



# JURNAL AKSI

## Akuntansi dan Sistem Informasi

<http://aksi.pnm.ac.id>

### BPNN's Empirical Analysis of Daily Rupiah Exchange Rate Volatility Utilizing Hidden Neuron Optimization

Imelda Saluza<sup>1)</sup>, Faradillah<sup>2)</sup> and Leriza Desitama Anggraini<sup>3)</sup>

<sup>1)</sup> Universitas Indo Global Mandiri

<sup>2)</sup> Universitas Indo Global Mandiri

<sup>3)</sup> Universitas Indo Global Mandiri

#### ARTICLE INFO

##### *Keywords:*

Volatility, BPNN,  
Neuron

##### *Article History:*

**Received:** 24 February 2022

**Accepted:** 8 May 2022

Corresponding author:

E-mail:

[imeldasaluza@uigm.ac.id](mailto:imeldasaluza@uigm.ac.id)

#### ABSTRACT

The exchange rate is the greatest financial market in its application. As a result, traders, investors, and other money market participants must be aware of the movement of currency exchange rate data. The fluctuation, or rise and fall, of currency exchange rates reveals the level of volatility in a country. The Backpropagation Neural Network is one of the models that can grasp the features of currency exchange rates (BPNN). BPNN is made up of three layers: input, hidden, and output, and each layer contains neurons. One of the challenges in designing a BPNN network architecture is determining the ideal number of hidden layer neurons. In this work, ten methodologies will be utilized to determine the number of hidden neurons; the ten approaches provide distinct empirical results in accordance with the goal of this study, which is to perform an empirical analysis of currency exchange rates by maximizing the number of hidden neurons. Empirical results reveal that the approach for calculating the number of hidden neurons performs well in terms of MAE and MSE. For the following seven periods, the best approach is used to forecast the Rupiah exchange rate.

#### INTRODUCTION

In its most common application, the exchange rate is used when there is commerce, investment, and so on. In this trade transaction, if the exchange rate lowers in a trade, the trade balance will deteriorate, and vice versa. In terms of investing, if investors trade at the correct moment with the proper strategy, they will gain; but, if they do not have the expertise, they will suffer significant losses. To that purpose, dealers, investors, and other financial market players active in the foreign exchange market should have a complete awareness of the variables that might impact currency exchange rate movement. As a result, studying the movement of financial markets becomes a difficult task (Bigus, 2009).

Currency is a component of the national exchange budget that serves as a medium of exchange. Globalization and emancipation have caused many nations' economies to become more open and tighter, with an interconnected economy, making currency exchange rates one of the important variables in international commerce and an open economy. When firms are purchasers of goods or services produced by other nations, exchange rates become significant. The buyer must make payments in the currency of the producer; in other words, the buyer must purchase the currency in order

**p-ISSN: 2528-6145, e-ISSN: 2541-3198 Accredited Third Grade by Ministry of Research, Technology of The Republic of Indonesia, Decree No: 148/M/KPT/2020**

**Cite this as:** Saluza, I., Faradillah and Anggraini, L.D. (2022). BPNN's Empirical Analysis of Daily Rupiah Exchange Rate Volatility Utilizing Hidden Neuron Optimization. JURNAL AKSI (Akuntansi dan Sistem Informasi), 7 (1), 41-51. <http://doi.org/10.32486/aksi.v7i1.249>

to conduct business. Foreign currency traders will gain from purchasing and selling these currencies at various and variable exchange rates. In actuality, the exchange rate is impacted by a variety of economic, political, and even psychological variables, all of which interact in a complicated manner. Exchange rate data sets exhibit considerable volatility, complexity, and noise because of market mechanics that are difficult to learn and comprehend from daily observations (Wei Huang, K.K. Lai, Y. Nakamori, 1995).

The rise and fall of currency exchange rates in the financial market demonstrates the level of volatility that exists between a country's currency and the currencies of other countries. The greater the volatility, the more volatile the exchange rate. As a result, currency exchange rates versus other nations' currencies may be overvalued or undervalued. And if the currency exchange rate sees severe volatility, the economy will face both micro and macroeconomic instability (Mukhlis, 2011). Currency exchange rate volatility will represent a danger of gain and loss in a country, including Indonesia.

The fluctuation of the Rupiah exchange rate against other currencies following the establishment of Indonesia's free-floating exchange rate system policy on August 14, 1998 has an influence on the growth of the national economy in a variety of sectors. The Indonesian economy remains extremely exposed to foreign economic fluctuations, which has led in economic openness, as evidenced in the growth of an outward-oriented national industry, which has a direct influence on the increasingly flexible exchange rate. As a result, one useful method is to forecast the direction of changes in currency exchange rates.

For investors, accurate forecasting of exchange rate volatility is critical. People who do business, borrowers and lenders, and policymakers must develop ways to safeguard assets from the risk of loss by considering future possibilities (Wei et al., 2019). As a result, managing currency rates has become highly critical, and anticipating exchange rates has become a very important demand.

