Available online at http://scik.org

Commun. Math. Biol. Neurosci. 2025, 2025:89

https://doi.org/10.28919/cmbn/9170

ISSN: 2052-2541

THE ASYMMETRIC VOLATILITY MODEL OF INDONESIAN CRUDE OIL PRICE WITH ARTIFICIAL NEURAL NETWORKS HYBRIDIZATION

DODI DEVIANTO[1,*], AULIA ZUARDIN[1], RAHMAWATI RAMADHAN[1], MAIYASTRI[1],

AFRIDHO AFNANDA[1], MASYHURI HAMIDI[2], AFRIDIAN WIRAHADI AHMAD[3]

[1]Department of Mathematics and Data Science, Universitas Andalas, Padang 25163, Indonesia

[2]Department of Management, Universitas Andalas, Padang 25163, Indonesia

[3]Department of Accounting, Politeknik Negeri Padang, Padang 25163, Indonesia

Abstract: Volatility is a measure of changes in security prices that can be used to measure investment risk. Volatility can be detected through a variance model that contains heteroscedasticity effects in the form of ARCH/GARCH. If volatility contains asymmetric effects, it can be overcome by exponential and threshold models, namely the EGARCH and TGARCH models. Although it can overcome heteroscedasticity and asymmetric effects, this variance model needs to maintain model stability and accuracy levels, so the neural networks of multilayer perceptron (MLP) and radial basis function (RBF) are proposed as a neuroinformatics approach. This study uses Indonesian crude oil price (ICP) data from January 2011 to December 2024. The modeling results show that the variance model with the hybrid MLP and RBF neural networks approach is able to provide better accuracy, although the EGARCH and TGARCH models can still overcome asymmetric volatility. The hybrid neural networks approach can provide more precise information on variance models that can be used as an alternative in investment decisions related to ICPs that have an asymmetric volatility structure with irregular price fluctuations.

Keywords: volatility; neural networks; GARCH; EGARCH; TGARCH.

2020 AMS Subject Classification: 62M10, 68T27, 68T07.

[*]Corresponding author

E-mail address: ddevianto@sci.unand.ac.id

Received February 1, 2025

1. INTRODUCTION

In analyzing energy needs in the oil and gas sector, a deep understanding of crude oil price behavior is key in designing business strategies, managing risks, and making investment decisions [1,2,3]. The increasing need for energy from oil requires a structured analysis of price stability parameters for both supply and demand. Oil price movements usually follow fluctuations in world crude oil. Crude oil is a natural, unrefined liquid and is an important natural resource and a vital commodity that affects almost every sector of the global economy. The use of crude oil through a refining process into products such as gasoline, jet fuel, and other petroleum products with varying selling prices in each country [4,5].

Crude oil price movements fluctuate and are influenced by trading in the commodity market. One of the parameters used to represent crude oil prices is the Indonesia crude oil price (ICP). Time series analysis of ICP data can provide significant insights into crude oil price volatility, the factors that influence it, and its implications for the Indonesian economy [4,6]. Global crude oil prices have experienced significant volatility over the decades. This volatility is influenced by various factors, including changes in supply and demand, geopolitical conflicts, changes in energy policy, and technological developments.

Volatility models can provide important information about price fluctuations in certain commodities [7,8,9]. Changes and movements in ICP volatility can affect the energy sector, fiscal policy, and the economic welfare of the country. Time series analysis methods that will be used in analyzing volatility in certain commodity prices involve the formation of GARCH, EGARCH, TGARCH models for handling asymmetric volatility in nonstable variance. The asymmetric variance model can be developed by using the property of infinitely divisible family exponential distribution [10,11], and change in volatility data structure [7,12]. Furthermore, neuroinformatic approaches through hybrid models with artificial neural networks multilayer perceptron (MLP) and radial basis function (RBF) can be used to capture volatility [12,13,14,15]. Volatility models that have asymmetric data and structural changes can be well adapted to these models, especially hybridization with neural networks to maintain stability and accuracy [16,17,18,19].

The performance evaluation of each model in predicting changes in crude oil prices will be assessed based on aspects such as predicted volatility, exponential effects, threshold effects, and asymmetry, which will be the focus of the analysis. The results of this case study are expected to provide practical guidance for market players, energy analysts, and policymakers in managing risks associated with crude oil price fluctuations. In addition, this study can contribute to the literature on time series analysis, especially in the context of volatility and crude oil price movements and their influence in the study of inflation movements [20,21], precious metal commodity price movements [22,23], and agricultural commodities [24,25,26]. The volatility model on specific commodities has a significant impact on economic stability with the growth of macroeconomic indicators observed in Indonesia and globally [27,28].

Modeling the volatility of ICP price movements can be done through time series analysis with a variance model approach. The model built is expected to find patterns and key factors that can be the basis for better decision-making in various aspects of the economy and industry when investing in the securities market. The hybrid approach in this study uses neural networks in an effort to maintain model stability and increase accuracy in the volatility model.

2. Material and Methods

2.1 Material and Data Source

This study uses ICP data from January 2011 to December 2024, with a total of 168 data obtained through the website https://www.ceicdata.com/en/indonesia/crude.oil.price. ICP price movement data will be determined by the GARCH, EGARCH, TGARCH volatility models and the proposed new hybrid model for strengthening stability and accuracy through the MLP and RBF neural networks approach.

2.2 Conceptual and Theoretical Approach

At this stage, data analysis is carried out to model the volatility of ICP price movements in the securities market using the GARCH, EGARCH, TGARCH models, and stability and accuracy enhancement models through the hybrid neural networks MLP and RBF models.

2.2.1 The GARCH Variance Model

In the basic time series model with the autoregressive integrated moving average (ARIMA) approach with variance of residual that does not only depend on the residual of the previous period but also depends on the variance of residual. Based on this, the autoregressive conditional heteroscedasticity (ARCH) model is developed by including the residual elements of the previous period and its variance of residual. The more common variance model is known as generalized autoregressive conditional heteroscedasticity (GARCH). The first-order variance model in the GARCH(1,1) model can be expressed in the following Equation (1):

$$\sigma^2{}_t = \alpha_0 + \alpha_1 e^2{}_{t-1} + \beta_1 \sigma^2{}_{t-1}, \tag{1}$$

where $\sigma^2{}_t$ is a variance model that is influenced by the square of the previous period residuals ($e^2{}_{t-1}$), but also by the variance of the previous period residuals ($\sigma^2{}_{t-1}$). The variance model is denoted as GARCH(1,1) because its variance residuals are only influenced by one previous period. If the variance of the residuals is influenced by the residuals of the previous period (lag p of the ARCH elements) and the variance of the residuals of the previous period (lag q of the GARCH elements), then the GARCH(1,1) model can be expressed as follows:

$$\sigma^2{}_t = \alpha_0 + \alpha_1 e^2{}_{t-1} + \cdots + \alpha_p e^2{}_{t-p} + \lambda_1 \sigma^2{}_{t-1} + \cdots + \lambda_q \sigma^2{}_{t-q}. \tag{2}$$

Equation (2) shows that the residual variance ($\sigma^2{}_t$) is not only influenced by a number of parameters as in parametric models, but also contains up to the p-th residual square for the q-th variance of residual, so that the GARCH model can also be estimated using the maximum likelihood estimation (MLE) method or with a computational approach.

2.2.2 The EGARCH Asymmetric Volatility Model

The asymmetric volatility model can be expressed in the form of an exponential GARCH (EGARCH) model, which is a variance model that accommodates the possibility of asymmetric volatility responses. In general, the EGARCH model with order p and q denoted as EGARCH(p,q) can be given by the following formula [7]:

$$\ln(\sigma_t^2) = \omega + \sum_{j=1}^{q} \beta_j \ln\left(\sigma_{t-j}^2\right) + \sum_{i=1}^{p} \alpha_i \left|\frac{e_{t-i}}{\sigma_{t-i}}\right| + \sum_{k=1}^{r} \gamma_r \frac{e_{t-k}}{\sigma_{t-k}} \ , \tag{3}$$

where $\omega, \alpha_i, \beta_j,$ and γ_r are the parameters to be estimated. The advantage of using EGARCH is that it does not need to meet the stationarity requirement. In Equation (3) it can be seen that the left side uses the natural logarithm of the conditional variance, which shows that the leverage effect is exponential, not quadratic or other. The use of the natural logarithm for the variance equation has guaranteed the nonnegative nature of the variance. The asymmetric effect occurs if $\gamma_i \neq 0$, which means γ is significant. The existence of the leverage effect is marked by the hypothesis $\gamma < 0$ [10,12].

2.2.3 The TGARCH Asymmetric Volatility Model

Another development of the GARCH model that accommodates the asymmetric volatility response is the GARCH threshold model (TGARCH) which is also known as the GJR-GARCH model according to the initials of the model's proposer. In general, the TGARCH model can be defined in the following formula:

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \, e_{t-i}^2 + \sum_{j=1}^{q} \beta_j \, \sigma_{t-j}^2 + \sum_{k=1}^{r} \gamma_k \, e_{t-k}^2 \, I_{t-k}. \tag{4}$$

where $\omega, \alpha_i, \beta_j, \gamma_k$ is the estimated parameter. The identity value of $I_{t-1} = 1$ if $\varepsilon_{t-i} < 0$, and $I_{t-1} = 0$ if $e_{t-i} > 0$. The decision of the volatility state is based on the event of good news at $e_{t-i} > 0$ and bad news at $e_{t-i} < 0$, which have different effects on the conditional variance in the *t*-period. Good news has an effect on the parameter γ_i, while bad news has an effect on the parameter $\gamma_i + \alpha_i$. If $\gamma_i > 0$ on bad news, it will leave volatility, also called the leverage effect. The asymmetric effect occurs when $\gamma_i \neq 0$ which means γ is significant. The stationarity condition on TGARCH is in accordance with Equation (5) as follows:

$$\sum_{i=1}^{p} \alpha_i + \sum_{j=1}^{q} \beta_j + \sum_{k=1}^{r} \gamma_k < 1. \tag{5}$$

This stationary model with asymmetric effects is stated in the TGARCH model as a variance model to describe its volatility.

2.2.4 Model Adjusted Volatility MLP

The multilayer perceptron (MLP) network consists of three types of neuron layers: input layer, output layer, and hidden layer. In the hidden layer, the activation function processes the input data before being forwarded to the next layer. This function limits the neuron output, usually in the range (0,1) or (−1,1). In this study, the sigmoid activation function is used, which produces values in the range (0,1) as in Equation (7) as follows:

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 + \sum_{k=1}^{r} \gamma_k \varepsilon_{t-k}^2 I_{t-k}. \tag{6}$$

The MLP hidden layer network has an activation function to process input data before being forwarded to the next layer. In this study, the sigmoid activation function is used, which produces a range of values (0,1) as in Equation (7) below:

$$f(x) = \frac{1}{1+e^{-x}} \ , \tag{7}$$

where the derivative of the sigmoid function is $f'(x) = f(x)(1 - f(x))$.

The backpropagation method is a method that is often used in MLP network learning algorithms as follows:

1. Determine the amount of input data. Load the time series X_t dataset into the artificial neural network as an input unit.
2. Defining the input and output data to be achieved
3. Normalization of input and output data using the equation t_{t1}=(Y_t,Y_{min})/(Y_{max},Y_{min}) where t_{t1} is the *t*-th output normalization data for *t* = 1, 2, …, n. Furthermore, to normalize input data X_t, the same method can be used as output data.

