)



(c)

Fig. 10. Prediction result of combined activation function of ReLU-Tanh function for each learning rate in BPNN: (a) learning rate 0.1, (b) learning rate 0.5, and (c) learning rate 0.9

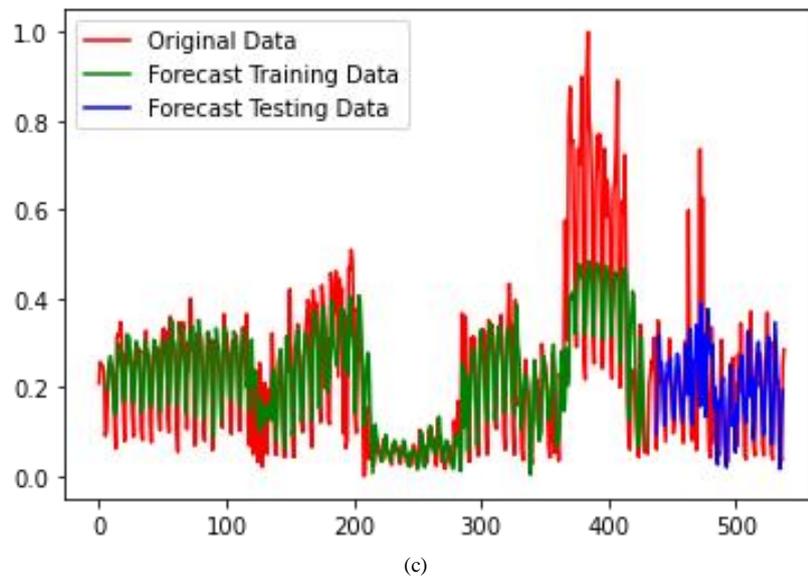
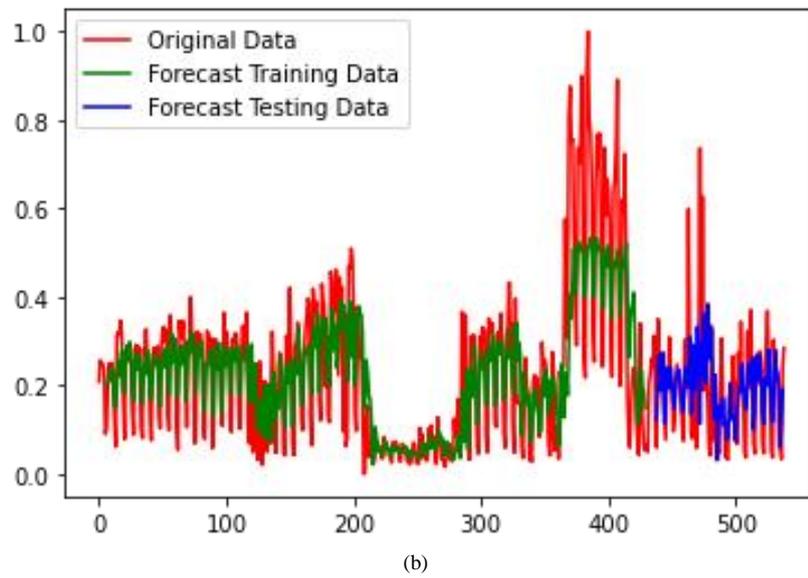
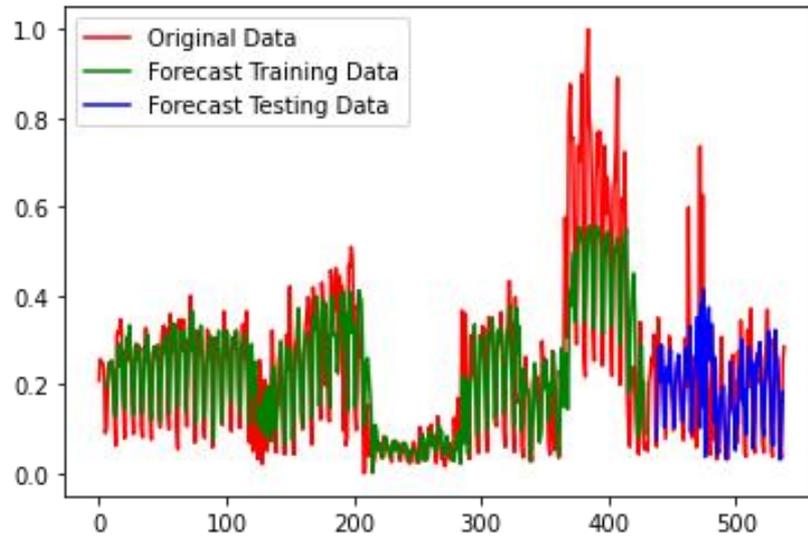
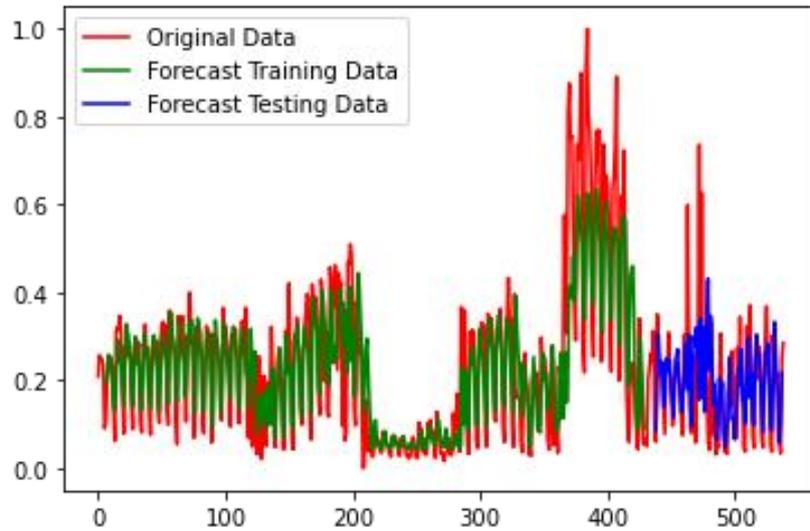
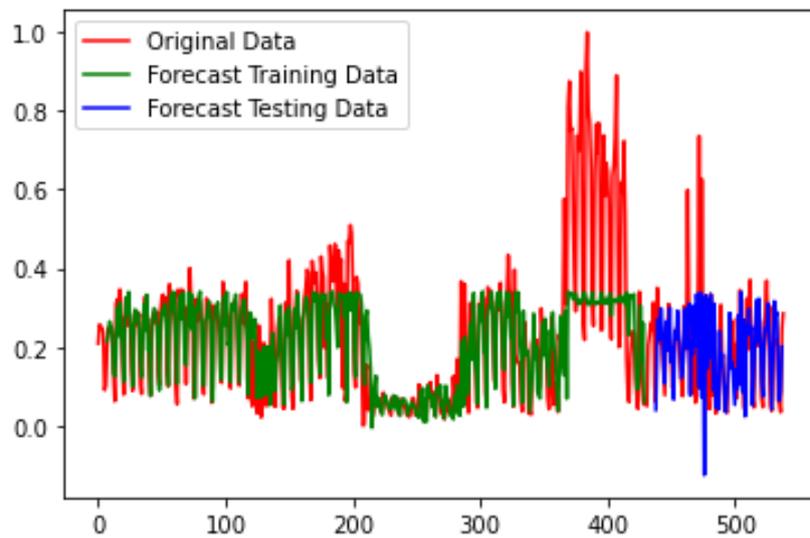