It is difficult to identify or detect nonlinear trends in time series when forecasting exchange rates using statistical approaches (Shin et al., 2020). Meanwhile, the exchange rate is the most volatile, nonlinear, noisy, and unstable market, making it challenging for linear models to capture these features. The Neural Network (NN) is a frequently used model that can grasp the properties of the exchange rate. NN is capable of analyzing previous data and forecasting future exchange rate changes. NN is essentially a human brain modeling implementation that can find functional links based on a collection of data and perform tasks like pattern recognition, classification, assessment, predicting modeling, and control. Among the benefits of NN is that it is well suited to discovering accurate solutions in environments characterized by complexity, noise, and irrelevant information (Wei Huang, K.K. Lai, Y. Nakamori, 1995).

The Neural Network (NN) is a mathematical model inspired by the function of organic neurons. The number of NNs connected with the tasks assigned to the network varies. Many writers have utilized and developed NN for predicting purposes. (Asadullah et al., 2020) created a successful model for forecasting irregular patterns in exchange rates using NN, and the findings demonstrate that NN outperforms a linear model in forecasting accuracy; (Pandey et al., 2020) shown in his research that NN outperforms several linear models; (Pang et al., 2020) established a unique NN strategy to forecasting utilizing NN with Long-Short Term Memory (LSTM), and the results demonstrate improved forecasting performance in forecasting the Shanghai composite index; (Panda & Narasimhan, 2007) estimating the exchange rate of the Rupee versus the Dollar using NN as a forecasting approach that splits the sample into in-sample and out-of-sample; (SANUSI et al., 2020) forecasting Malaysia's GDP, the findings demonstrate that the application of NN can solve the problem of multicollinearity between variables as well as the non-linear problem in the increase of Malaysia's GDP value; while study (Amran & Ariffin, 2020) investigated several designs for estimating the Malaysian currency exchange rate; (Sahu et al., 2020) using Virtual Data Points (VDP) enhances the accuracy of NN predicting.

Because of the inclusion of many properties for carrying out forecasting tasks, NN has been widely employed as a viable alternative technique to predicting. NN differs from other models in various ways, which is why it is frequently employed in predicting. (1) NN is diametrically opposed to the old approach since it is a data-driven self-adaptive technique (Hill et al., 1996), (2) NN is a generalizable technique (Huang et al., 2007), (3) NN is a universal functional approach (Memon et al., 2020), and (4) NN is a nonlinear technology (Hu et al., 1999). These are the benefits of NN; nevertheless, in addition to these benefits in its use, NN has drawbacks. The disadvantages of NN include the need for many neural thresholds, a slightly slower convergence rate, and many computations. Backpropagation Neural Networks are one sort of NN that is commonly employed (BPNN). BPNN is a type of multilayer NN. Multilayer NN networks are typically trained using a gradient descent algorithm (Faris et al., 2019).

In its application, BPNN has various benefits, including extremely excellent adaptive qualities and the capacity to train independently, generalization ability, strong nonlinear mapping ability, and very high learning accuracy (K. Gnana Sheela & Deepa, 2013). It has even been theoretically demonstrated

that NN with three layers may approximate a continuous nonlinear function with arbitrary precision. This is what makes BPNN so well suited to solving complicated issues. However, as an optimization approach, BPNN has limitations in terms of convergence speed, is very sensitive to weight changes, and is frequently caught in local minimums.

NN has two primary characteristics that might impact network performance. The first feature is the network structure, and the second is the (Faris et al., 2019) learning algorithm. One of the most difficult difficulties with network construction when using NN is deciding which hidden neurons to employ [8 in automated]. This choice is critical if the chosen NN architecture is trained to produce low errors but is unable to generalize its prediction performance. This indicates that the design of the NN training process has an overtraining problem. Overtraining is an issue comparable to overfitting data. This issue emerges when the network fits so well with the training data that its generalizability for validation and testing is compromised.

Previous scholars have achieved several advances in forecasting using NN with the goal of improving predicting accuracy. However, in its application, NN includes numerous aspects that may be utilized to design a network architecture, such as learning methods, activation functions, the number of neurons, and the parameters employed. There are various methods to establish a network on a NN. There has been no agreement on these factors up to this point. As a result, different judgments have varying degrees of efficacy (Wei Huang, K.K. Lai, Y. Nakamori, 1995). Such as selecting the number of hidden layers, the more hidden layers there are, the longer the computation time and the possibility of overfitting (Baum & Haussler, 1989); output neurons on the NN, the more neurons there are, the lower the predicting accuracy value (Markova, 2019). As a result, when developing NN, it is vital to plan for more accurate forecasts while avoiding overfitting.

The problem of currency exchange rate data features and the best selection of hidden neurons in the creation of the architecture when generating predictions formed the foundation for performing empirical research in this work. The goal of this work is to give an empirical analysis of historical Rupiah exchange rate data by choosing the number of hidden neurons.

