

Exchange Rate Prediction of Rupiah to US Dollar Based on JISDOR Using Support Vector Regression

Elly Pusporani^{1*}, Nabila Nurdin¹, Anastasya Veronika Sitorus¹

¹Department of Mathematics, Universitas Airlangga, Indonesia

elly.pusporani@fst.unair.ac.id

ABSTRACT

Article History:

Received : 12-03-2026

Revised : 25-05-2026

Accepted : 05-06-2026

Online : 23-10-2026

Keywords:

Support Vector

Regression;

Artificial Neural

Network;

JISDOR;

Exchange Rate

Prediction;

Machine Learning.



The exchange rate between the Indonesian Rupiah and the United States Dollar (USD) is a critical macroeconomic indicator that shapes international trade, foreign investment, and monetary policy. However, machine learning-based forecasting studies that specifically target the Jakarta Interbank Spot Dollar Rate (JISDOR) as the primary output variable remain limited, leaving a notable gap in the Indonesian financial forecasting literature. This study employs a quantitative time-series approach to address that gap by applying Support Vector Regression (SVR) with a two-stage grid search hyperparameter optimization to JISDOR data spanning January 2023 to January 2025, and benchmarks its performance against an Artificial Neural Network (ANN). Nonlinearity in the data was confirmed through the Teräsvirta test, which justified the use of kernel-based nonlinear models. The optimized SVR with a Radial Basis Function (RBF) kernel achieved an RMSE of 43.49 and a MAPE of 0.22% on the test set, outperforming the best ANN architecture which recorded an RMSE of 49.72 and a MAPE of 0.25%. Neither model showed signs of overfitting, as the errors on the test set were consistently lower than those on the training set. These findings indicate that optimized SVR serves as a reliable and practically applicable forecasting tool for the Rupiah to USD exchange rate. For policymakers, the model offers a quantitative basis for determining the appropriate timing of market interventions, while for financial market participants, it supports more precise hedging decisions and exchange rate risk management.



Crossref

<https://doi.org/10.31764/jtam.v10i4.38787>



This is an open access article under the **CC-BY-SA** license

A. INTRODUCTION

The rupiah exchange rate against the United States dollar (USD) is a key indicator of Indonesia's macroeconomic stability, influencing international trade, foreign investment, and monetary policy. Its fluctuations directly affect price stability and inflation, serve as a key consideration in the formulation of monetary policy by Bank Indonesia, and determine the level of risk and profitability for businesses and investors (Amri et al., 2025). Volatile exchange rate movements pose real risks to exporters, importers, and financial market participants, making the ability to accurately predict exchange rate movements of significant practical value to policymakers and economic actors. Contemporary studies indicate that the foreign exchange market is complex, non-linear, and influenced by a combination of fundamental factors and market sentiment, making classical prediction models sometimes incapable of capturing these dynamics (Junior et al., 2023). This limitation has driven researchers to explore alternative approaches, as demonstrated by studies using JISDOR-based data that reveal time-varying

volatility in the Rupiah–USD exchange rate that linear models fail to adequately represent (Ahmar et al., 2024).

Bank Indonesia created the transparent reference rate, known as JISDOR, in an effort to address the needs of market participants. An exchange rate represents the value of a nation's currency in relation to another, indicating how much foreign currency can be obtained for a single unit of domestic currency. For instance, in December 2024, Bank Indonesia recorded that one US dollar was equivalent to Rp 16,162, illustrating the Rupiah's exchange value at that time (Johnson, 2024; Marwadi & Rokhim, 2016). However, literature specifically using JISDOR as a target variable in forecasting studies using a machine learning approach is still relatively limited (Ahmar et al., 2024).

The development of machine learning methods in financial forecasting has introduced various non-linear algorithms, one of which is Support Vector Regression (SVR), designed to address regression problems in non-linear data through the kernel trick and the optimal margin principle. SVR has been applied to various financial problems, such as stock prices, commodities, and exchange rates, with results often more accurate than classical parameter models (Zhang et al., 2022). Applications specifically to the Rupiah–USD pair have confirmed that SVR with multiple kernel functions — linear, polynomial, and RBF — achieves high prediction accuracy, with the RBF kernel demonstrating particularly strong performance in capturing non-linear exchange rate dynamics (Alida & Mustikasari, 2020). Reported that SVR can provide competitive prediction performance in the foreign exchange market when compared to conventional time series models. Furthermore, recent research also shows a trend of combining SVR with hybrid approaches to improve performance on volatile financial data (Alamsyah et al., 2024). This trend is supported by evidence that SVR integrated with metaheuristic optimization for hyperparameter tuning consistently outperforms standalone and conventional forecasting models across multiple currency pairs when evaluated using MAPE and RMSE (Hu et al., 2021).

The literature review reveals several important findings but also clear research gaps. First, while comparative studies exist between SVR and other models, such as ARIMA or LSTM, many of these studies use international data samples or exchange rate data other than JISDOR; few focus on the Indonesian domestic context, targeting JISDOR (Ahmar et al., 2024; Ningrum et al., 2025). Second, several studies note that SVR performance is sensitive to kernel selection and C/ϵ parameters, often requiring optimization or hybrid approaches to achieve optimal results. However, the application of optimization techniques to the JISDOR dataset is underreported (Jazayeri, 2024). Third, several recent studies have also shown that combining SVR with models that address heteroscedasticity (volatility clustering) can improve prediction quality in highly volatile exchange rates, but the application of SVR-based heteroscedastic models to Rupiah data remains rarely explored (Alamsyah et al., 2024).