4. Split the data set into training data and testing data..
5. The design of artificial neural network architecture can be determined from the number of units in each layer.
6. The process of training artificial neural networks.
   a. Initialize the network. Initial values of the weights need to be determined in order to perform the training process. Artificial neural networks are generally initialized with random weights.
   b. Feed forward. For each unit where x_{ti} for i = 1, 2, …, N, then the feed forward steps can be arranged as follows:
      i. In the hidden layer, calculate $in_{tp}^{h} = w_{p0}^{h} + \sum_{i=1}^{N} x_{ti} w_{pi}^{h}$.
      ii. Determine $\imath_{tp} = f_p^h(in_{tp}^h)$, where f_p^h is the activation function.
      iii. In the output layer, calculate the value of $in_{tj}^{o} = w_{j0}^{o} + \sum_{p=1}^{L} \imath_{tp} w_{jp}^{o}$.
      iv. Determine $y_{tj} = f_j^o(in_{tj}^o)$.
   c. Residual assessment. If $|(t_{t1} - y_{t1})| > 0.001$, the *t*-data then the stopping criteria have not been met. So, the artificial neural network training process cannot be stopped.
   d. The backpropagation method with the following steps:
      i. Calculate $\delta_{tj}^{o} = \left(t_{tj} - y_{tj}\right) f_j^{o\prime}(in_{tj}^{o})$ to find out the residual information on the output unit.
      ii. Calculate $\delta_{tp}^{h} = f_p^{h\prime}(in_{tp}^{h})\left(\delta_{tj}^{o}\right) w_{jp}^{o}$ to find out the residual on each hidden layer unit.
      iii. On the output layer, calculate $\Delta w_{jp}^{o} = \eta \delta_{tj}^{o} \imath_{tp}$ to find out the weight and bias correction values.
      iv. On the hidden layer, calculate $\Delta w_{pi}^{h} = \eta \delta_{tp}^{h} x_{ti}$ to find out the weight and bias correction values.
   e. Weight adjustment. The steps for weight adjustment are as follows:
      i. At the output layer, calculate $w_{jp}^{o}(t+1) = w_{jp}^{o}(t) + \Delta w_{jp}^{o}(t)$.
      ii. At the hidden layer, calculate $w_{pi}^{h}(t+1) = w_{pi}^{h}(t) + \Delta w_{pi}^{h}(t)$.

The notation used is where $w_{pi}^{h}(t+1)$ and $w_{jp}^{o}(t+1)$, which states the new weights that will be used in the next process.

7. The validation process of the artificial neural network model using the decision principle. If the validation results show that the MSE value of the test is greater than the MSE value of the training results, then an additional training process is carried out.
8. Final MLP modeling and model evaluation.

2.2.5 Model Adjusted Volatility RBF

The radial basis function (RBF) network has a network model that is almost similar to the MLP artificial neural network, where this artificial neural network changes the input non-linearly using the Gaussian activation function in the hidden layer, before continuing to the linear process in the output layer.

The training stages with the RBF algorithm can be explained as follows:

1. Input data for the training process.
2. Determine the number of hidden neurons.
3. The training process begins by normalizing the data so that the data will be in the range of (0,1).
4. The next step in the training process is to calculate the weights between the input layer and the hidden layer.
5. Find the value of the spread that will be used with the equation $\sigma = d_{max}/\sqrt{m}$.
6. In the Gaussian activation function, calculate the RBF network using the following equation:

$$\psi(x, c_i) = \exp\left(-\frac{||x-c||^2}{2\sigma^2}\right). \tag{8}$$

7. Forming a Gaussian matrix from the calculation results in Step 6.
8. Calculating the weights from the hidden layer to the output layer using the pseudo-inverse matrix linear equation system. where the new weights (w) are calculated by multiplying the pseudo-inverse of the matrix by the target vector (d) from the training data, namely

$$w = (G^T G)^{-1}\, G^T\, d. \tag{9}$$

9. Training data is done by trial and error.
10. The testing process is carried out by entering training weights, training data, and real data. Furthermore, the output of the testing process is as follows:

$$f(x) = \textstyle\sum_{j=1}^{m} w_j \psi(x, c_i). \tag{10}$$

11. Final RBF modeling and model evaluation.

2.2.6 Best Model Evaluation and Model Error Rate Measurement

The measurement of the best model in this study uses the Akaike information criterion (AIC) and Bayesian information criterion (BIC) in a few canonical statistical problems and state the results of its statistical optimality therein. As a comparison for the best selection model, in addition to AIC, the measurement of modeling error is also used to evaluate model performance by comparing prediction results with actual data, as well as for BIC and vice versa. The equations for AIC and BIC are stated as follows:

$$AIC = -2\log \hat{L}(\theta \mid data, g) + 2K \tag{11}$$

$$BIC = -2\log \hat{L}(\theta \mid data, g) + \log(n) K \tag{12}$$

where $\hat{L}(\theta \mid data, g)$ is the likelihood function of model parameters given in the data and the model g, the synmbol of K is the number of parameters to be estimated.

The error measurement process can be expressed in model accuracy, such as mean absolute error (MAE), which is used to evaluate the model using the sum of absolute errors. Furthermore, the root mean squared error (RMSE) can also be used, which is the average value of the difference between the forecast results and the actual values that have been squared, then the square root is calculated. Meanwhile, the mean absolute percentage error (MAPE) is the average of the overall percentage of errors (differences) between the actual data and the modeling data. The formulation of the three accuracy measures can be stated as follows:

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |X(t) - F(t)|, \tag{13}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n} (X(t) - F(t))^2}, \tag{14}$$

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n} \left|\frac{X(t) = F(t)}{X(t)}\right|. \tag{15}$$

The smaller the value produced by the four methods, the better the model used.

3. RESULT AND DISCUSSION

3.1 Stationary Model of ICP Movement

This study uses monthly ICP data from January 2010 to December 2024, with a total of 180 data obtained through the Indonesia Central Bureau of Statistics. The data that has been obtained is then used to observe data patterns to see stationarity by making plots of the data. The data plot can be seen in Figure 1. The movement of the ICP price fluctuates with various trends, where there was a decline in 2014 until the year detected in 2020 when the COVID-19 pandemic hit Indonesia, which affected the securities market, including the movement of the ICP price.

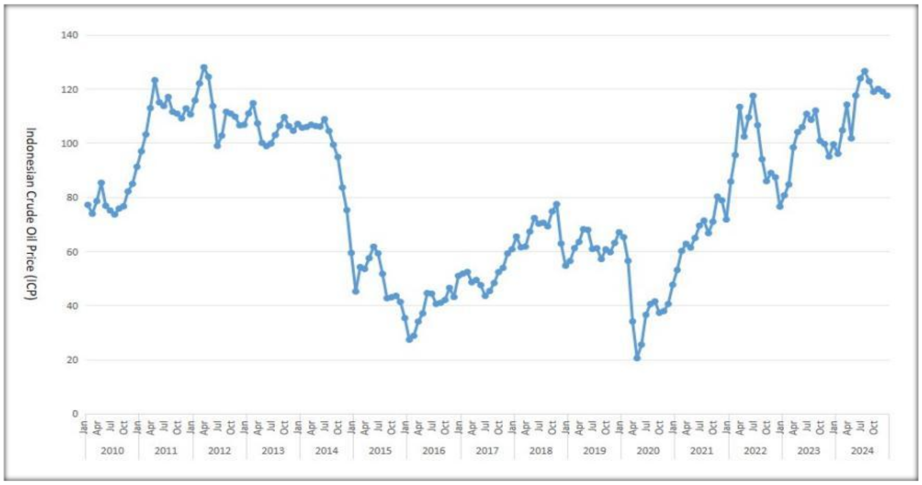

Figure 1. Monthly ICP data.

The volatility model is built based on the ARIMA model by considering the heteroscedasticity effect. The model to be built is a stationary model. The steps in forming the model are first, the ICP data is converted into a natural logarithm with the following formulation:

$$R_t = \ln\left(\frac{d_t}{d_{t-1}}\right), \tag{15}$$

where R_t is the natural logarithm at time t, the notation dt represents the data at time t, and $d_{t\text{-}1}$ is the data at time (t-1). The natural logarithm data of ICP is presented in Figure 2, which graphically shows that the fluctuations indicate stationarity.

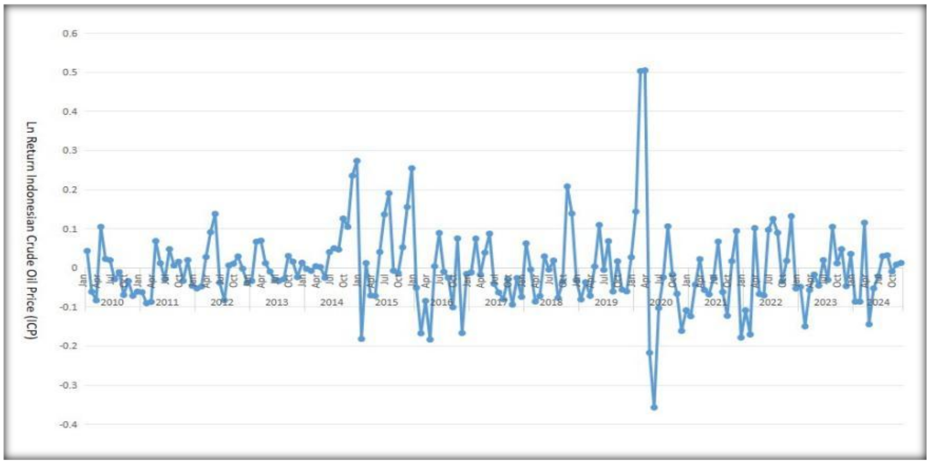

Figure 2. Plot of ICP natural logarithm data.

Next, the ADF stationarity test is carried out on the ICP natural logarithm data with a *p-value* = 0.01, which is smaller than the significant level $\alpha = 0.05$, then the ICP natural logarithm data is stated as stationary. Next, to check the stationarity of the data against the variance, the Box-Cox test is used with λ symbolized as a transformation parameter, and the $\lambda = 1$ value is obtained, which indicates that the ICP natural logarithm data is stationary against the variance. Because the ICP natural logarithm data is stationary against the mean and variance, the next step is to identify the ARIMA model.

The ARIMA model identification is made based on the autocorrelation function (ACF) and partial autocorrelation function (PACF) values. The ACF and PACF results are presented in a plot like Figure 3.

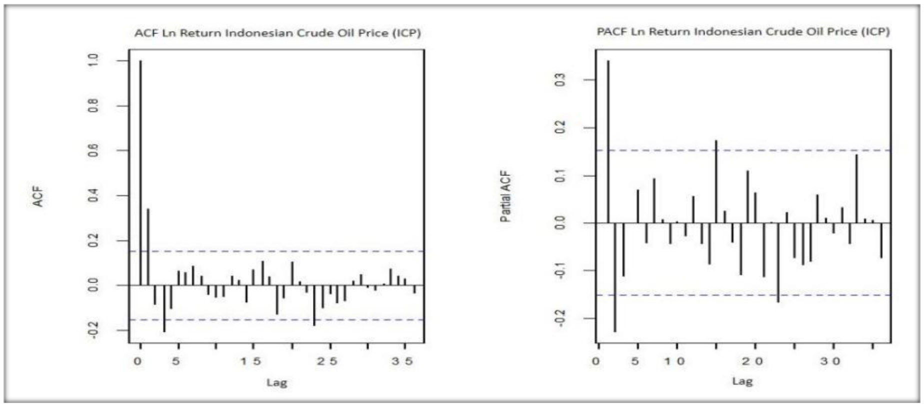

Figure 3. ACF and PACF plot of natural logarithm data of ICP.