## MATERIALS AND METHODS

### BPNN Algorithm

To estimate financial data, the BPNN algorithm was employed. A feed forward network made of multi-layered network neurons is a NN network. The output neurons from each layer are transferred to the subsequent levels, where the network weights determine the transmission. The backpropagation algorithm's training process is divided into two steps: the first is forward propagation, which provides information; the second is error propagation, which aims to correct the weights by spreading the error between the actual output and the desired output from the output layer back to the input layer. Overall, BPNN performance connects each layer and is then managed using a supervised learning approach. The following equation is used to model NN:

$$\hat{Y}(t + 1) = F_2[V^T(t)F_1(W(t)X(t))] \dots \dots \dots (1)$$

where:

$$X = (X_0, X_1, \dots, X_p)^T \quad : \quad \text{input BPNN};$$

$$\hat{Y} = (\hat{y}_0, \hat{y}_1, \dots, \hat{y}_k)^T \quad : \quad \text{output BPNN};$$

$X$  and  $\hat{Y}$  is a valid number

$$W = \begin{bmatrix} w_{10} & w_{11} & \dots & w_{1p} \\ w_{20} & w_{21} & \dots & w_{2p} \\ \vdots & \vdots & & \vdots \\ w_{q0} & w_{q1} & \dots & w_{qp} \end{bmatrix} = (w_0, w_1, \dots, w_p);$$

$$V = \begin{bmatrix} v_{10} & v_{20} & \dots & v_{k0} \\ v_{11} & v_{21} & \dots & v_{k1} \\ \vdots & \vdots & & \vdots \\ v_{1q} & v_{2q} & \dots & v_{kq} \end{bmatrix} = (v_0, v_1, \dots, v_k)$$

$$F_1(W(t)X(t)) = \left( F_1(\text{net}_0(t)) \ F_1(\text{net}_1(t)) \ \dots \ F_1(\text{net}_q(t)) \right)^T;$$

$$net_i(t) = \sum_{j=0}^p w_{ij}(t)x_j(t), i = 1,2, \dots, q$$

$$w_{i0}(t), i = 1,2, \dots, q$$

$$v_{ij}(t), i = 1,2, \dots, q; j = 1,2, \dots, k$$

$$v_{i0}(t)$$

$$F_1(\cdot) \text{ dan } F_2(\cdot)$$

output from hidden neuron n-i;

hidden neuron bias;

hidden neuron weights i to output neuron j ;

bias in output neurons;;

Nonlinear, continuous, finite, and derivable activation function.

**BPNN Structure Proposal**

In its application, BPNN has three layers: the input layer, the hidden layer, and the output layer. The input layer connects directly with the external world, which is then sent to the hidden layer for processing before being shown in the output layer.

The input layer is made up of two categories of input: fundamental and technical (Bigus, 2009). Fundamental input contains independent factors that impact a dependent variable, while technical input includes time series data, moving averages, and so on. Meanwhile, Walczak noted that the input was made up of univariate and multivariate data. Univariate input uses data straight from the time series data that is being forecasted; in other words, univariate input is more reliant on the data's predictive abilities. The network input for univariate time series prediction is past data, while the network output is future data. Meanwhile, multivariate input incorporates information from sources other than the time series data. The inputs were chosen for a variety of reasons, including the desire to fully use the information available between the sample observations, even if the link that happened was unclear or difficult to explain. As a result, using the autocorrelation criteria is one of the criteria that may be utilized to overcome issues with univariate data. After finding the input layer, the hidden layer must be determined.

The hidden layer is an intermediate layer between the input and output layers that is made up of neurons with activation functions. Countless research have been conducted to establish the number of neurons in this layer. The issue that emerges is determining the best number of neurons to use in constructing the network design. Overtraining can occur if the number of hidden neurons is chosen incorrectly. The "trial and error" method has been commonly used to calculate the number of hidden neurons, however it is not optimum. Many researchers have discovered a formula for calculating the number of hidden neurons.

The following layer is the output layer; the output layer of the NN is used to show the pattern to the external world. The number of neurons in the input layer is chosen to suit the goal of the NN. Figure 1 depicts the network structure of the BPNN architecture used in this study:

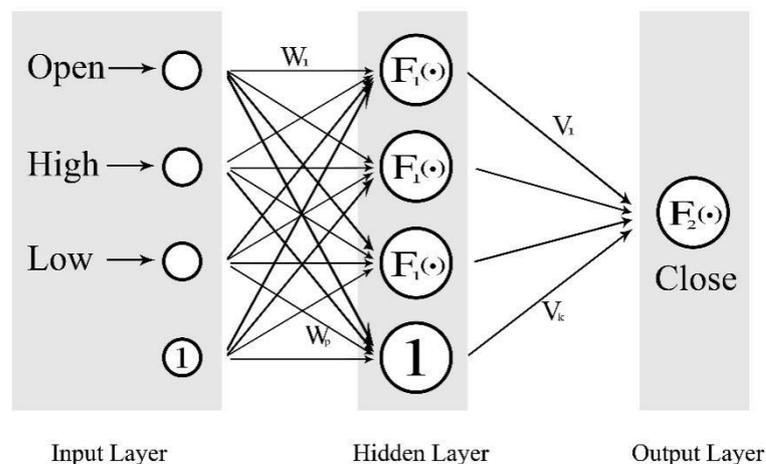


Figure 1. BPNN Architecture

The closing price, highest price, and lowest price are utilized as inputs for the prediction model for the volatility of the Rupiah exchange rate, implying that the input layer design consists of three neurons. According to Figure 1, the pairing between the input layer and the target output layer is as follows:

$$(X_1, X_2, X_3: Y) = (\text{open, high, low: closed})$$

For example,  $W_p$  represents the weight produced between the input layer and the hidden layer, and  $W_o$  represents the weight formed between the hidden layer and the output layer. And the activation function connects each neuron one to one  $F_1(.)$  dan  $F_2(.)$ .