Based on these identified gaps, this study is designed to address several important issues by utilizing the latest JISDOR data as the primary prediction target to enhance policy relevance, applying Support Vector Regression with a hyperparameter optimization procedure and comparing its predictive performance with Artificial Neural Network to evaluate its advantages in modeling the Rupiah–USD exchange rate, and further exploring modeling strategies that incorporate heteroscedasticity characteristics inherent in the foreign exchange market. The

heteroscedasticity effect, characterized by non-constant variance in residuals over time, is a well-documented challenge in financial time series modeling that standard ARIMA-based models alone are insufficient to resolve, requiring the integration of more flexible computational approaches (Devianto et al., 2023). The originality of this research lies in its combined approach of using up-to-date JISDOR data, implementing an optimized SVR specifically for the Rupiah context, and examining heteroscedasticity aspects that have received limited attention in previous studies (Zhang et al., 2022; Mubarak et al., 2024).

According to this framework, the study intends to use JISDOR data to analyze the historical behavior of the Rupiah–USD exchange rate, create a predictive model based on SVR with hyperparameter optimization, compare the predictive performance of the optimized SVR with ANN using out-of-sample evaluation metrics like RMSE and MAPE, and offer market participants and monetary authorities policy recommendations for managing exchange rate risk (Aji & Surjandari, 2020). The importance of hyperparameter optimization in SVR for exchange rate prediction has been demonstrated through hybrid tuning approaches, where optimized SVR models measured by RMSE and MAPE consistently outperform baseline SVR and conventional time series models (Halder et al., 2024). These objectives highlight both the theoretical contribution through testing SVR in the JISDOR context and addressing heteroscedasticity and the practical contribution through hedging strategies and monetary policy recommendations based on more accurate exchange rate forecasts (Junior et al., 2023).

In brief, this research contributes to the following: (a) adding to the Indonesian foreign exchange forecasting literature with JISDOR-based empirical data; (b) evaluating the effectiveness of optimized SVR on Rupiah exchange rate data, thereby adding comparative evidence between machine learning and classical econometric methods; and (c) providing practical recommendations for monetary authorities and business actors to reduce exposure to exchange rate volatility. The urgency of rupiah exchange rate forecasting for monetary policy planning and economic stability has been consistently affirmed in recent Indonesian literature, reinforcing the policy relevance of developing more accurate and data-driven prediction models (Pratiwi et al., 2025). These theoretical and practical contributions are expected to narrow the gap between the forecasting method literature and policy needs in the Indonesian foreign exchange market (Jazayeri, 2024; Zhang et al., 2022).

Furthermore, regarding the results and implications, if the optimized SVR demonstrates superior predictive performance compared to ANN, the implication is that monetary authorities and market participants should consider machine learning methods as complementary tools in hedging planning and strategies. Furthermore, if the heteroscedasticity model combined with the SVR is proven to improve accuracy, practical recommendations would include applying the model to project short-term risks and adjust market intervention policies (Alamsyah et al., 2024).

B. METHODS

The overall research procedure employed in this study is illustrated in Figure 1. The flowchart presents the sequential stages of data collection, preprocessing, model development, evaluation, and result analysis conducted in this research.