In Figure 3, the ACF plot of the ICP natural logarithm data shows the most significant ACF coefficient values at the first and second lags, while the PACF coefficient values are also significant at the first and second lags. The ARIMA model can be formed by looking at the significance of the ACF and PACF values; the parameter estimation results are shown in Table 1 with a model with significant parameters, where the best model is obtained with the smallest AIC and BIC values, namely the ARIMA(0,0,1) model.

Table 1. Estimated parameter values, AIC and BIC criteria for the ARIMA model.

Model	AR(1)	AR(2)	MA(1)	MA(2)	σ^2	AIC	BIC
ARIMA(0,0,1)	-	-	0.4024	-	0.0095	-300.09	-270.11
ARIMA(0,0,2)	-	-	0.4215	0.0405	0.0095	-298.25	-265.37
ARIMA(1,0,0)	0.3439	-	-	-	0.0098	-294.95	-266.95
ARIMA(1,0,1)	0.0506	-	0.3597	-	0.0095	-298.16	-264.34
ARIMA(1,0,2)	0.9478	-	.0.5835	0.4165	0.0092	-298.87	-267.71

Next, a diagnostic test is performed on the ARIMA(0,0,1) model to see if there are any heteroscedasticity, autocorrelation, and residual normality effects. Based on Table 2, it can be seen that the heteroscedasticity effect test using Lagrange multiplayer shows that there is a non-constant variance in the residual. This effect must be overcome with a variance model, which will be the main focus of this study. Meanwhile, based on the Ljung-Box test shows that the residual data has no autocorrelation, while based on the Shapiro-Wilk test shows that the residual data is normally distributed.

Table 2. Heteroscedasticity, autocorrelation, and normality tests for the residual of ARIMA(0,0,1) model.

Diagnostic Test	**Test Statistic**	***p-value***	**Decision**
Heteroscedasticity	Lagrange Multiplayer	2.4351e+08	There is a heteroscedasticity effect
Autocorrelation	Ljung-Box	8.2518e-06	There is no autocorrelation
Normality	Shapiro-Wilk	3.2110e+05	Normally distributed

The ICP movement model shows that there is a short-term influence that is formed into an autoregressive model. The fluctuation of the ICP shows a dynamic role in oil trading in the securities market. The dynamics of the ICP movement require further analysis by describing its volatility. Volatility in the financial context can be understood as a measure of how much the price of an asset fluctuates or changes over a certain period of time. The volatility model is an important tool in risk management because it indicates the level of uncertainty in the ICP movement.

3.2 The GARCH as the Variance Model of ICP Volatility

The GARCH model is a model that can overcome the variance of residuala that have heteroscedasticity properties. Identification of the GARCH model is done by making ACF and PACF plots of the quadratic residuals of the ARIMA(0,0,1) model, as can be seen in Figure 4.

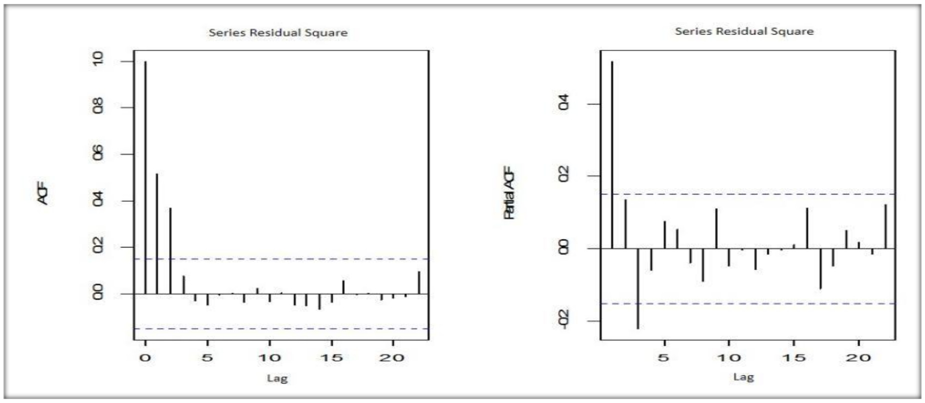

Figure 4. Plot ACF and PACF on ARIMA(0,0,1) model residuals.

From the results of the ACF and PACF plots in Figure 4, it can be seen that the lag value in the ACF is significant at the first, second, and third lags. While the PACF plot shows a significant value at the first and third lags, so there are several possible GARCH models with significant parameter estimation results, which are presented in Table 3.

Table 3. Parameter value estimation, AIC, and BIC value of the GARCH model.

Model	ω	α_1	α_2	α_3	β_1	β_2	β_3	**AIC**	**BIC**
GARCH(1,0)	0.0036	0.7867	-	-	-	-	-	-2.0316	-1.994
GARCH(1,1)	0.0018	0.7727	-	-	0.2083	-	-	-2.0463	-1.9903
GARCH(1,3)	0.0001	0.0761	-	-	0.0258	0.0001	0.0002	-2.0233	-1.9305
GARCH(2,0)	0.0026	0.7430	0.1714	-	-	-	-	-2.0442	-1.988
GARCH(2,1)	0.0020	0.7599	0.0533	-	0.1495	-	-	-2.0341	-1.9594
GARCH(2,3)	0.0001	0.0758	0.0054	-	0.0221	0.0001	0.0005	-2.0117	-1.8999
GARCH(3,0)	0.0024	0.7536	0.1639	0.0227	-	-	-	-2.0329	-1.9582
GARCH(3,1)	0.0001	0.0761	0.0001	0.0001	0.0207	-	-	-2.0218	-1.9284

Based on Table 3, the best model with the smallest AIC and BIC values is the GARCH(1,1) model. This model means that the residual variance is influenced by one previous data lag and is also influenced by one previous residual square lag. This short-term influence provides investors with the opportunity to make quick decisions amidst the fluctuations in ICP price movements in the securities market.

3.3 The EGARCH Asymmetric Variance Model as ICP Volatility

The EGARCH model is a time series model used to capture volatility with asymmetric effects. In addition, it is used to measure asymmetric effects through the logarithm of the conditional variance in GARCH(1,1). The parameter estimation and coefficient values of the model are presented in Table 4.

Table 4. Parameter estimation of EGARCH(1,1).

Model	**Estimation**	**Standard error**	*t-value*	Pr(>|t|)
$\boldsymbol{\phi_1}$	-0.8946	0.0001	-24094.47	<0.0001
$\boldsymbol{\phi_2}$	-0.9290	0.0002	-7289.22	<0.0001
$\boldsymbol{\theta_1}$	0.3571	0.0001	3900.01	<0.0001
$\boldsymbol{\theta_2}$	0.5840	0.0001	8052.23	<0.0001
$\boldsymbol{\theta_3}$	-0.5261	0.0001	-3773.52	<0.0001
$\boldsymbol{\omega}$	-4.4214	0.0008	-5496.03	<0.0001
$\boldsymbol{\alpha_1}$	0.2920	0.0004	778.41	<0.0001
$\boldsymbol{\beta_1}$	0.5014	0.0009	577.35	<0.0001
$\boldsymbol{\gamma_1}$	-1.7270	0.0012	-1410.08	<0.0001

Based on the parameter estimation of EGARCH(1,1) in Table 4, it can be seen that the model contains asymmetric effects with the value γ_1 being negative; this is to confirm that volatility was more responsive to negative shocks than positive.

3.4 The TGARCH Asymmetric Variance Model as ICP Volatility

The TGARCH is a model that also captures asymmetric effects in volatility. Positive and negative shocks to the measured variables have different impacts on volatility where the parameter estimation and coefficient values of the model are presented as follows in Table 5.

Table 5. Parameter estimation of TGARCH(1,1).

Model	**Estimation**	**Standard error**	t-value	Pr(>|t|)
$\boldsymbol{\phi_1}$	-0.2237	0.0534	4.1871	0.0001
$\boldsymbol{\phi_2}$	-0.8784	0.0778	-11.2885	<0.0001
$\boldsymbol{\theta_1}$	-0.4122	0.1099	-3.7476	0.0002
$\boldsymbol{\theta_2}$	0.9705	0.0613	-3.7476	<0.0001
$\boldsymbol{\theta_3}$	.0.5618	0.1125	15.8114	<0.0001
$\boldsymbol{\omega}$	0.0040	0.0029	1.3458	0.1783
$\boldsymbol{\alpha_1}$	0.1490	0.0927	1.6053	0.1783
$\boldsymbol{\beta_1}$	0.5942	0.2367	2.5096	0.0120
$\boldsymbol{\gamma_1}$	-1.0000	0.9775	-1.0229	0.3063

Based on Table 5, the parameter γ_1 confirmed asymmetric effects on ICP return data by showing a negative value. Hence, the volatility in TGARCH(1,1) was more responsive to negative shocks than positive. However, based on the *t-value* and *p-value* of parameter γ_1 was not statistically significant, the model of TGARCH(1,1) showed insignificant and asymmetric effects on ICP return data. This confirmed there was no significant bias in the residuals, either from positive or negative shocks. The variance model of TGARCH(1,1) had the ability to capture the dynamics of volatility, although asymmetric effects were insignificant. This was because the model showed various diagnostic tests fit the data, and there was no remaining autocorrelation or heteroscedasticity. The model of TGARCH(1,1) also showed good fit results in measuring the ICP logarithm natural of return data.

3.5 The Model Adjusted Volatility with Neural Networks Approach MLP and RBF

In this section, a neural network model will be built with a neuroinformatics approach where the variance model is taken over to become an artificial neural network model. The neural networks model will be applied to the residual ARIMA (0,0,1) data using the MLP and RBF models. First, the MLP neural networks architecture will be created as in Figure 5, optimally built from two input units, three units in the hidden layer, and one output unit. The neural networks to be built uses the backpropagation method with a sigmoid activation function between layers. The weight training process on the artificial neural networks with the backpropagation method is explained in the methodology section.

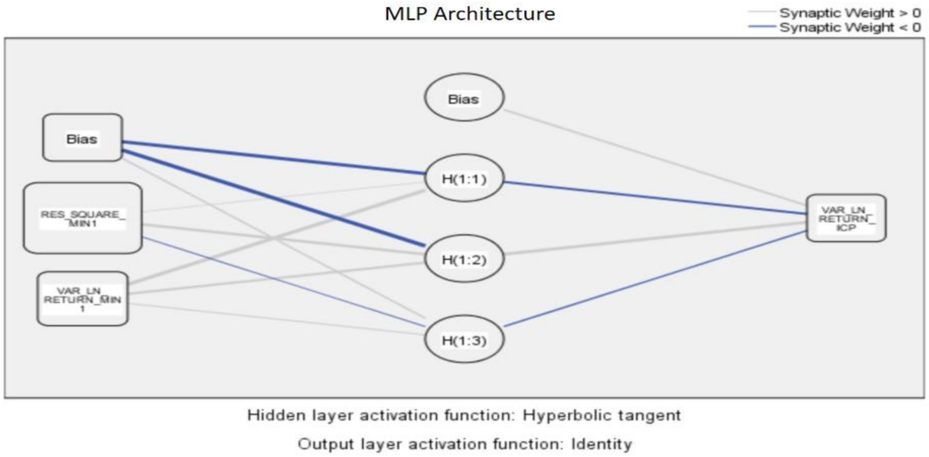

Figure 5. MLP Architecture for a residual model of ARIMA(0,0,1).