Figure 1 depicts how the layer performs computations independently of the supplied data and then transfers the results to subsequent levels until the network output is determined. The input is sent to the hidden layer through an activation function  $F_1(.)$  with weights  $W_p$ , and the network learns the function using the input and output of the prior learning process. In addition, the output value is sent together with the activation function and weights. The suggested model's results are utilized to forecast the volatility of the Rupiah exchange rate. The suggested network in Figure 1 is represented by the equation below.

Input vector	$[X_1, X_2, X_3]$
Output vector	$[Y]$
Weight vector of input to hidden vector	$\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1p} \\ w_{21} & w_{22} & \cdots & w_{2p} \\ w_{31} & w_{32} & \cdots & w_{3p} \end{bmatrix}$
Weight vector of hidden to hidden output target	$\begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1p} \\ v_{21} & v_{22} & \cdots & v_{2p} \\ v_{31} & v_{32} & \cdots & v_{3p} \end{bmatrix}$

### Neuron Selection in the Hidden Layer

The selection of the ideal number of neurons in the hidden layer becomes one of the main difficulties in establishing the topology of the NN network, because it might induce overtraining during the design phase. Furthermore, tradeoff issues will arise. The output of neurons gets unstable as the number of neurons increases, and vice versa. One of the most difficult tasks to address is estimating the volatility of the Rupiah exchange rate (K. Gnana Sheela & Deepa, 2013).

As most prediction research is heuristic at the moment, numerous approaches for calculating the amount of neurons in the hidden layer have evolved throughout time. Table 1 demonstrates multiple methods for calculating the number of neurons in the buried layer. The symbols used in table 1 have been standardized as follows:

$N_i$	=	number of input neurons
$N_h$	=	number of hidden neurons
$N_o$	=	number of output neurons
$n$	=	Total number of neurons in the network

Table 1. Methods of determining number of hidden neurons

$N_h$	Description	Published
$\frac{4N_i - 2}{N_i - 3}$	To calculate the number of buried neurons, the researcher did a comparative study of numerous research outcomes using his proposed technique. When compared to existing ways, the suggested methodology delivers greater performance with a higher degree of accuracy, lower error, enhanced stability, and faster convergence (Madhiarasan & Deepa, 2017).	2016
$\frac{3N_i}{(N_i + 6)}$	Propose a method for calculating the number of hidden neurons in forecasting wind speed. The evaluation findings reveal that the performance is satisfactory (Meng et al., 2016).	2016
$\frac{8N_i - 4}{N_i - 4}$	Provide a recommended method for calculating the number of hidden neurons. In practice, wind data is used to evaluate the approach. The assessment findings suggest that the proposed technique performs well (Madhiarasan & Deepa, 2016).	2015

$\frac{\sqrt{N_i + N_o}}{2}$	This paper suggests a method for calculating the number of hidden neurons in forecasting wind speed. The suggested method enhances forecasting accuracy, allowing for more trustworthy and precise projections (Wang & Hu, 2015).	2015
$\left(\sqrt[3]{\frac{N_i}{N_o}}\right)^2$	Make realistic suggestions for examining data samples in order to determine the amount of hidden neurons. When employing BPNN, the model is good since it saves training time and network testing (Grabusts & Zorins, 2015).	2015
$\frac{4N_i}{N_i - 2}$	Researchers have presented a novel method for determining the number of hidden neurons in a NN network. The suggested method can decrease mistakes and improve network stability and accuracy. Experiments reveal that the novel approach performs well in estimating hidden neurons (K. G. Sheela & Deepa, 2014).	2014
$\frac{6N_i}{(N_i + 1)}$	The proposed method performs well when measured using statistical equations (Madhiarasan & Deepa, 2017).	2013
$\frac{(4N_i^2 + 3)}{N_i^2 - 8}$	The researchers put 101 proposed criteria to the test in order to assess the amount of buried neurons. According to the results, the suggested model enhances accuracy and minimizes mistakes (K. Gnana Sheela & Deepa, 2013).	2013
$\sqrt{N_i + N_o + \alpha}$	Researchers use NN to forecast rural per capita life consumption by constructing a model to identify the number of hidden neurons. The findings demonstrate that the model is more accurate. To calculate the number of hidden neurons, the model equation takes into account the number of neurons in the input layer, the number of neurons in the output layer, and a value that is an integer between 0 and 10 (Qian & Yong, 2013).	2013
$2^n - 1$	In the lack of a "trial and error" approach for calculating the number of hidden neurons, the study's findings present a benefit. The trial findings revealed an accuracy rate of 85% (Hunter et al., 2012).	2012