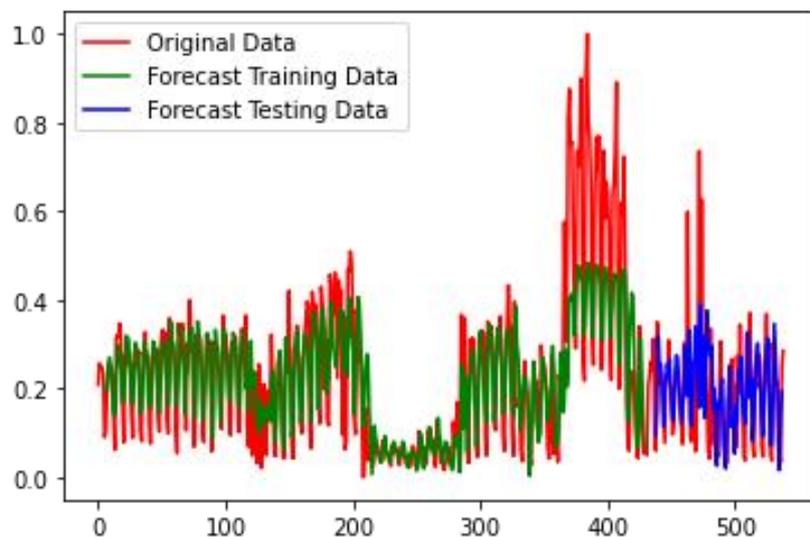
Fig. 11. Prediction result of combined activation function of Tanh-Sigmoid function for each learning rate in BPNN: (a) learning rate 0.1, (b) learning rate 0.5, and (c) learning rate 0.9



(a)



(b)



(c)

Fig. 12. Prediction result of combined activation function of Tanh-ReLU function for each learning rate in BPNN: (a) learning rate 0.1, (b) learning rate 0.5, and (c) learning rate 0.9

Declarations

Author contribution

All authors contributed equally as the main contributor of this paper. All authors read and approved the final paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of interest

The authors declare no known conflict of financial interest or personal relationships that could have appeared to influence the work reported in this paper.

Additional information

Reprints and permission information is available at <http://journal2.um.ac.id/index.php/keds>.

Publisher's Note: Department of Electrical Engineering - Universitas Negeri Malang remains neutral with regard to jurisdictional claims and institutional affiliations.