Table 6. Parameter estimation of MLP.

Predictor		Predicted			
		Hidden-Layer-1			Output-Layer
		H(1:1)	H(1:2)	H(1:3)	VAR_ICP
Input-Layer	(Bias)	-1.4512	-1.9142	0.2962	
	RES_MIN1	0.1084	1.0872	-.1192	
	VAR_MIN1	1.5695	0.9382	0.2492	
Hidden- Layer-1	(Bias)				0.4665
	H(1:1)				-0.7272
	H(1:2)				1.1172
	H(1:3)				-0.2643

In this MLP model, a neural network model will be built with a neuroinformatics approach, and the variance model will be taken over to become an artificial neural network MLP model. The neural networks model with networks initialization is based on 70% training data and 30% testing data, where the weight selection is done randomly. The training process is stopped when the residue (error) reaches 0.0001 between the output of the artificial neural networks and the desired target output.

Table 6 presents the results of parameter estimation using the MLP method, where the networks structure includes two input and bias layers, three hidden layers, and one output layer. Each parameter in the table describes the contribution of each predictor variable to the neurons in the hidden layer and the output layer. After the information from the hidden layer is processed, the output value of the neurons in the hidden layer is sent to the output layer. The estimated parameters indicate the connection weights from the neurons in the hidden layer to the output layer. Thus, Table 6 describes how the weights and biases affect the flow of information through the MLP artificial neural networks, starting from the input layer, through the hidden layers, to the final output. This indicates that the MLP has succeeded in recognizing new data well through the model built during the training process.

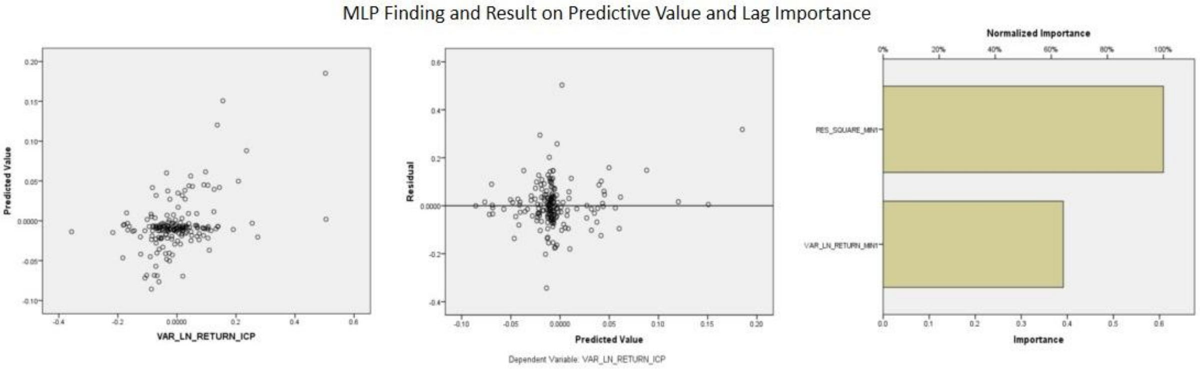

Figure 6. Predicted value and lag importance on MLP model.

Based on Figure 6, the left part shows the relationship between variance and predicted value of the MLP model. The data distribution appears concentrated around zero, with some values spreading further randomly; this shows that the model can capture data behavior with small errors. In Figure 6, the center part shows the relationship between residual values and predicted values, residuals that are randomly spread around the zero line; this shows that the MLP model no longer shows a certain systematic pattern in its errors, which indicates that the model works quite well. In Figure 6, the right part is a bar graph showing the importance value that contributes from each variable to the final result of the MLP model, which is able to capture the relationship between lag variables and asymmetric volatility values, where residuals are a more dominant factor than variance in influencing prediction results. This provides information that the volatility model for ICP data with the MLP neural networks approach is able to provide good and stable predictions.

The next process is to build a variance model through a neuroinformatics approach with the RBF method. This model is carried out based on the analysis of the ACF and PACF plots of the residual ARIMA(0,0,1) model, which is significant at the first lag. This RBF model is also built based on 70% training data and 30% testing data, where the selection of weights is done randomly. The RBF architecture is used to process this data, as shown in Figure 7. This RBF architecture has a configuration with two input units, two hidden layers, and one output unit. This model utilizes radial basis functions to transform inputs non-linearly and produce output predictions on the output layer. The process of training weights on artificial neural networks with the RBF method is described in the methodology section.

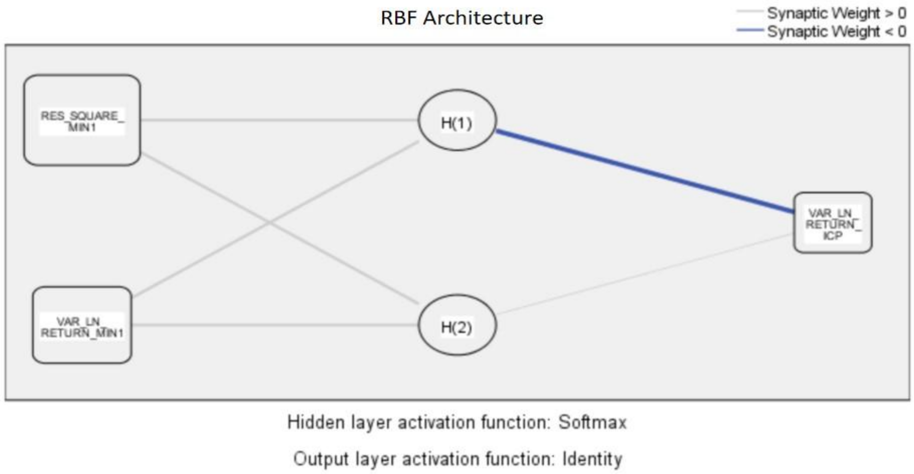

Figure 7. RBF Architecture for a residual model of ARIMA(0,0,1).

Table 7. Parameter estimation of RBF.

Predictor		Predicted		
		Hidden-Layer		Output-Layer
		H(1)	H(2)	VAR_ICP
Input-Layer	RES_MIN1	-0.0682	2.1893	
	VAR-MIN1	-0.0342	1.0822	
Hidden-Uni- Width		0.8421	4.1751	
Hidden-Layer	H(1)			-1.1427
	H(2)			0.7059

Before processing the data, the initial step in RBF is data normalization to change its value into the range of (0,1); this is to improve the accuracy and stability of training. The data is then divided into training data and testing data to ensure that the model can generalize well on new data.

This RBF architecture is obtained by randomly initializing the weights on the hidden layer, and the output weights are calculated using the least square method to minimize errors. The results of parameter estimation can be seen in Table 7.

The RBF learning process consists of two stages: determining the center and width of the radial basis function and determining the connection weight between the hidden layer and the output layer. The RBF modeling process can predict nonlinear time series data well and interpolate missing data by minimizing errors. The ability of RBF to perform nonlinear transformations has made this method widely applied in various fields. However, the selection of RBF parameters, such as the center and width of the optimal basis function, is still a challenge. Table 7 shows the parameters in the hidden layer after the validation process is complete; this indicates that RBF has also succeeded in recognizing new data well through the model built during the training process.

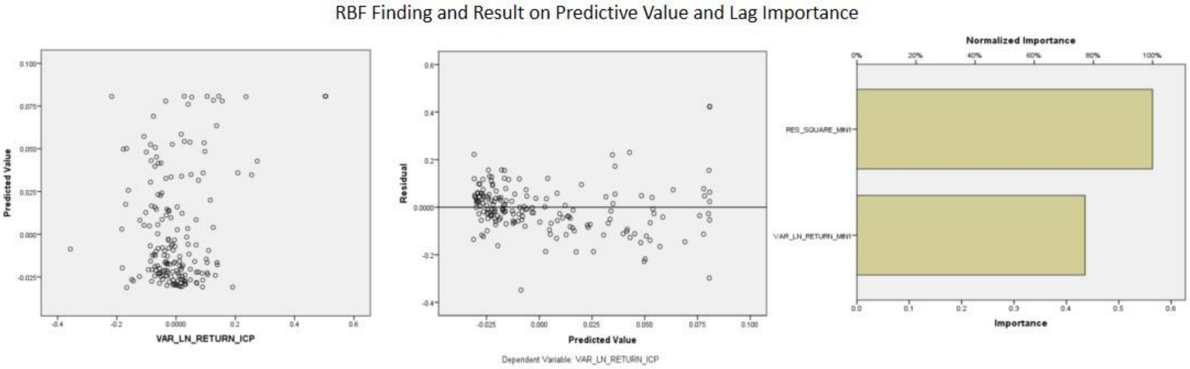

Figure 8. Predicted value and lag importance on RBF model.

Based on Figure 8, the left side shows the relationship between variance and predicted value of the RBF model. The data distribution still appears concentrated around zero, with some values spreading further upward randomly; this shows that the model can still capture data behavior with small errors. In Figure 8, the center part shows the relationship between residual values and predicted values, residuals that are randomly spread around the longer zero line; this shows that the RBF model no longer shows a certain systematic pattern in its errors, although some of its values are in the negative part, this still indicates that the model works quite well. In Figure 8, the right side is a bar graph showing the importance value that contributes from each variable to the final result of the MLP model, which is able to capture the relationship between lag variables and asymmetric volatility values, where residuals are a more dominant factor than variance in influencing prediction results. This provides information that the volatility model for ICP data with the RBF neural networks approach is able to provide good and stable predictions.

3.6 The Hybrid Volatility Model with Neural Networks Approach MLP and RBF

In this section, a hybrid model will be built using the residual variance model that has been built through the neural networks approach with the neuroinformatics approach, namely using the MLP and RBF models. The classical volatility model GARCH and the asymmetric models EGARCH and TGARCH will be hybridized with neural networks.

In the first stage, a hybrid GARCH-MLP model will be built, namely taking the residual from the GARCH(1,1) volatility model into the MLP neural networks model. The hybrid GARCH-MLP model, as in Figure 9, is optimally built from two input units, two units in the hidden layer, and one output unit. The neural networks to be built on the hybrid GARCH-MLP model uses the backpropagation method with a sigmoid activation function between layers.

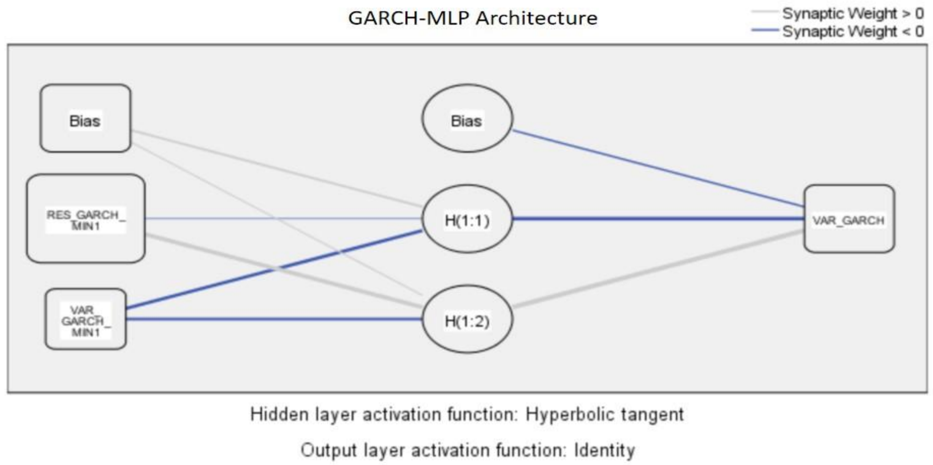

Figure 9. GARCH-MLP Architecture for a residual model of GARCH(1,1).