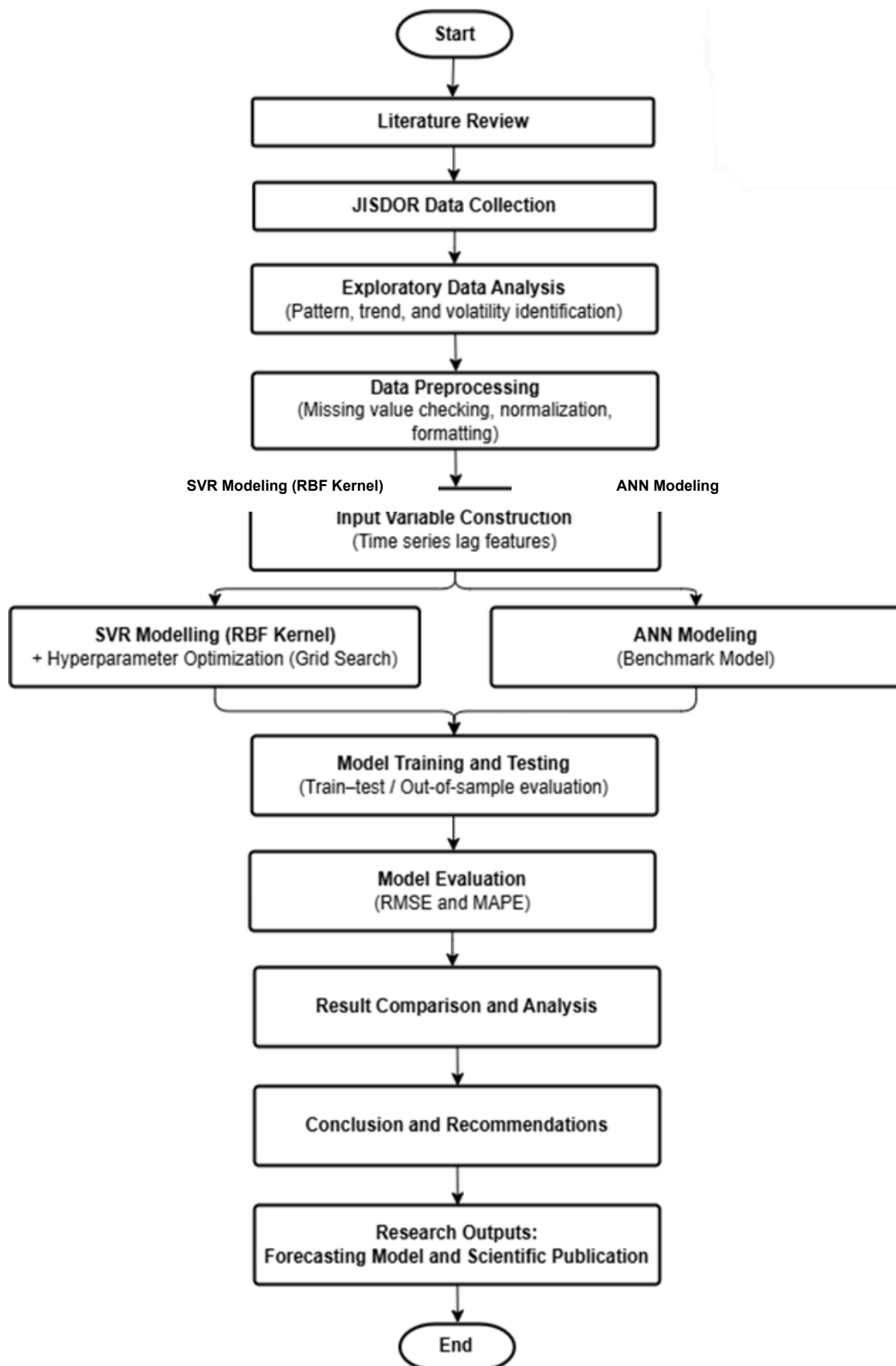


Figure 1. Research Flowchart of the Proposed Exchange Rate Forecasting Framework

1. Exploratory Data Analysis

This study employed a quantitative time-series approach using daily Jakarta Interbank Spot Dollar Rate (JISDOR) data published by Bank Indonesia from January 2023 to January 2025. This period was chosen to represent exchange rate dynamics before, during, and after the COVID-19 pandemic, allowing the analysis to reflect the volatile foreign exchange market conditions over a relatively long period (Andreas, 2025). The dataset was divided into 80% training data and 20% testing data to evaluate model generalization performance. Prior to modeling, the data preprocessing stage included missing value checking, normalization, and stationarity testing using the Augmented Dickey–Fuller (ADF) and KPSS tests. Nonlinearity testing was conducted using the Teräsvirta test to justify the use of nonlinear machine learning methods. The operational definitions and measurement scales of the variables used in this study are presented in Table 1.

Table 1. Operational Definition and Measurement Variables

Variables	Operational Definition	Variable Type	Measurement / Scale
Rupiah exchange rate against USD (JISDOR)	The value of 1 USD in Rupiah as recorded on the JISDOR on a particular day	Dependent variable	Continuous numeric (Rupiah scale)
Variable Predictor	Lagged exchange rates (JISDOR t-lag exchange rates), domestic interest rates, inflation, US interest rates, trade balance, etc. (depending on the available data we use)	Independent variables	Numerical continuous; some are transformed to log, or differentiated if not stationary.
Parameter SVR	SVR hyperparameters such as C (penalty parameter), ϵ (epsilon), kernel type (e.g. RBF)	Model variables	Numerical values/hyperparameters, selected through optimization
ANN Architecture	Network structure of Artificial Neural Network including the number of hidden layers, number of neurons per layer, and activation functions (e.g., ReLU, tanh)	Model comparison variable	Numeric and categorical values determined through architecture tuning

2. Support Vector Regression (SVR)

The Support Vector Regression (SVR) model was developed using several nonlinear kernel functions, namely Radial Basis Function (RBF), Polynomial, and Sigmoid kernels. The optimal model was selected based on the smallest Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE). Support Vector Regression (SVR) is a modification of the Support Vector Machine (SVM) used to solve regression problems by finding the best regression function that minimizes error and maximizes margin (Saputra et al., 2019).

The effectiveness of Support Vector Regression (SVR) is strongly influenced by the type of kernel function selected during model construction. The chosen kernel plays a crucial role in handling data nonlinearity, particularly in high-dimensional feature spaces (Du et al., 2024;

Wang et al., 2020). A kernel function essentially represents an inner product in a transformed feature space, expressed as $\phi(x_i) \cdot \phi(x)$, and can be formulated as:

$$K(x_i, x) = \phi(x_i) \cdot \phi(x) \quad (1)$$

This formulation allows the SVR regression function to be expressed as:

$$y_i = f(x_i) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(x_i, x) + b \quad (2)$$

where α_i and α_i^* are the Lagrange multipliers, $K(x_i, x)$ represents the kernel function, and b is the bias term. Proper tuning of kernel parameters is essential, as it directly affects the model's predictive accuracy. Table 2 provides an overview of several kernel functions that are commonly applied in SVR modelling (Galena et al., 2025).

Table 2. Common Kernel Functions in SVR

Kernel Type	Formula	Description
Linear	$K(x_i, x) = x_i^T x$	Uses the dot product directly in the input space.
Polynomial	$K(x_i, x) = (\gamma x_i^T x + r)^p$, $p = 1, 2, \dots$	Captures polynomial relationships with adjustable degree p .
Radial Basis Function (RBF)	$K(x_i, x) = \exp(-\gamma \ x_i - x\ ^2)$	Handles nonlinear patterns effectively in high-dimensional spaces.
Sigmoid	$K(x_i, x) = \tanh(\gamma x_i^T x + r)$	Uses the hyperbolic tangent function, producing outputs between -1 and 1.

In this context, $\tanh$ is a hyperbolic function that maps values between -1 and 1, γ is a kernel parameter that controls sensitivity to input differences, and T denotes the transpose operator of a matrix. The primary objective of hyperparameter tuning is to determine the most effective combination of parameters that yields the highest prediction accuracy on the test dataset. Grid search is a widely used method for this purpose, but it often requires significant computational time due to its exhaustive search process. To improve efficiency, the grid search strategy is typically executed in two stages: a loose grid stage and a finer grid stage (Purnama & Hendarsin, 2020).

In the loose grid stage, candidate hyperparameters are selected from a coarse set of integer power values to provide a broad overview of the parameter space. Once a promising region is identified, the finer grid stage focuses on searching around those candidate values in smaller intervals. This two-stage approach enables more precise tuning while reducing unnecessary computation, resulting in a more efficient model optimization process (Zhang & O'Donnell, 2020).