**Accuracy assessment of models**

The number of buried neurons was calculated using error criteria such as Mean Absolute Error (MAE) and Mean Square Error (MSE) (MSE). The following equation is the formula for these conditions:

$$MAE = \frac{1}{N} \sum_{i=1}^N (Y'_i - Y_i),$$

$$MSE = \sum_{i=1}^N \frac{(Y'_i - Y_i)^2}{N} \dots\dots\dots (2)$$

where  $Y_i$  is the predicted output,  $Y'_i$  is the actual output, and  $N$  is the sample count. The network design process is crucial to network performance.

**RESULTS AND DISCUSSION**

**Data Collection**

The data utilized in this study are daily real data of the Rupiah exchange rate versus the US Dollar from January 2015 to February 2022, collected from the website [id.investing.com](http://id.investing.com). The network's inputs are the opening price, the maximum price, and the lowest price. The closing price is utilized as the output. This study offered a total of 1,817 samples. The data set utilized to conduct an empirical study of the Rupiah exchange rate is depicted in Figure 1.

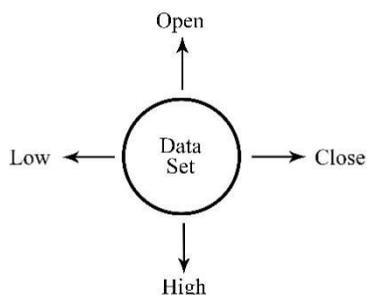


Figure 2. Utilized Data Set

In addition, Figure 2 depicts the existing data collection. Figure 2 demonstrates that the Rupiah exchange rate data is exceedingly unpredictable, making it impossible to anticipate the data using statistical models. This is due to the data's volatility.

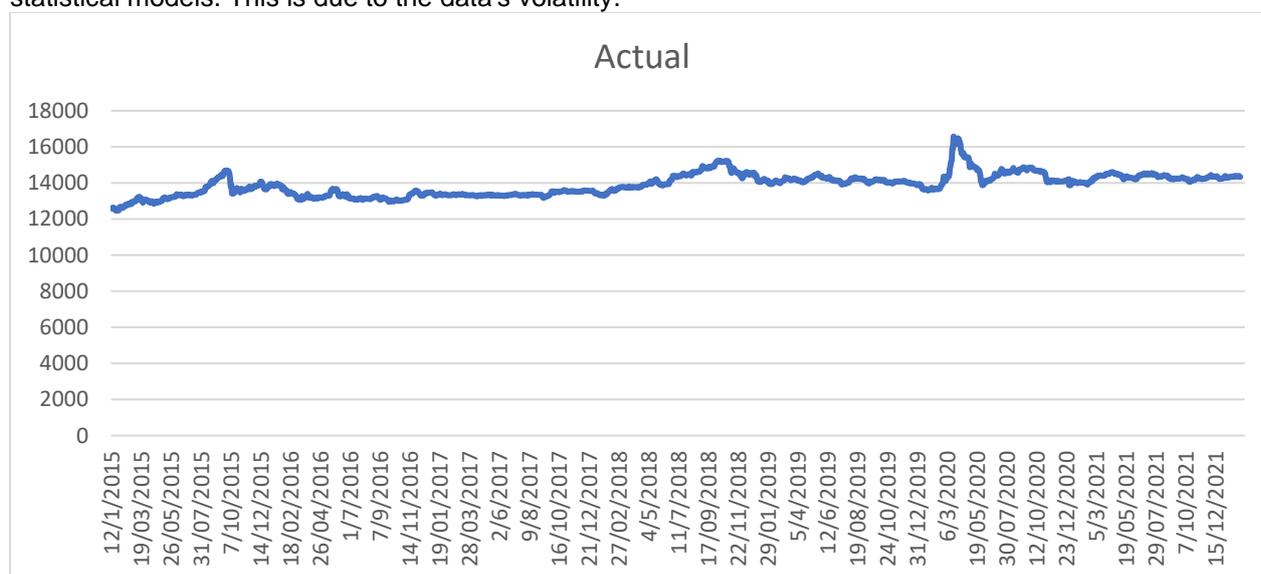


Figure 3. Actual Rupiah Exchange Rate

### Data Normalization

The act of normalizing or scaling the data is a critical step once the data set has been acquired. This stage is carried out with the goal of dealing with actual data with varying ranges. As a result, normalization was employed to scale the actual time in the 0-1 range.