Table 8. Parameter estimation of GARCH-MLP.

Predictor		Predicted		
		Hidden-Layer-1		Output-Layer
		H(1:1)	H(1:2)	VAR_GARCH
Input-Layer	(Bias)	0.1451	0.0595	
	RES_GARCH_MIN1	-0.0423	0.5215	
	VAR_GARCH_MIN1	-0.2864	-0.2681	
Hidden-Layer-1	(Bias)			-0.0746
	H(1:1)			-0.4377
	H(1:2)			0.8559

In the GARCH-MLP model, the networks initialization is based on 70% training data and 30% testing data, where the weight selection is done randomly. The training process is stopped when the residual (error) reaches 0.0001 between the output of the artificial neural networks and the desired target output. Table 8 presents the results of the parameter estimation of the GARCH-MLP hybrid model, where each parameter in the table describes the contribution of each predictor variable to the neurons in the hidden layer and the output layer. The resulting parameters indicate that the GARCH-MLP hybrid model successfully recognized new data well through the model built during the training process.

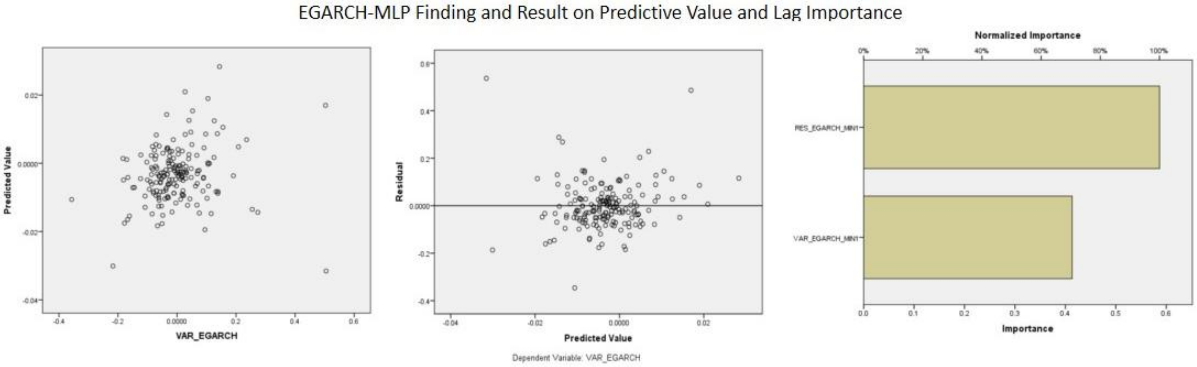

Figure 10. Predicted value and lag importance on GARCH-MLP model.

Based on Figure 10, the left part shows the relationship between variance and predicted value of the hybrid GARCH-MLP model. The data distribution appears concentrated around zero, with some values spreading further randomly, while in Figure 10, the center part shows the relationship between residual values and predicted values, where the residuals are also randomly spread around the more centered zero line. This shows that the hybrid GARCH-MLP model no longer shows a certain systematic pattern in its errors, indicating that the model works quite well. In Figure 10, the right part is a bar graph showing the residual variable as a more dominant factor than the variance in influencing the prediction results. This provides information that the volatility model for ICP data with the hybrid GARCH-MLP approach is able to provide good and more stable predictions.

In the second stage, a hybrid GARCH-RBF model will be built, namely taking the residual from the GARCH(1,1) volatility model into an RBF neural networks model. The hybrid GARCH-RBF model, as in Figure 11, is optimally built from two input units, four units in the hidden layer, and one output unit. The neural networks to be built on the hybrid GARCH-RBF model uses the backpropagation method with a sigmoid activation function between layers.

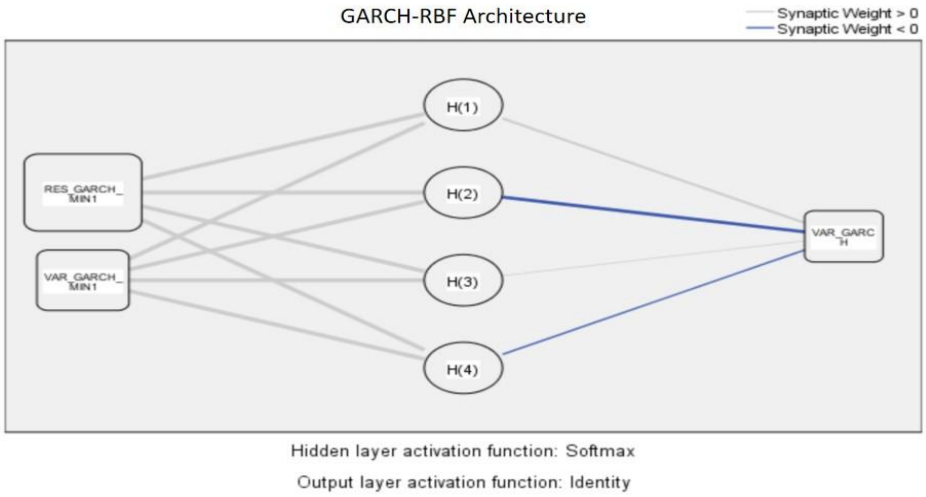

Figure 11. GARCH-RBF Architecture for a residual model of GARCH(1,1).

Table 9. Parameter estimation of GARCH-RBF.

Predictor		Predicted				
		Hidden-Layer				Output-Layer
		H(1)	H(2)	H(3)	H(4)	VAR_GARCH
Input-Layer	RES_GARCH_MIN1	1.8184	-0.2917	0.2816	-0.8688	
	VAR_GARCH_MIN1	1.4633	0.8216	-0.2241	-0.6772	
Hidden-Unit-Width		1.3826	0.4937	0.7428	0.5869	
Hidden-Layer	H(1)					0.5619
	H(2)					-0.9025
	H(3)					0.0885
	H(4)					-0.2835

In the GARCH-RBF model, the network initialization is based on 70% training data and 30% testing data, where the weight selection is also done randomly. The training process is stopped when the residual (error) reaches 0.0001 between the output of the artificial neural networks and the desired target output. Table 9 presents the results of the parameter estimation of the GARCH-RBF hybrid model, where each parameter in the table describes the contribution of each predictor variable to the neurons in the hidden layer and the output layer. The resulting parameters indicate that the GARCH-RBF hybrid model successfully recognized new data well through the model built during the training process.

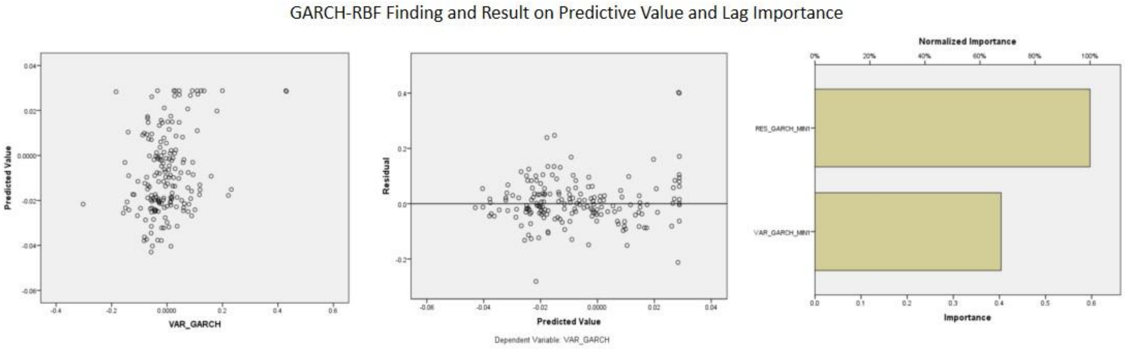

Figure 12. Predicted value and lag importance on GARCH-RBF model.

Based on Figure 12, the left part shows the relationship between variance and predicted value of the hybrid GARCH-RBF model. The data distribution appears concentrated around zero, with some values spreading further upward randomly, while in Figure 12, the center part shows the relationship between residual values and predicted values, where residuals are also randomly spread around the longer zero line. This shows that the hybrid GARCH-RBF model no longer shows a specific systematic pattern in its errors, indicating that the model works quite well. In Figure 12, the right part, as a bar graph, shows the residual variable as a more dominant factor than the variance in influencing the prediction results. This provides information that the volatility model for ICP data with the hybrid GARCH-RBF approach is able to provide good and more stable predictions.

In the third stage, a hybrid EGARCH-MLP model will be built, namely taking the residual from the EGARCH(1,1) volatility model into the MLP neural networks model. The hybrid EGARCH-MLP model, as in Figure 13, is optimally built from two input units, one unit in the hidden layer, and one output unit. The neural networks built on the hybrid EGARCH-MLP model use the backpropagation method with a sigmoid activation function between layers.

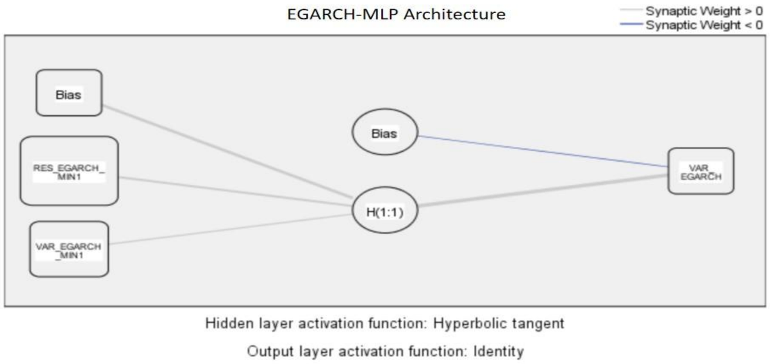

Figure 13. EGARCH-MLP Architecture for a residual model of EGARCH(1,1).

Table 10. Parameter estimation of EGARCH-MLP.

Predictor		Predicted	
		Hidden-Layer 1	Output-Layer
		H(1:1)	VAR_EGARCH
Input-Layer	(Bias)	0.1882	
	RES_EGARCH_MIN1	0.1813	
	VAR_EGARCH_MIN1	0.1158	
Hidden-Layer-1	(Bias)		-0.0971
	H(1:1)		0.3816

In the EGARCH-MLP model, the network initialization is based on 70% training data and 30% testing data, where the weight selection is done randomly. The training process is stopped when the residual (error) reaches 0.0001 between the output of the artificial neural networks and the desired target output. Table 10 presents the results of the parameter estimation of the hybrid EGARCH-MLP model, where each parameter in the table describes the contribution of each predictor variable to neurons in the hidden layer and the output layer. The resulting parameters indicate that the hybrid EGARCH-MLP model successfully recognized new data well through the model built during the training process.