3. Artificial Neural Network (ANN)

The Artificial Neural Network (ANN) model was constructed using feedforward neural network architectures with one and two hidden layers. Several neuron combinations and activation functions (Logistic, ReLU, and Tanh) were evaluated to determine the best-performing architecture. Data normalization was applied before model training to improve convergence stability. The optimal ANN architecture was selected based on prediction accuracy on testing data. In general, the ANN function can be written as follows:

$$y = f\left(\sum_{i=1}^n In_i x_i + b\right) \quad (3)$$

with x_i as input, In_i weight, b bias, and $f(\cdot)$ nonlinear activation functions such as sigmoid, ReLU, or tanh.

ANN is a computational approach inspired by the structure and learning mechanisms of the human brain, consisting of interconnected neurons capable of modeling nonlinear relationships between variables (Abdolrasol et al., 2021), (Pusporani et al., 2022). The architecture of an ANN model is generally determined by three main components, namely the network structure, learning algorithm, and activation function. Various activation functions that can be used to generate the output are presented in Table 3.

Table 3. Activation Functions in ANN

Activation Function	Equation	Description
Linear / Identity	$f(x) = x$	The output is directly proportional to the input, commonly used for regression tasks.
Hyperbolic Tangent (Tanh)	$f(x) = \tanh(x)$	Maps input values into the range of -1 to 1, often used in hidden layers to capture nonlinearity.
Sigmoid	$f(x) = 1 / (1 + \exp(-x))$	Maps input values into the range of 0 to 1, useful for probabilistic outputs.

The architecture of an ANN model is determined by three key components: the network structure, the learning algorithm, and the activation function. Some of the various kinds of activation functions that can be used to get the output can be seen in Table 3. Backpropagation (BP), a popular learning technique in Artificial Neural Networks (ANN), effectively computes gradients by using the chain rule. In order to generate outputs, input signals go through the network in the feedforward phase of BP. In the backward phase, mistakes are calculated and propagated back to modify the network weights. ANN can reduce prediction errors and enhance performance over iterations because to this process (Farizawani et al., 2020), (Abdolrasol et al., 2021).

One of the most common network architectures is Feedforward Neural Network (FFNN), which is composed of an input layer, one or more hidden layers, and an output layer. The output of the FFNN can be expressed as:

$$y_t = f^0 \left\{ v_0 + \sum_{k=1}^K \left(v_k f^h \left(\sum_{i=1}^I w_{ik} y_{(t-p)} \right) \right) \right\} \quad (4)$$

where v_k represents the weight of the k^{th} neuron in the hidden layer, w_{ik} denotes the weight of the i^{th} input connected to the k^{th} hidden neuron, and $f^h(\cdot)$ and $f^0(\cdot)$ are the activation functions for the hidden and output layers, respectively (Pusporani et al., 2019).

In addition to FFNN, deeper architectures such as Deep Learning Neural Network (DLNN) use multiple hidden layers to improve representation learning and handle more complex patterns. ANN as a field originated in the 1940s, with the first practical implementation in the late 1950s through the perceptron network. It gained further momentum in 1988 when the BP algorithm was developed, addressing early criticisms from Minsky and Papert. Today, ANN is recognized as a powerful nonlinear estimation tool capable of learning complex functional relationships from data.

4. Model Comparison and Evaluation

The model was then validated using test data with evaluation metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE). This evaluation method aligns with previous research that emphasizes the importance of comparing accuracy using more than one error measure for more robust results (Andreas, 2025). The formula for calculating MAPE is as follows:

$$MAPE = \frac{\sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right|}{n} \times 100\% \quad (5)$$

where y_t is the actual value at time t , and $\hat{y}_t$ represents the predicted value at the same time. Based on the MAPE analysis, the interpretation of forecasting accuracy levels is as follows Table 4.

Table 4. Criteria MAPE

MAPE Value	Criteria
$MAPE \leq 10\%$	Very Accurate
$10 < MAPE \leq 20\%$	Accurate
$20 < MAPE \leq 50\%$	Less Accurate
$MAPE > 50\%$	Not Accurate

Furthermore, while the Root Mean Squared Error (RMSE) gauges the average magnitude of prediction mistakes, the Mean Squared Error (MSE) shows the average of the squared disparities between the predicted and actual values. Lower MSE and RMSE values indicate better model performance and higher prediction accuracy. The formula for calculating MSE and RMSE is as follows:

$$MSE = \frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2 \quad (6)$$

$$RMSE = \sqrt{MSE}$$

The methodology used in this study is expected to address the literature gap regarding JISDOR exchange rate prediction using a machine learning approach. Most previous studies have been limited to the use of ARIMA or ANN models, while research on the application of SVR to the JISDOR exchange rate is still relatively rare in the Indonesian literature (Mubarok et al., 2024; Sari et al., 2025). Therefore, this study contributes to broadening the understanding of the effectiveness of SVR while comparing it with traditional approaches. This study can provide theoretical benefits for the development of exchange rate forecasting methods and practical benefits for policymakers and financial market players in Indonesia.

C. RESULT AND DISCUSSION

1. Exploratory Data Analysis

Data exploration was carried out to obtain a plot of the JISDOR exchange rate data for the period January 2023 to January 2025, which is shown in Figure 2.

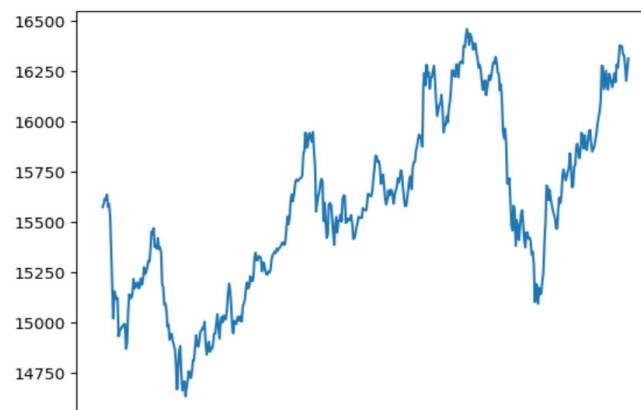


Figure 2. Plot of JISDOR Exchange Rate Data for the Period January 2023 to January 2025

Based on Figure 2, the JISDOR exchange rate data from January 2023 to January 2025 shows relatively stable fluctuations with short-term upward and downward trends. This pattern indicates the presence of nonlinear characteristics in the data. To confirm this, the Teräsvirta test was conducted, as shown in Table 5.