Table 2. Proposed BPNN Designed Parameters

BPNN parameters	
Input neurons	3 [open, high, low]
Number of hidden layer	1
Output neurons	1 [closed]
Number of epochs	10000
Learning rate	0.1

Table 2 shows the design of the BPNN parameters utilized in this work, which includes input neurons, the number of hidden layers, and so on. The network's input layer is made up of three neurons, and the output layer is made up of one neuron. As for the number of neurons from the hidden layer, estimate the minimal mistake using the approach in table 1. The BPNN method begins with the input being transferred to the hidden layer by multiplying the weights with the activation hyperbolic function, and the output from the hidden layer being transferred to the target output layer by multiplying the weights with the activation identity function. The training is done on normalized data, with the stop condition being the accomplishment of the optimal error from the data set utilized and quantifying the

error using equations (2). The network will be tested when it has been trained. The trained and tested NNs are used for each technique of identifying hidden neurons discussed.

## RESULTS

Several methodologies have been employed to determine the number of concealed NN neurons, according to the findings of earlier investigations. Pruning and constructive methods are the two types of techniques. The network pruning technique begins with a big number of hidden neurons and then trims the neurons until the minimum weight is attained, whereas the constructive strategy begins with a small number of neurons and then adds neurons until the minimum weight is obtained. Both of these widely used systems have some disadvantages. As indicated in table 1, some of the most recent study findings give a mechanism for calculating the number of buried neurons.

This study employed eleven methods discovered by earlier researchers to determine the amount of hidden neurons. In order to assess network performance, the 10 techniques were compared using MAE and MSE. The most optimum network design was employed to anticipate the Rupiah exchange rate. Table 3 summarizes the outcomes of the 10 techniques employed.

Table 3. The results of BPNN designed parameters

No.	Different Methods	Published	Number of hidden neurons	Training Set		Test Set	
				MAE	MSE	MAE	MSE
1	(Madhiarasan & Deepa, 2017)	2016	-				
2	(Meng et al., 2016)	2016	3	0.0024	1.1649e-05	0.0022	1.6762e-05
3	(Madhiarasan & Deepa, 2016)	2015	-				
4	(Wang & Hu, 2015)	2015	1	0.0047	5.1932e-05	0.0064	9.7178e-05
5	(Grabusts & Zorins, 2015)	2015	2	0.0082	1.1735e-04	0.0034	2.0320e-05
6	(K. G. Sheela & Deepa, 2014)	2014	12	0.0018	4.0867e-06	0.0034	2.5773e-05
7	(Madhiarasan & Deepa, 2017)	2013	5	0.0011	2.2260e-06	0.0019	6.3073e-06
8	(K. Gnana Sheela & Deepa, 2013)	2013	39	0.0026	1.2181e-05	0.0028	1.1701e-05
9	(Qian & Yong, 2013)	2013	4	0.0044	3.8241e-05	0.0016	1.7624e-05
10	(Hunter et al., 2012)	2012	7	0.0016	6.4849e-06	0.0015	1.5880e-05

According to table 3, there are two approaches for calculating the number of hidden neurons that cannot be employed out of the 10 methods. This is due to the fact that the denominator of the two equations yields a negative and zero value, resulting in an indeterminate number of neurons, or the number of neurons cannot be counted. This is due to the fact that the proposed equation for determining the number of hidden neurons is only updated for the amount of inputs utilized by the researcher.

Table 3 also shows how many hidden neurons were employed in the procedure. The least number of neurons is 1 and the largest number of neurons is 39, according to the 10 techniques employed. Furthermore, the 10 techniques were trained using training and testing data. The performance of each network is then measured using MAE and MSE, and the approach that delivers the highest performance is chosen as the optimal design, which is subsequently utilized for forecasting. According to table 3, all of the approaches utilized gave the best predicting results. However, some approaches perform well just on training data and not so well on testing data. There is one approach among the 10 that produces the best results for both training and testing data. This approach generates a total of 5 hidden neurons using the  $\frac{6N_i}{(N_i+1)}$  equation. Furthermore, it is possible to deduce that the number of extremely tiny or very big neurons does not produce ideal outcomes. Furthermore, as illustrated in Figure 4, the selected architecture is employed to forecast Rupiah exchange rate data.