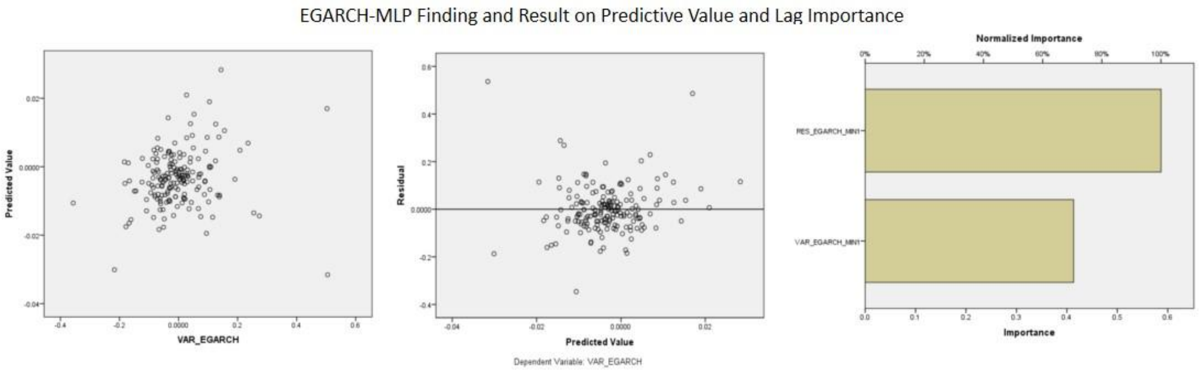

Figure 14. Predicted value and lag importance on EGARCH-MLP model.

Based on Figure 14, the left part shows the relationship between variance and predicted value of the hybrid EGARCH-MLP model. The data distribution appears concentrated around zero, with some values spreading further randomly, while in Figure 14, the center part shows the relationship between residual values and predicted values, where residuals are also randomly spread around the more centralized zero line. This shows that the hybrid EGARCH-MLP model no longer shows a certain systematic pattern in its errors, indicating that the model works quite well. In Figure 14, the right part is a bar graph showing the residual variable as a more dominant factor than the variance in influencing the prediction results. This provides information that the volatility model for ICP data with the hybrid EGARCH-MLP approach is able to provide good and more stable predictions.

In the fourth stage, a hybrid EGARCH-RBF model will be built, namely taking the residual from the EGARCH(1,1) volatility model into an RBF neural networks model. The hybrid EGARCH-RBF model, as in Figure 15, is optimally built from two input units, two units in the hidden layer, and one output unit. The neural networks built on the hybrid EGARCH-RBF model use the backpropagation method with a sigmoid activation function between layers.

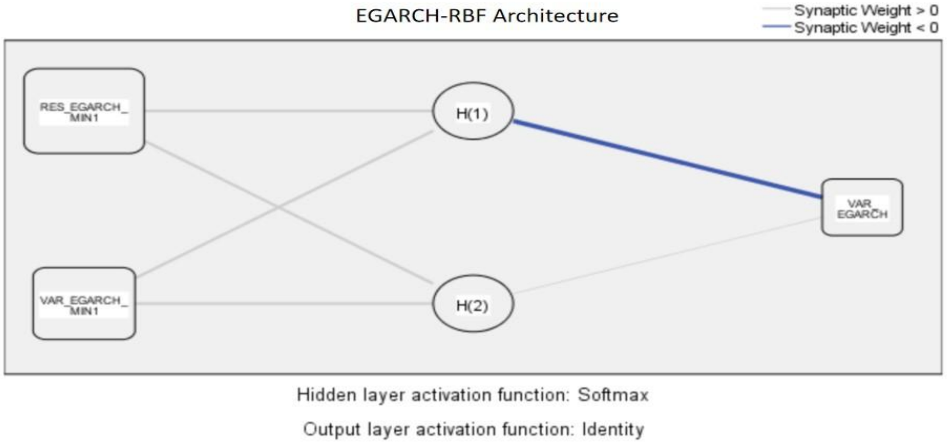

Figure 15. EGARCH-RBF Architecture for a residual model of EGARCH(1,1).

Table 11. Parameter estimation of EGARCH-RBF.

Predictor		Predicted		
		Hidden-Layer		Output-Layer
		H(1)	H(2)	VAR_EGARCH
Input-	RES_EGARCH_MIN1	-0.0681	2.1896	
Layer	VAR_EGARCH_MIN1	-0.0347	1.0824	
Hidden-Unit-Width		0.8425	4.1752	
Hidden-	H(1)			-1.1420
Layer	H(2)			0.7054

In the EGARCH-RBF model, the network initialization is based on 70% training data and 30% testing data, where the weight selection is done randomly. The training process is stopped when the residual (error) reaches 0.0001 between the output of the artificial neural networks and the desired target output. Table 11 presents the results of the parameter estimation of the EGARCH-RBF hybrid model, where each parameter in the table describes the contribution of each predictor variable to neurons in the hidden layer and the output layer. The resulting parameters indicate that the EGARCH-RBF hybrid model successfully recognized new data well through the model built during the training process.

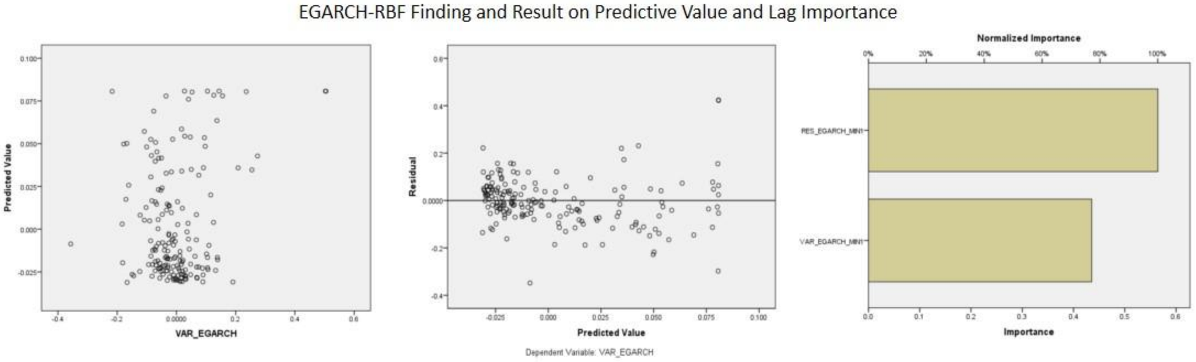

Figure 16. Predicted value and lag importance on EGARCH-RBF model.

Based on Figure 16, the left part shows the relationship between variance and predicted value of the hybrid EGARCH-RBF model. The data distribution appears concentrated around zero, with some values spreading further upward randomly, while in Figure 16, the center part shows the relationship between residual values and predicted values, where residuals are also randomly spread around the more centralized and elongated zero line. This shows that the hybrid EGARCH-RBF model no longer shows a certain systematic pattern in its errors, indicating that the model works quite well. In Figure 16, the right part, in the form of a bar graph, shows the residual variable as a more dominant factor than the variance in influencing the prediction results. This provides information that the volatility model for ICP data with the hybrid EGARCH-RBF approach is able to provide good and more stable predictions.

In the fifth stage, a hybrid TGARCH-MLP model will be built, namely taking the residual from the TGARCH(1,1) volatility model into the MLP neural networks model. The hybrid TGARCH-MLP model, as in Figure 17, is optimally built from two input units, three units in the hidden layer, and one output unit. The neural networks built in the hybrid TGARCH-MLP model use the backpropagation method with a sigmoid activation function between layers.

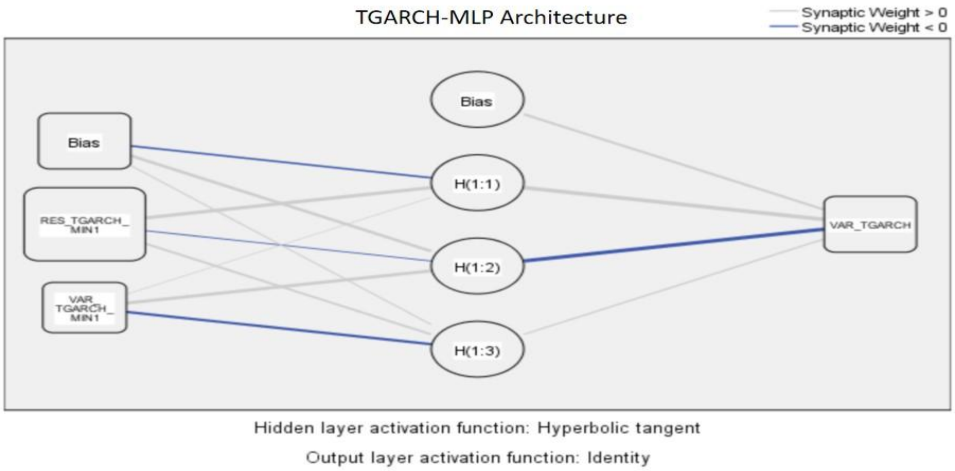

Figure 17. TGARCH-MLP Architecture for a residual model of TGARCH(1,1).

Table 12. Parameter estimation of TGARCH-MLP.

Predictor		Predicted			
		Hidden-Layer-1			Output-Layer
		H(1:1)	H(1:2)	H(1:3)	VAR_TGARCH
Input-Layer	(Bias)	-0.2831	0.4821	0.0953	
	RES_TGARCH_MIN1	0.5547	-0.0831	0.3892	
	VAR_TGARCH_MIN1	0.0751	0.4902	-0.4481	
Hidden-Layer-1	(Bias)				0.4349
	H(1:1)				1.0048
	H(1:2)				-0.5961
	H(1:3)				0.1528

In the TGARCH-MLP model, the network initialization is based on 70% training data and 30% testing data, where the weight selection is done randomly. The training process is stopped when the residual (error) reaches 0.0001 between the output of the artificial neural networks and the desired target output. Table 12 presents the results of the parameter estimation of the hybrid TGARCH-MLP model, where each parameter in the table describes the contribution of each predictor variable to the neurons in the hidden layer and the output layer. The resulting parameters indicate that the hybrid TGARCH-MLP model successfully recognized new data well through the model built during the training process.

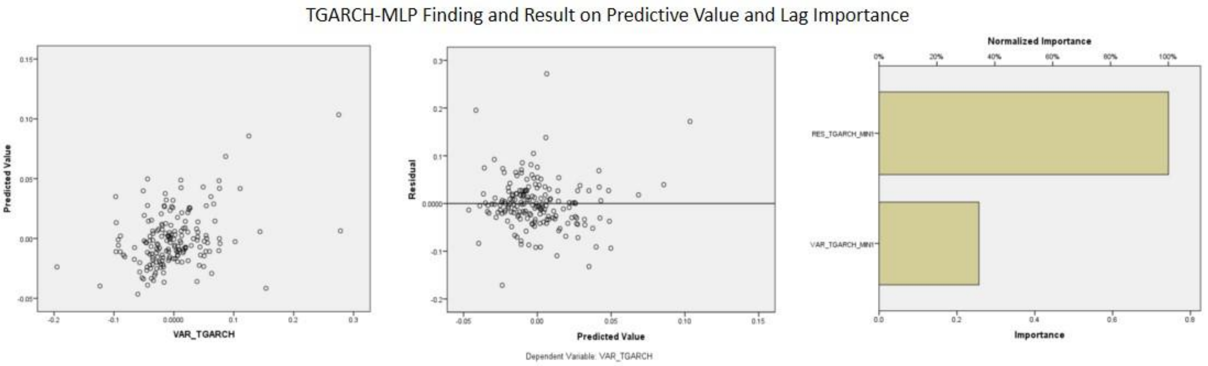

Figure 18. Predicted value and lag importance on TGARCH-MLP model.