Table 5. Terasvirta Test Results

Data	Test Statistics	P-Value
JISDOR	F = 11,021	0,00002078

The Teräsvirta test results show that the JISDOR data from January 2023 to January 2025 exhibits a nonlinear pattern, as indicated by a significance value (p-value) of less than 0.05. Therefore, this study employs Support Vector Regression (SVR) and Neural Network (NN)

methods, both of which are capable of capturing nonlinear and complex relationships in time series data. Next, a lag analysis is carried out, which is shown in Figure 3.

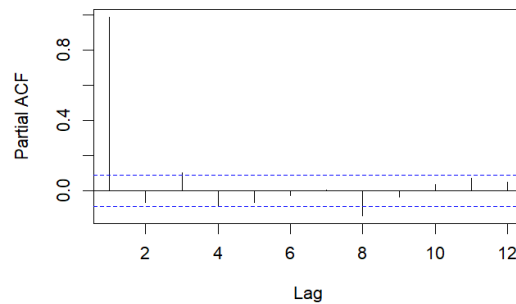


Figure 3. PACF Plot for Determining the Influential Lag

Figure 3 indicates that the first lag (y_{t-1}) is the only one with a substantial influence, indicating that the observation data from the first time period is not employed. In SVR modeling, the data division y_t for training data begins at $t = 2$ to $t = 477$, and for testing data, it begins at $t = 478$ to $t = 496$. These findings support previous studies stating that exchange rate movements are highly nonlinear and volatile, making machine learning approaches more suitable than conventional linear models for capturing complex market behavior.

2. Support Vector Regression (SVR)

Based on the results of data exploration, nonlinear kernel functions such as the Radial Basis Function (RBF), Polynomial, and Sigmoid were applied to construct the SVR model, as the data exhibited nonlinear characteristics. Next, a comparison of the SVR Kernel is carried out, which is shown in Table 6.

Table 6. Initial Modeling of SVR with Nonlinear Kernel Function

Kernel Functions	Parameter	RMSE	MAPE
Radial Basis Function (RBF)	Cost (C) = 1	59.79239	0.28 %
Polynomial	Gamma (γ) = 1	204.4671	1.09 %
Sigmoid	Epsilon (ϵ) = 0,1	9358.881	52.25 %

From the comparison results of the three kernels used in SVR, the RBF kernel provided the best performance with the smallest RMSE of 59.79 and MAPE of 0.28%, so it was selected as the optimal kernel. This result indicates that the RBF kernel is more effective in capturing nonlinear exchange rate patterns compared to Polynomial and Sigmoid kernels. Next, SVR parameter optimization is based on the Grid Search Optimization Method, as shown in Table 7.

Table 7. Loose Grid Method Parameter Value Range

Parameter	Value Range
Cost (C)	$2^{-5}; 2^{-4}; 2^{-3}; \dots; 2^5; 2^6; 2^7$
Gamma (γ)	$2^{-7}; 2^{-6}; \dots; 2^4; 2^5$
Epsilon (ϵ)	0; 0,01; ...; 0,99; 1

Table 7 presents the parameter ranges used in the loose grid stage, while the optimal results obtained are shown in Table 8.

Table 8. Optimal Parameter Values at the Loose Grid Stage

Parameter	Parameter Optimal
Cost (C)	2^5
Gamma (γ)	2^{-7}
Epsilon (ϵ)	0

The next stage is to use the finer grid stage to fine-tune the parameters after determining the best SVR model parameters in the loose grid stage. The finer grid is a further stage of the loose grid that aims to obtain optimal values. C , ϵ , and γ optimal in the vicinity of the SVR parameter values at the loose grid stage. The range of parameter values used at the finer grid stage to obtain optimal parameter values for the parameters C , ϵ , and γ are presented in Table 9.

Table 9. Parameter Value Range of Finer Grid Method

Parameter	Value Range
Cost (C)	$2^4; 2^{4,25}; \dots; 2^{5,75}; 2^6$
Gamma (γ)	$2^{-8}; 2^{-7,75}; \dots; 2^{-6,25}; 2^{-6}$
Epsilon (ϵ)	0

The range of parameter values within the ideal parameter values at the loose grid stage is displayed in Table 9. The ideal values for the finer grid stage will be established using these values. Parameter selection is carried out at the finer grid stage C and γ with parameter values ϵ based on the previous stage, namely the loose grid stage. In addition, the range of optimal parameter values, especially in selecting the value C and γ that is, the power value is within the optimum parameter range of the loose grid stage. The lowest error value is used to determine the ideal parameter selection. The grid search method's ideal parameters for the finer grid stage are shown in Table 10 as follows.

Table 10. Parameter Value Range of Finer Grid Method

Parameter	Parameter Optimal
Cost (C)	$2^{5,5}$
Gamma (γ)	2^{-8}
Epsilon (ϵ)	0

Table 10 shows the optimal SVR parameters used to determine prediction accuracy. The prediction accuracy results of the SVR model after tuning are presented in Table 11.

Table 11. SVR Model Prediction Accuracy Results After Tuning

RMSE	MAPE	Information
59.60527	0.28 %	The prediction results are very accurate

Based on Table 11, the SVR model for training data with the RBF Kernel function has a high level of accuracy in modeling the JISDOR value as shown in the graph presented in Figure 3.