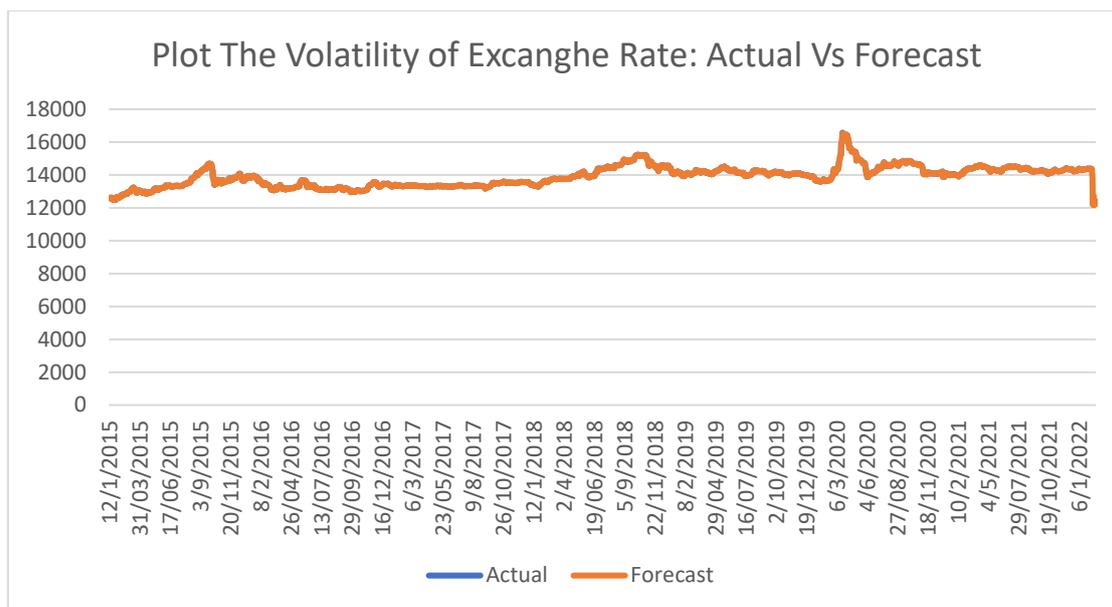


Figure 4. Plot the Volatility of Exchange Rate: Actual Vs Forecast

Figure 4 indicates that the prediction data match the actual data, implying that NN can be used to examine the volatility of highly changing exchange rate data. The researcher then anticipates the Rupiah exchange rate for the following seven periods, as indicated in table 4.

Table 4. Predictions for the Rupiah Exchange Rate

Day	Actual	Forecast
7/2/2022	14395	14395.99
8/2/2022	14390	14390.96
9/2/2022	14355	14355.76
10/2/2022	14343	14343.68
10/3/2022		13781.45
10/4/2022		12326.57
10/5/2022		12164.72
10/6/2022		12343.81
10/7/2022		12869.10
10/8/2022		14114.85
10/9/2022		17345.29

## CONCLUSIONS AND SUGGESTION

This research conducts an empirical study of 10 approaches for calculating the number of hidden neurons in BPNN. The 10 techniques were compared in order to forecast the volatility of Rupiah exchange rate data, and the performance of each design was evaluated using MAE and MSE. Following that, the ideal network performance will be chosen, and forecasting for the following seven periods will be performed. Based on the research findings, it is possible to deduce that the method's use must be tailored to the parameters available. Because not all approaches for determining the number of hidden neurons can be applied. The number of tiny or big hidden neurons calculated based on the equations did not offer ideal results for both training and testing data. As a result, training and testing all techniques is the best strategy to decide the final topology of BPNN. With the equation  $\frac{6N_i}{(N_i+1)}$ , the optimal number of hidden neurons in this study is five. These findings demonstrate good performance for both training and testing data.

## REFERENCES

- Amran, I. M., & Ariffin, A. F. (2020). Forecasting Malaysian Exchange Rate using Artificial Neural Network. *Jurnal Intelek*, 15(2), 136–145. <https://doi.org/10.24191/ji.v15i2.323>
- Asadullah, M., Ahmad, N., & Dos-Santos, M. J. P. L. (2020). Forecast Foreign Exchange Rate: The Case Study of PKR/USD. *Mediterranean Journal of Social Sciences*, 11(4), 129. <https://doi.org/10.36941/mjss-2020-0048>
- Baum, E. B., & Haussler, D. (1989). What Size Net Gives Valid Generalization? *Neural Computation*, 1(1), 151–160. <https://doi.org/10.1162/neco.1989.1.1.151>
- Bigus, J. P. (2009). Data Mining with Neural Networks. *New York*, 5(1), 1–154. <http://dl.acm.org/citation.cfm?id=231007>
- Faris, H., Mirjalili, S., & Aljarah, I. (2019). Automatic selection of hidden neurons and weights in neural networks using grey wolf optimizer based on a hybrid encoding scheme. *International Journal of Machine Learning and Cybernetics*, 10(10), 2901–2920. <https://doi.org/10.1007/s13042-018-00913-2>
- Grabusts, P., & Zorins, A. (2015). The influence of hidden neurons factor on neural network training quality assurance. *Vide. Tehnologija. Resursi - Environment, Technology, Resources*, 3, 76–81. <https://doi.org/10.17770/etr2015vol3.213>
- Hill, T., O'Connor, M., & Remus, W. (1996). Neural network models for time series forecasts. *Management Science*, 42(7), 1082–1092. <https://doi.org/10.1287/mnsc.42.7.1082>
- Hu, M. Y., Zhang, G. P., Jiang, C. X., & Patuwo, B. E. (1999). A cross-validation analysis of neural network out-of-sample performance in exchange rate forecasting. *Decision Sciences*, 30(1), 197–216. <https://doi.org/10.1111/j.1540-5915.1999.tb01606.x>
- Huang, W., Lai, K. K., Nakamori, Y., Wang, S., & Yu, L. (2007). Neural networks in finance and economics forecasting. *International Journal of Information Technology and Decision Making*, 6(1), 113–140. <https://doi.org/10.1142/S021962200700237X>
- Hunter, D., Yu, H., Pukish, M. S., Kolbusz, J., & Wilamowski, B. M. (2012). Selection of proper neural network sizes and architectures-A comparative study. *IEEE Transactions on Industrial Informatics*, 8(2), 228–240. <https://doi.org/10.1109/TII.2012.2187914>
- Madhiarasan, M., & Deepa, S. N. (2016). A novel criterion to select hidden neuron numbers in improved back propagation networks for wind speed forecasting. *Applied Intelligence*, 44(4), 878–893. <https://doi.org/10.1007/s10489-015-0737-z>
- Madhiarasan, M., & Deepa, S. N. (2017). Comparative analysis on hidden neurons estimation in multi layer perceptron neural networks for wind speed forecasting. *Artificial Intelligence Review*, 48(4), 449–471. <https://doi.org/10.1007/s10462-016-9506-6>
- Markova, M. (2019). Foreign exchange rate forecasting by artificial neural networks. *AIP Conference Proceedings*, 2164(October). <https://doi.org/10.1063/1.5130812>
- Memon, I. Z., Talpur, S., Narejo, S., Junejo, A. Z., & Hassan, F. (2020). Short-term prediction model for multi-currency exchange using artificial neural network. *Proceedings - 3rd International Conference on Information and Computer Technologies, ICICT 2020, June*, 102–106. <https://doi.org/10.1109/ICICT50521.2020.00024>
- Meng, A., Ge, J., Yin, H., & Chen, S. (2016). Wind speed forecasting based on wavelet packet decomposition and artificial neural networks trained by crisscross optimization algorithm. *Energy Conversion and Management*, 114, 75–88. <https://doi.org/10.1016/j.enconman.2016.02.013>
- Mukhlis, I. (2011). Analisis Volatilitas Nilai Tukar Mata Uang Rupiah Terhadap Dolar. *Journal of Indonesian Applied Economics*, 005(02), 172–182. <https://doi.org/10.21776/ub/jiae/2016/005.02.8>
- Panda, C., & Narasimhan, V. (2007). Forecasting exchange rate better with artificial neural network. *Journal of Policy Modeling*, 29(2), 227–236. <https://doi.org/10.1016/j.jpolmod.2006.01.005>