Based on Figure 18, the left part shows the relationship between variance and predicted value of the hybrid TGARCH-MLP model. The data distribution appears concentrated around zero, with some values spreading further randomly, while in Figure 18, the center part shows the relationship between residual values and predicted values, where the residuals are also randomly spread around the more centered zero line. This shows that the hybrid TGARCH-MLP model no longer shows a certain systematic pattern in its errors, indicating that the model works quite well. In Figure 18, the right part, in the form of a bar graph, shows the residual variable as a very dominant factor than the variance in influencing the prediction results. This provides information that the volatility model for ICP data with the hybrid TGARCH-MLP approach is able to provide good and more stable predictions.

In the sixth stage, a hybrid TGARCH-RBF model will be built, namely taking the residual from the TGARCH(1,1) volatility model into the RBF neural networks model. The hybrid TGARCH-RBF model, as in Figure 19, is optimally built from two input units, three units in the hidden layer, and one output unit. The neural networks built on the hybrid TGARCH-RBF model use the backpropagation method with a sigmoid activation function between layers.

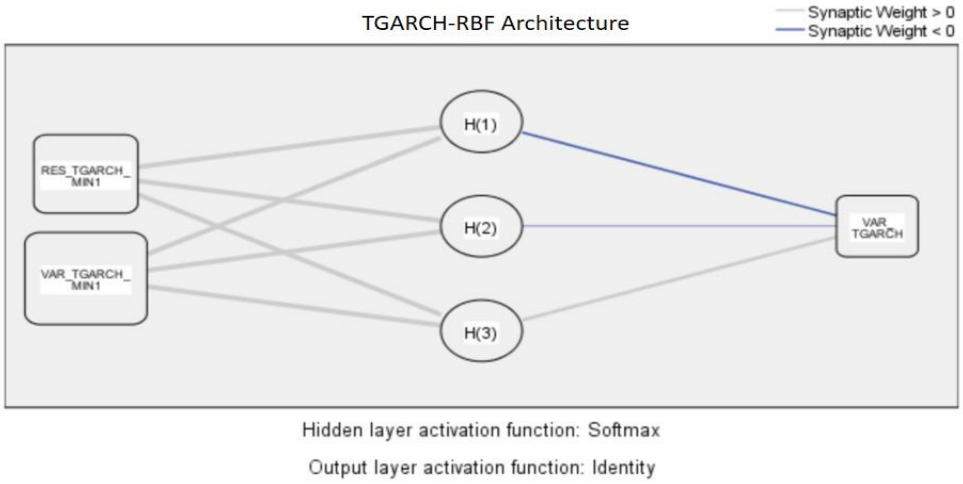

Figure 19. TGARCH-RBF Architecture for a residual model of TGARCH(1,1).

Table 13. Parameter estimation of TGARCH-RBF.

Predictor		Predicted			
		Hidden-Layer			Output-Layer
		H(1)	H(2)	H(3)	VAR_TGARCH
Input-Layer	RES_TGARCH_MIN1	-0.1723	0.0465	2.0932	
	VAR_TGARCH_MIN1	-0.5281	0.8263	-2.1472	
Hidden-Unit-Width		0.7861	0.7352	2.6015	
Hidden-Layer	H(1)				-0.5602
	H(2)				-0.5323
	H(3)				0.7269

In the TGARCH-RBF model, the network initialization is based on 70% training data and 30% testing data where the weight selection is done randomly. The training process is stopped when the residual (error) reaches 0.0001 between the output of the artificial neural network and the desired target output. Table 13 presents the results of the parameter estimation of the hybrid TGARCH-RBF model where each parameter in the table describes the contribution of each predictor variable to the neurons in the hidden layer and the output layer. The resulting parameters indicate that the hybrid TGARCH-RBF model has successfully recognized new data well through the model built during the training process.

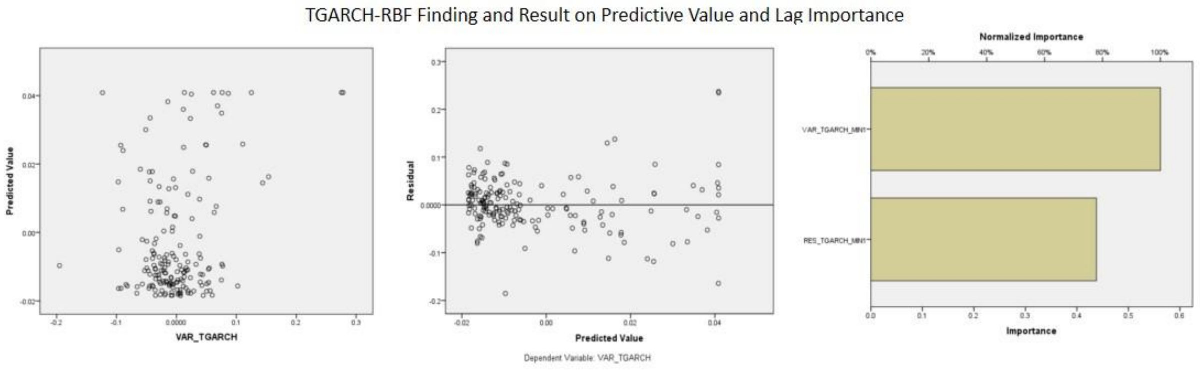

Figure 20. Predicted value and lag importance on TGARCH-RBF model.

Based on Figure 20, the left part shows the relationship between variance and predicted value of the hybrid TGARCH-RBF model. The data distribution appears concentrated around zero, with some values spreading further upward randomly, while in Figure 20, the center part shows the relationship between residual values and predicted values, where residuals are also randomly spread around the more centered zero line. This shows that the hybrid TGARCH-RBF model no longer shows a specific systematic pattern in its errors, indicating that the model works quite well. In Figure 20, the right part, as a bar graph, shows the residual variable as a more dominant factor than the variance influencing the prediction results. This provides information that the volatility model for ICP data with the hybrid TGARCH-RBF approach is able to provide good and more stable predictions.

3.7 Model Evaluation

The volatility model was built using the GARCH variance model, and volatility can be adjusted with a neuroinformatics approach based on the MLP and RBF neural network models. Then, this asymmetric model is developed using the EGARCH and TGARCH volatility models. In an effort to improve the accuracy and stability of the model, a hybrid volatility model was developed. The volatility model built can be detected using the AIC and BIC criteria, while the accuracy of the model is determined through the MAE, RMSE, and MAPE criteria.

TABLE 14. Best selection model and accuration for volatility model.

Model	Best Selection Model		Model Accuration		
	AIC	BIC	MAE	RMSE	MAPE (%)
GARCH(1,1)	-3.0463	-2.9903	123.3412	103.8612	7.6451
EGARCH(1,1)	-4.3219	-3.8719	92.2208	77.0808	5.1434
TGARCH(1,1)	-4.2257	-3.6094	103.1444	85.0944	5.9912
MLP	-5.8241	-4.9241	57.7556	39.9856	2.2588
RBF	-5.6324	-4.1587	67.0416	47.2616	2.7118
GARCH(1,1)-MLP	-6.6653	-6.0531	60.0832	41.2332	2.1361
GARCH(1,1)-RBF	-6.1763	-5.9823	61.1832	42.0332	2.2361
EGARCH(1,1)-MLP	-9.5761	-8.6765	57.2344	37.4544	1.8612
EGARCH(1,1)-RBF	-9.4877	-8.5651	57.3532	38.8932	1.9711
TGARCH(1,1)-MLP	-9.0785	-8.4762	58.8864	37.7064	2.0322
TGARCH(1,1)-RBF	-8.7561	8.1641	56.3832	38.4332	2.1361

In Table 14, it can be seen through the best selection criteria that the hybrid models have smaller errors than the standard models with AIC and BIC criteria. Meanwhile, the accuracy criteria also show this where the hybrid model has better accuracy; in addition, all models have MAPE values less than 10% , indicating that all models have very good accuracy and can be used as a guideline in forming a volatility model. In addition, based on the model plot and actual data at Figure 21 also shows very good accuracy for all models; all models are close to the actual value, although the performance of the hybrid model still shows relatively better accuracy than other models.

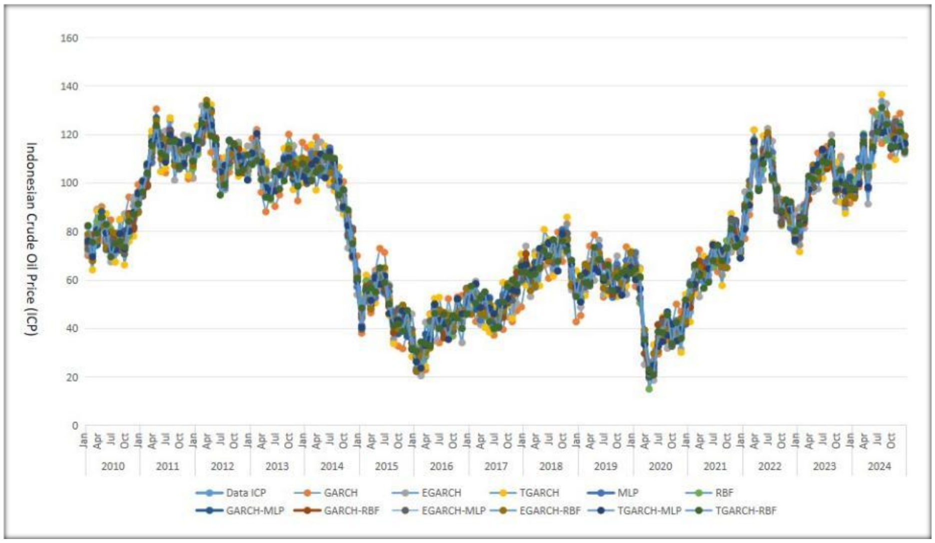

Figure 21. Plot model volatility and actual data.

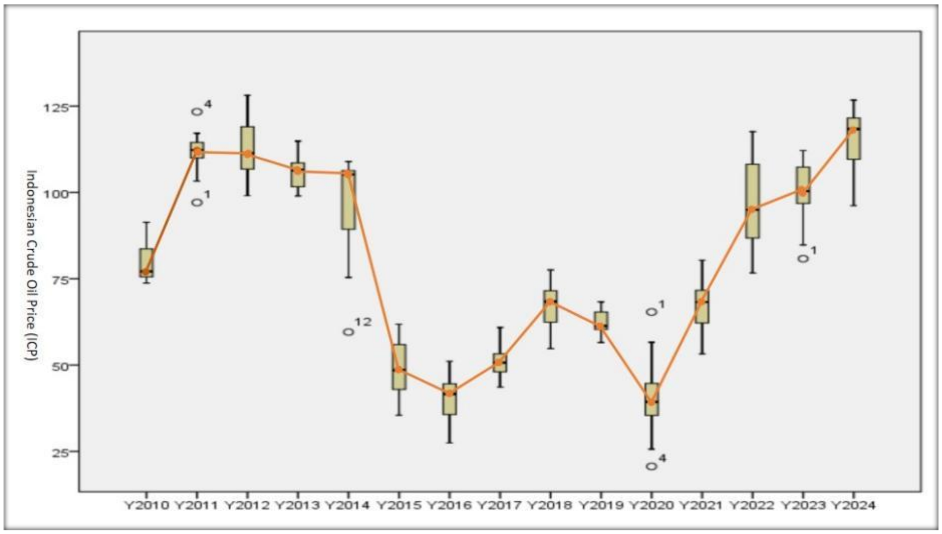

Figure 22. Boxplot ICP annually.