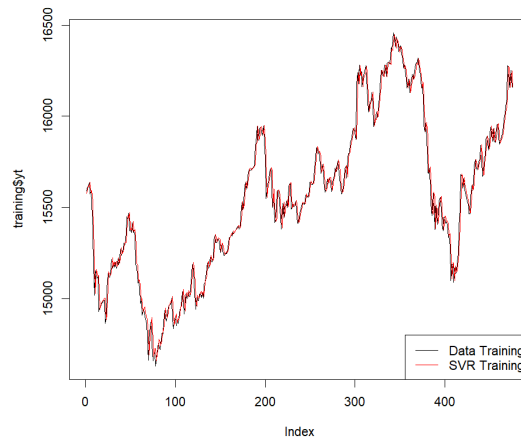


Figure 4. Plot of SVR Model Predictions After Tuning

Based on Figure 4 and Table 11, it shows that the model with the best parameters is the model after tuning with parameter values $C = 45.25483$, $\varepsilon = 0$, and $\gamma = 0.00390625$. Calculations to obtain the model with the help of the R program, obtained the value ω amounting to 118.8895 and the value b of -0.1600989. So the SVR model equation is written as follows.

$$\hat{y}_i = 118.8895 \exp\left(-0.00390625||x_i - x||^2\right) - 0.1600989$$

Next, prediction of testing data is carried out with SVR which is displayed in Table 12.

Table 12. SVR Model Prediction Accuracy Results for Testing Data

Model Type	RMSE	MAPE
Data Testing	43.48656	0.22 %

Table 12 shows that the SVR model successfully predicted the testing data with high accuracy. This is because the SVR model's testing data prediction results had a minimum RMSE value of 43.48656 and a MAPE value of 0.22%, which is categorized as less than 10%. These results indicate that the optimized SVR model has strong generalization capability and can accurately predict JISDOR exchange rate movements. The findings are consistent with previous studies by Zhang and Jazayeri, which reported that SVR performs effectively in forecasting highly volatile financial time-series data. However, this study extends previous research by specifically applying optimized SVR to the Indonesian JISDOR exchange rate context, which remains limited in the literature. This means the SVR model was able to predict the testing data with high accuracy. A visualization of the SVR model's testing data prediction results is presented in Figure 5.

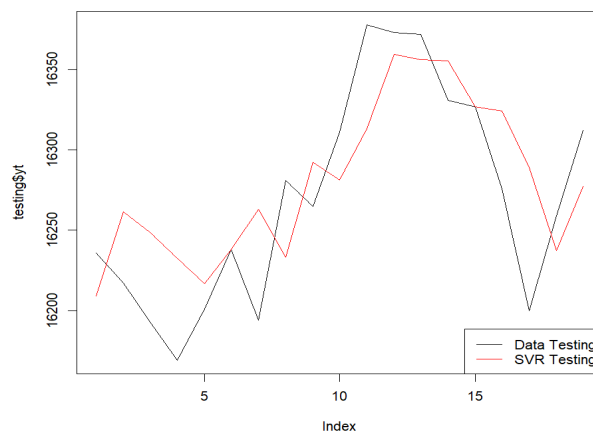


Figure 5. Plot of SVR Model Prediction Based on Testing Data

Based on Figure 5 and Table 12, the prediction results with testing data show a very high level of accuracy in the JISDOR value so that the SVR method can be used as a comparison with other methods such as ANN.

3. Artificial Neural Network (ANN)

Based on the results of data exploration, the exchange rate data exhibited a nonlinear pattern, making the Artificial Neural Network (ANN) approach suitable for capturing complex relationships within the data. Then, before being fed into the model, the data is normalized to the range $[0.1, 1]$. This ensures that the value scale is not too large, thus ensuring a more stable neural network learning process and faster convergence. Once the predictions are obtained, the results were denormalized back to their original scale so that they could be directly compared with the actual exchange rate values, as shown in Table 13.

Table 13. Data Training Transformation Results

yt	yt-1	yt_transform	yt-1_transform
15590	15572	0.57218	0.563308
15615	15590	0.584502	0.57218
15610	15615	0.582037	0.584502
15635	15610	0.594359	0.582037
15574	15635	0.564294	0.594359
15589	15574	0.571687	0.564294
15527	15589	0.541128	0.571687
15366	15527	0.461774	0.541128
15177	15366	0.36862	0.461774
15019	15177	0.290745	0.36862
⋮	⋮	⋮	⋮
16159	16270	0.852629	0.907338
16208	16159	0.876780	0.852629
16251	16208	0.897974	0.876780
16162	16251	0.854107	0.897974
16157	16162	0.851643	0.854107

Next, model was tested using various combinations of network architectures, consisting of different numbers of hidden layers, neurons in each hidden layer, and activation functions. This process aimed to determine the best configuration for achieving optimal prediction performance. The following Table 14 presents the architecture combinations used in the initial modeling stage.

Table 14. Initial Modeling of ANN

Hidden Layer	Neurons in Hidden Layer	Activation Function
1	(5), (10), (50), (100), (1000)	
2	(5,5), (5,10), (5,50), (5,100), (5,1000), (10,5), (10,10), (10,50), (10,100), (10,1000), (50,5), (50,10), (50,50), (50,100), (50,1000), (100,5), (100,10), (100,50), (100,100), (100,1000), (100,5), (100,10), (100,1000), (1000,5), (1000,10), (1000,50), (1000,100), (1000,1000)	Logistic ReLU Tanh

Based on Table 14, the tested network architectures consist of models with one and two hidden layers. These structures can be illustrated in Figure 6 and Figure 7:

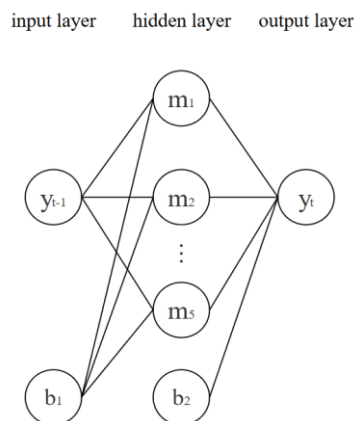


Figure 6. ANN Architecture with One Hidden Layers

Figure 6 represents one example of the neurons in hidden layer structure, specifically for a model with a single hidden layer. Based on Figure 5, the structure of an ANN consisting of three main layers: the input layer, hidden layer, and output layer. The input layer contains a single neuron representing the variable y_{t-1} . The signal from this neuron is transmitted to the hidden layer, which consists of five neurons ($m_1, m_2, \dots, m_5$) that process and extract nonlinear patterns from the data. In this figure, the neurons labeled m_1 to m_5 correspond to the number of neurons contained in the hidden layer. If the hidden layer consists of 10, 100, or more neurons, the notation will follow accordingly, becoming m_1 to m_{10} , m_1 to m_{100} , and so on. The processed information is then passed to the output layer with a single neuron (y_t) that produces the predicted value. Each layer is equipped with a bias term, namely b_1 for the hidden layer and b_2 for the output layer, which helps adjust the neuron activation to allow the model to learn data patterns more flexibly and accurately.