- Pandey, T. N., Jagadev, A. K., Dehuri, S., & Cho, S. B. (2020). A novel committee machine and reviews of neural network and statistical models for currency exchange rate prediction: An experimental analysis. *Journal of King Saud University - Computer and Information Sciences*, 32(9), 987–999. <https://doi.org/10.1016/j.jksuci.2018.02.010>
- Pang, X., Zhou, Y., Wang, P., Lin, W., & Chang, V. (2020). An innovative neural network approach for stock market prediction. *Journal of Supercomputing*, 76(3), 2098–2118. <https://doi.org/10.1007/s11227-017-2228-y>
- Qian, G., & Yong, H. (2013). *Forecasting the Rural Per Capita Living Consumption Based on Matlab BP Neural Shanghai University of Engineering Science*. 4(17), 131–137.
- Sahu, K. K., Nayak, S. C., & Behera, H. S. (2020). Forecasting Currency Exchange Rate Time Series With Fireworks Algorithm-Based Higher Order Neural Network, With Special Attention To Training Data Enrichment. *Computer Science*, 21(4), 463–489. <https://doi.org/10.7494/csci.2020.21.4.3474>
- SANUSI, N. A., MOOSIN, A. F., & KUSAIRI, S. (2020). Neural Network Analysis in Forecasting the Malaysian GDP. *Journal of Asian Finance, Economics and Business*, 7(12), 109–114. <https://doi.org/10.13106/JAFEB.2020.VOL7.NO12.109>
- Sheela, K. G., & Deepa, S. N. (2014). Selection of number of hidden neurons in neural networks in renewable energy systems. *Journal of Scientific and Industrial Research*, 73(10), 686–688.
- Sheela, K. Gnana, & Deepa, S. N. (2013). Review on methods to fix number of hidden neurons in neural networks. *Mathematical Problems in Engineering*, 2013. <https://doi.org/10.1155/2013/425740>
- Shin, N. R., Yun, D. Y., & Hwang, C. (2020). *Artificial neural network algorithm comparison for exchange rate prediction*. 12(3), 125–130.
- Wang, J., & Hu, J. (2015). A robust combination approach for short-term wind speed forecasting and analysis - Combination of the ARIMA (Autoregressive Integrated Moving Average), ELM (Extreme Learning Machine), SVM (Support Vector Machine) and LSSVM (Least Square SVM) forecasts. *Energy*, 93, 41–56. <https://doi.org/10.1016/j.energy.2015.08.045>
- Wei Huang, K.K. Lai, Y. Nakamori, S. W. (1995). Forecasting foreign exchange rates with artificial neural networks. *Intelligent Engineering Systems Through Artificial Neural Networks*, 5(1), 771–778.
- Wei, Y., Sun, S., Ma, J., Wang, S., & Lai, K. K. (2019). A decomposition clustering ensemble learning approach for forecasting foreign exchange rates. *Journal of Management Science and Engineering*, 4(1), 45–54. <https://doi.org/10.1016/j.jmse.2019.02.001>