References

- [1] M. Kihl, P. Ödling, C. Lagerstedt, and A. Aurelius, "Traffic analysis and characterization of internet user behavior," *2010 Int. Congr. Ultra Mod. Telecommun. Control Syst. Work. ICUMT 2010*, no. November, pp. 224–231, 2010.
- [2] V. J. Ribeiro, Z. L. Zhang, S. Moon, and C. Diot, "Small-time scaling behavior of Internet backbone traffic," *Comput. Networks*, vol. 48, no. 3, pp. 315–334, 2005.
- [3] J. K. Taylor and C. Cihon, *Statistical techniques for data analysis*, second edition. 2004.
- [4] P. Purnawansyah, H. Haviluddin, R. Alfred, and A. F. O. Gaffar, "Network Traffic Time Series Performance Analysis Using Statistical Methods," *Knowl. Eng. Data Sci.*, vol. 1, no. 1, p. 1, 2017.
- [5] M. Hanif, F. Sami, M. Hyder, and M. I. Ch, "Hidden Markov Model for Time Series Prediction," *J. Asian Sci. Res.*, vol. 7, no. 5, pp. 196–205, 2017.
- [6] C. You & K. Chandra, "Time series models for Internet data traffic," *Conf. Local Comput. Networks*, 164–171, 1999.
- [7] M. S. Mahdavinejad, M. Rezvan, M. Berekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine learning for internet of things data analysis: a survey," *Digit. Commun. Networks*, vol. 4, no. 3, pp. 161–175, 2018.
- [8] M. Wang, Y. Cui, X. Wang, S. Xiao, and J. Jiang, "Machine learning for networking: Workflow, advances and opportunities," *arXiv*, pp. 1–8, 2017.
- [9] E. S. Yu and C. Y. R. Chen, "Traffic prediction using neural networks," *IEEE Glob. Telecommun. Conf.*, vol. 2, no. May, pp. 991–995, 1993.
- [10] N. Boutaba, R. Salahuddin, M. A. Limam, N. Ayoubi, S. Shahriar and M. Solano, F. E. Aicedo, O, "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities," *J. Internet Serv. Appl.*, vol. 9, no. 5, pp. 1–99, 2018.
- [11] C. N. Babu and B. E. Reddy, "A moving-average filter based hybrid ARIMA-ANN model for forecasting time series data," *Appl. Soft Comput. J.*, vol. 23, pp. 27–38, 2014.
- [12] C. Narendra Babu and B. Eswara Reddy, "Performance comparison of four new ARIMA-ANN prediction models on internet traffic data," *J. Telecommun. Inf. Technol.*, vol. 2015, no. 1, pp. 67–75, 2015.
- [13] J. Rynkiewicz, "Hybrid HMM / MLP models for time series prediction," *Eur. Symp. Artif. Neural Networks*, no. April, pp. 455–462, 1999.
- [14] S. Amin, "Backpropagation – Artificial Neural Network (BP-ANN): Understanding gender characteristics of older driver accidents in West Midlands of United Kingdom," *Saf. Sci.*, vol. 122, no. July 2019, p. 104539, 2020.
- [15] Y. Ling, Q. Yue, C. Chai, Q. Shan, D. Hei, and W. Jia, "Nuclear accident source term estimation using Kernel Principal Component Analysis, Particle Swarm Optimization, and Backpropagation Neural Networks," *Ann. Nucl. Energy*, vol. 136, p. 107031, 2020.
- [16] J. N. Ogunbo, O. A. Alagbe, M. I. Oladapo, and C. Shin, "N-hidden layer artificial neural network architecture computer code: geophysical application example," *Heliyon*, vol. 6, no. 6, p. e04108, 2020.
- [17] M. Lopez-Martin, B. Carro, and A. Sanchez-Esguevillas, "Neural network architecture based on gradient boosting for IoT traffic prediction," *Futur. Gener. Comput. Syst.*, vol. 100, pp. 656–673, 2019.
- [18] S. Wang, W. Zhu, Y. Shen, J. Ren, H. Gu, and X. Wei, "Temperature compensation for MEMS resonant accelerometer based on genetic algorithm optimized backpropagation neural network," *Sens. Actuators, A Phys.*, 316, p.112393, 2020.
- [19] G. Panchal, A. Ganatra, Y. P. Kosta, and D. Panchal, "Behaviour Analysis of Multilayer Perceptrons with Multiple Hidden Neurons and Hidden Layers," *Int. J. Comput. Theory Eng.*, no. June 2017, pp. 332–337, 2011.
- [20] A. J. Thomas, M. Petridis, S. D. Walters, S. M. Gheytsi, and R. E. Morgan, "Two hidden layers are usually better than one," *Commun. Comput. Inf. Sci.*, vol. 744, pp. 279–290, 2017.
- [21] S. Narejo and E. Pasero, "An application of internet traffic prediction with deep neural network," *Smart Innov. Syst. Technol.*, vol. 69, pp. 139–149, 2017.
- [22] C. E. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," *arXiv*, pp. 1–20, 2018.