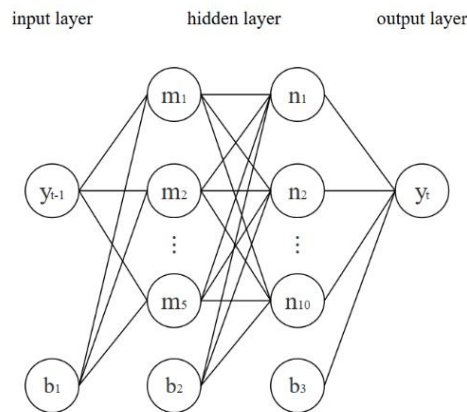


Figure 7. ANN Architecture with Two Hidden Layers

Similarly, Figure 7 shows another example of the Neurons in Hidden Layer structure, but for a model with two hidden layers. Based on Figure 6, the structure of an ANN consisting of one input layer, two hidden layers, and one output layer. The input y_{t-1} is processed through two hidden layers, each containing neurons with their respective weights and biases (b_1, b_2, b_3). The first hidden layer produces outputs from neurons m_1 to m_5 , which are then passed to the second hidden layer (n_1 to n_{10}), before generating the final output y_t in the output layer. The notation m and n follows the number of neurons in each hidden layer, where m represents the neurons in the first hidden layer and n represents the neurons in the second hidden layer. If the number of neurons in the hidden layers increases to 10, 100, or more, the notation will expand accordingly, such as m_1 to m_{10} or m_1 to m_{100} in the first layer, and n_1 to n_{10} or n_1 to n_{100} in the second layer. This structure enables the network to learn more complex nonlinear relationships between inputs and outputs.

Based on the analysis results with the help of Python software, the results obtained were that the best activation function is Tanh which is known to effectively capture complex nonlinear patterns and the best architecture is two hidden layers with 1000 and 50 neurons. This indicates that the model performs best when the network is sufficiently deep, allowing it to better capture the fluctuation patterns of the JISDOR exchange rate.

Based on the results of the neural network model testing, the RMSE value obtained for the training data was 59.545 and for the testing data was 49.717. RMSE indicates the average error of the model's predictions against the actual data in the same units as the data, namely the JISDOR exchange rate. A smaller RMSE value for the testing data compared to the training data indicates that the model has good generalization capabilities. In other words, the prediction error rate for new data (testing) is actually lower than for the training data, so the model does not experience symptoms of overfitting. Furthermore, the evaluation results with MAPE showed a value of 0.282% for the training data and 0.251% for the testing data. MAPE measures relative error as a percentage. The very small MAPE value, which is less than 1%, for both the training and testing data, proves that the neural network model has a very high level of prediction accuracy and the prediction results can be said to be very close to the actual JISDOR value. The following is a plot of the ANN model's predictions based on the test data, as shown in Figure 8.

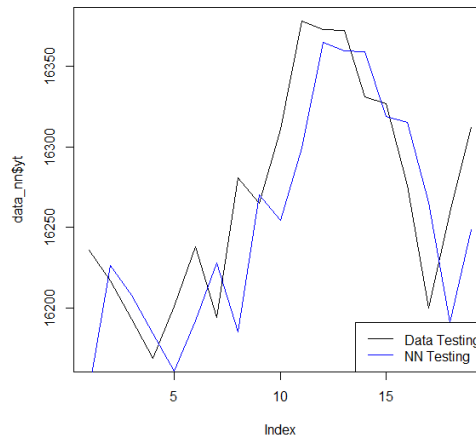


Figure 8. Plot of ANN model prediction based on testing data

Based on Figure 8, the prediction results using the testing data demonstrate a very high level of accuracy in estimating the JISDOR value. This indicates that the Artificial Neural Network (ANN) method can serve as a valid comparison to the SVR method in predicting JISDOR exchange rate fluctuations. The findings demonstrate that the ANN model was able to effectively identify nonlinear patterns in the exchange rate data and generate highly accurate forecasts. However, when compared with SVR, the ANN approach produced slightly higher prediction errors, indicating that SVR provided more stable and reliable performance for the JISDOR dataset analyzed in this study. These results are consistent with previous studies reporting that ANN performs well in forecasting financial time-series data characterized by nonlinear relationships. Nonetheless, the present study suggests that an optimized SVR model may achieve better generalization capability for exchange rate prediction, particularly when using relatively limited predictor variables.

4. Model Comparison and Evaluation

The following are the prediction accuracy results for the SVR and ANN models, as shown in Table 15.

Table 15. Prediction Accuracy Results of SVR and ANN Models

Model Type	RMSE	MAPE
Support Vector Regression	43.48656	0.22 %
Artificial Neural Network	49.71652	0.25 %

Table 15 shows that both methods, Support Vector Regression (SVR) and Artificial Neural Network (ANN), are able to provide highly accurate prediction results for the JISDOR exchange rate data. However, the SVR method shows a better level of accuracy than ANN. This is indicated by the lower RMSE value (43.49) and smaller MAPE (0.22%) in SVR compared to ANN which has an RMSE of 49.72 and a MAPE of 0.25%. Both models also do not experience overfitting problems, because the MAPE value in the testing data is lower than in the training data, so it can be concluded that both SVR and ANN have good generalization capabilities. However, SVR is superior in modeling JISDOR exchange rate data compared to ANN because it produces smaller prediction errors and higher accuracy, as shown in Figure 9.