The ICP volatility model shows asymmetric effects that can be captured by the variance model, the MLP and RBF neural networks model, and the hybrid model. The model can show good stability and accuracy, as shown in Figure 22, where the boxplot of actual data per year shows good stability with very few outliers. Based on the volatility analysis with asymmetric variance effects, the investment strategies that can be applied are diversification, hedging, stop-lost, scalping, and active investment. Diversification strategies can help reduce the risk of asymmetric volatility in various types of assets, such as stocks, bonds, and commodities such as ICP. Hedging strategies are used for investments that involve buying or selling assets to reduce risk, such as the strategy of buying call or put options to reduce risk. Stop-loss strategies involve selling assets if the price falls below a certain level, which is helpful in reducing the risk of asymmetric volatility. Scalping strategies involve buying and selling assets in a short period of time to take advantage of price movements in an effort to take advantage of asymmetric volatility. Active investment strategies involve continuously monitoring and adjusting investment portfolios to help deal with asymmetric volatility more effectively.

Investment strategies to reduce the risk of asymmetric volatility in ICP price movements can also be done through option instruments that give the right to buy or sell assets at a certain price. Futures instruments that involve buying or selling assets at a certain price in the future. Hedge fund instruments that involve professional fund management to generate profits. Exchange-traded fund instruments that involve buying and selling assets on the stock exchange. The exchange-traded fund can be used to deal with asymmetric volatility. Risk management can also be used to identify and measure asymmetric volatility risk. Furthermore, asymmetric volatility risk investments are carried out with portfolio supervision and emotional management in effective investment decisions.

4. Conclusion

An indicator of investment risk is volatility, which measures of changes in security prices. The variance model of GARCH with heteroscedasticity effects can be used to identify volatility. The asymmetric volatility can be captured by using the adjusted exponential and threshold variance models such as EGARCH and TGARCH. While this variance model can overcome heteroscedasticity and asymmetric effects, it must maintain model stability and accuracy levels. For this reason, the hybrid model with a neuroinformatics approach through neural networks MLP and RBF was proposed. The results of this study, which employs ICP stock movements, demonstrate that while the EGARCH and TGARCH models can still overcome asymmetric volatility, the variance model using the hybrid neural networks MLP and RBF is able to deliver stability and superior accuracy. The hybrid model performance can be a guideline for making investment decisions about ICPs with asymmetric volatility structures and unpredictable price changes; the hybrid neural networks technique can offer more accurate information on variance models that can be utilized as an alternative strategy in risk management investment. Furthermore, asymmetric volatility risk investments are carried out with portfolio supervision and emotional management as guidelines in effective investment decisions based on a hybrid model with neural networks.

Acknowledgment

The authors sincerely appreciate the reviewers' valuable comments and suggestions, which helped us improve the quality of the manuscript. The authors also thank Universitas Andalas for funding this research under the scheme of indexed publication research with contract number T/55/UN.16.17/PT.01.03/IS-RPT/2022.

Conflict of Interests

The authors declare that there is no conflict of interests.

References

[1] W. Mensi, R. Gök, E. Gemici, S.H. Kang, Tail Risk Contagion and Connectedness Between Crude Oil, Natural Gas, Heating Oil, Precious Metals, and International Stock Markets, Int. Econ. 181 (2025), 100570. https://doi.org/10.1016/j.inteco.2024.100570.

[2] N. Haukvik, H. Cheraghali, P. Molnár, The Role of Investors' Fear in Crude Oil Volatility Forecasting, Res. Int. Bus. Financ. 70 (2024), 102353. https://doi.org/10.1016/j.ribaf.2024.102353.

[3] Z. Dai, H. Zhu, Time-varying Spillover Effects and Investment Strategies Between Wti Crude Oil, Natural Gas and Chinese Stock Markets Related to Belt and Road Initiative, Energy Econ. 108 (2022), 105883. https://doi.org/10.1016/j.eneco.2022.105883.

[4] C. Behera, B.N. Rath, The Interconnectedness Between Crude Oil Prices and Stock Returns in G20 Countries, Resour. Polic. 91 (2024), 104950. https://doi.org/10.1016/j.resourpol.2024.104950.

[5] S. Liu, W. Fang, X. Gao, F. An, M. Jiang, Y. Li, Long-term Memory Dynamics of Crude Oil Price Spread in Non-dollar Countries Under the Influence of Exchange Rates, Energy 182 (2019), 753-764. https://doi.org/10.1016/j.energy.2019.06.072.

[6] S.S. Sharma, D.H.B. Phan, B. Iyke, Do Oil Prices Predict Indonesian Macroeconomy?, Econ. Model. 82 (2019), 2-12. https://doi.org/10.1016/j.econmod.2019.08.008.

[7] Ermanely, D. Devianto, F. Yanuar, R. Hidayat, Analyzing Composite Stock Price Index Volatility in Response to Changes in Data Structure Using Bayesian Markov-Switching GARCH, Int. J. Adv. Sci. Eng. Inf. Technol. 14 (2024), 1209-1215. https://doi.org/10.18517/ijaseit.14.4.19776.

[8] O.V. De la Torre-Torres, E. Galeana-Figueroa, J. Álvarez-García, A Test of Using Markov-switching Garch Models in Oil and Natural Gas Trading, Energies 13 (2019), 129. https://doi.org/10.3390/en13010129.

[9] M. Shakeel, M.R. Rabbani, I.T. Hawaldar, V. Chhabra, F.K. Zaidi, Is There an Intraday Volatility Spillover Between Exchange Rate, Gold and Crude Oil?, J. Open Innov. Technol. Mark. Complex. 9 (2023), 100094. https://doi.org/10.1016/j.joitmc.2023.100094.

[10] D. Devianto, On the Class of Infinitely Divisible Exponential Distribution, J. Phys. Conf. Ser. 1097 (2018), 012084. https://doi.org/10.1088/1742-6596/1097/1/012084.

[11] D. Devianto, M. Maiyastri, K. Azmi, On the Levy Measure Property of Gamma Distribution, AIP Conf. Proc. 2242 (2020), 030003. https://doi.org/10.1063/5.0007842.

[12] X. Ni, Z. Wang, A. Akbar, S. Ali, Measuring Natural Resources Rents Volatility: Evidence from Egarch and Tgarch for Global Data, Resour. Polic. 76 (2022), 102553. https://doi.org/10.1016/j.resourpol.2022.102553.

[13] S. Aras, Stacking Hybrid Garch Models for Forecasting Bitcoin Volatility, Expert Syst. Appl. 174 (2021), 114747. https://doi.org/10.1016/j.eswa.2021.114747.

[14] Y. Luan, S. Ye, Y. Li, L. Jia, X. Yue, Revisiting Natural Resources Volatility Via Tgarch and Egarch, Resour. Polic. 78 (2022), 102896. https://doi.org/10.1016/j.resourpol.2022.102896.

[15] H. Dai, G. Huang, H. Zeng, F. Zhou, Pm2.5 Volatility Prediction by Xgboost-mlp Based on Garch Models, J. Clean. Prod. 356 (2022), 131898. https://doi.org/10.1016/j.jclepro.2022.131898.

[16] C. Ciner, Forecasting the Aggregate Market Volatility by Boosted Neural Networks, Financ. Res. Lett. 72 (2025), 106505. https://doi.org/10.1016/j.frl.2024.106505.

[17] A. Hashemi Fath, F. Madanifar, M. Abbasi, Implementation of Multilayer Perceptron (Mlp) and Radial Basis Function (Rbf) Neural Networks to Predict Solution Gas-oil Ratio of Crude Oil Systems, Petroleum 6 (2020), 80-91. https://doi.org/10.1016/j.petlm.2018.12.002.

[18] W. Kristjanpoller R, E. Hernández P, Volatility of Main Metals Forecasted by a Hybrid Ann-garch Model with Regressors, Expert Syst. Appl. 84 (2017), 290-300. https://doi.org/10.1016/j.eswa.2017.05.024.

[19] D. Devianto, M. Yollanda, M. Maiyastri, F. Yanuar, The Soft Computing Ffnn Method for Adjusting Heteroscedasticity on the Time Series Model of Currency Exchange Rate, Front. Appl. Math. Stat. 9 (2023), 1045218. https://doi.org/10.3389/fams.2023.1045218.

[20] E. Arif, E. Herlinawati, D. Devianto, M. Yollanda, D. Permana, Hybridization of Long Short-term Memory Neural Network in Fractional Time Series Modeling of Inflation, Front. Big Data 6 (2024), 1282541. https://doi.org/10.3389/fdata.2023.1282541.

[21] D. Widiarsih, W.D. Taifur, E. Ridwan, D. Devianto. Inflation connectedness network in Indonesia, Int. J. Econ. Financ. Stud. 15 (2023), 391-417.

[22] E. Arif, D. Devianto, M. Yollanda, A. Afrimayani, Analysis of Precious Metal Price Movements Using Long Memory Model and Fuzzy Time Series Markov Chain, Int. J. Energy Econ. Polic. 12 (2022), 202-214. https://doi.org/10.32479/ijeep.13531.

[23] E. Arif, E. Herlinawati, D. Devianto, M. Yollanda, A. Afrimayani, The Performance of Long Memory Fractional Series Price Model of Essential Trace Element Zinc, Commun. Math. Biol. Neurosci. 2024 (2024), 88. https://doi.org/10.28919/cmbn/8686.

[24] D. Devianto, E. Wahyuni, M. Maiyastri, M. Yollanda, The Seasonal Model of Chili Price Movement with the Effect of Long Memory and Exogenous Variables for Improving Time Series Model Accuracy, Front. Appl. Math. Stat. 10 (2024), 1408381. https://doi.org/10.3389/fams.2024.1408381.

[25] M. Bonato, O. Cepni, R. Gupta, C. Pierdzioch, Financial Stress and Realized Volatility: the Case of Agricultural Commodities, Res. Int. Bus. Financ. 71 (2024), 102442. https://doi.org/10.1016/j.ribaf.2024.102442.

[26] G. Avinash, V. Ramasubramanian, M. Ray, et al. Hidden Markov Guided Deep Learning Models for Forecasting Highly Volatile Agricultural Commodity Prices, Appl. Soft Comput. 158 (2024), 111557. https://doi.org/10.1016/j.asoc.2024.111557.

[27] D. Devianto, M. Yollanda, S. Maryati, et al. The Bayesian Vector Autoregressive Model as an Analysis of the Government Expenditure Shocks While the Covid-19 Pandemic to Macroeconomic Factors, J. Open Innov.: Technol. Mark. Complex. 9 (2023), 100156. https://doi.org/10.1016/j.joitmc.2023.100156.

[28] Y. Xu, B. Guan, W. Lu, S. Heravi, Macroeconomic Shocks and Volatility Spillovers Between Stock, Bond, Gold and Crude Oil Markets, Energy Econ. 136 (2024), 107750. https://doi.org/10.1016/j.eneco.2024.107750.