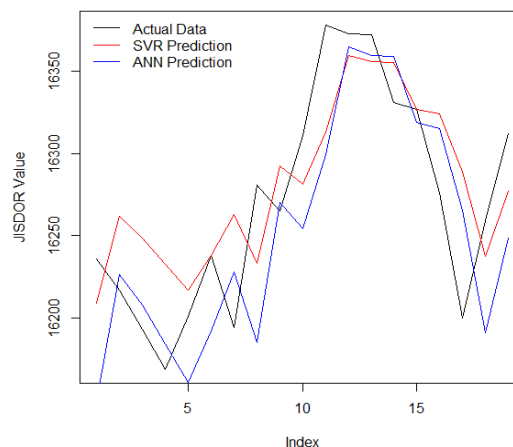


Figure 9. Comparison of Actual, SVR, and ANN Prediction Results

Figure 9 presents a comparison between the actual JISDOR exchange rate data and the predicted results generated by the Support Vector Regression (SVR) and Artificial Neural Network (ANN) models. The black line represents the actual data, while the red and blue lines correspond to the SVR and ANN predictions, respectively. It can be observed that both models closely follow the actual data trend, indicating a high level of prediction accuracy. The SVR model's prediction line appears slightly closer to the actual data compared to the ANN, suggesting that SVR produces smaller prediction errors and better captures the JISDOR exchange rate movement pattern.

This finding directly addresses the research gap identified in the introduction regarding the limited application of optimized SVR for JISDOR exchange rate prediction in the Indonesian context. The results are also consistent with previous studies showing that nonlinear machine learning approaches tend to outperform conventional forecasting methods in capturing complex financial time-series dynamics. From a practical perspective, the high predictive accuracy of the SVR model may support monetary authorities, investors, and business actors in improving exchange rate risk management and short-term financial decision-making under volatile market conditions. Nevertheless, this study is still limited by the use of lagged exchange rate variables and a relatively short observation period, so future studies are recommended to incorporate broader macroeconomic indicators and hybrid volatility-based models such as GARCH-SVR to improve forecasting robustness.

D. CONCLUSION AND SUGGESTIONS

This study demonstrates that a Support Vector Regression (SVR) model with a deeply optimized RBF kernel can predict the Rupiah exchange rate against the US Dollar based on JISDOR data with high accuracy, achieving an RMSE of 43.49 and a MAPE of 0.22%. SVR outperforms the Artificial Neural Network (ANN) model, which recorded an RMSE of 49.72 and MAPE of 0.25%, confirming the superiority of SVR in both accuracy and generalization for this forecasting task. The success of the SVR model opens up opportunities for applying similar methods to assist monetary policy decision-making and exchange rate risk management in Indonesia. Nevertheless, this study has several limitations: the data is restricted to the January

2023–January 2025 period, which may not fully capture longer-term structural exchange rate dynamics, and the model relies solely on historical JISDOR values without incorporating exogenous macroeconomic variables such as inflation rates, interest rate differentials, or commodity prices that are known to influence exchange rate movements. For future research, it is recommended to extend the observation period to cover a longer time horizon, incorporate relevant macroeconomic and external variables as additional input features, and explore hybrid models that combine SVR with GARCH-family models to better capture the heteroscedasticity inherent in foreign exchange market volatility.

ACKNOWLEDGEMENT

The authors acknowledge the Faculty of Sains and Technology at Airlangga University, Surabaya, Indonesia, for their continuous support and resources, which significantly contribute to this research.

REFERENCES

- Abdolrasol, M. G. M., Hussain, S. M. S., Ustun, T. S., Sarker, M. R., Hannan, M. A., Mohamed, R., Ali, J. A., Mekhilef, S., & Milad, A. (2021). Artificial neural networks based optimization techniques: A review. *Electronics*, 10(21), 2689. <https://doi.org/10.3390/electronics10212689>
- Ahmar, A. S., Al Idrus, S., & Asmar, A. (2024). Analyzing Rupiah-USD exchange rate dynamics: A Study with ARCH and GARCH Models. *JOIV: International Journal on Informatics Visualization*, 8(3–2), 1802–1809. <https://doi.org/10.62527/joiv.8.3-2.3251>
- Aji, A. B., & Surjandari, I. (2020). Hybrid vector autoregression–recurrent neural networks to forecast multivariate time series jet fuel transaction price. *IOP Conference Series: Materials Science and Engineering*, 909(1), 012079. <https://doi.org/10.1088/1757-899X/909/1/012079>
- Alida, M., & Mustikasari, M. (2020). Rupiah Exchange Prediction of US Dollar Using Linear, Polynomial, and Radial Basis Function Kernel in Support Vector Regression. *Jurnal Online Informatika (JOIN)*, 5(1), 53–60. <https://doi.org/10.15575/join.v5i1.537>
- Amri, I. F., et al. (2025). Rupiah exchange rate prediction against the US Dollar using a deep neural network with a multi-output sliding window approach. *MethodsX*, 14, 103692. <https://doi.org/10.1016/j.mex.2025.103692>
- Andreas, C. (2025). Perbandingan Pendekatan Statistika dan Machine Learning dalam Peramalan Data Time Series. *Jurnal Sistem Informasi Triguna Dharma (JURSI TGD)*, 4(4), 778–785. <https://doi.org/10.53513/jursi.v4i4.11429>
- Ayitey Junior, M., Appiahene, P., Appiah, O., & Bombie, C. N. (2023). Forex market forecasting using machine learning: Systematic Literature Review and meta-analysis. *Journal of Big Data*, 10(1), 9. <https://doi.org/10.1186/s40537-022-00676-2>
- Devianto, D., Yollanda, M., Maiyastri, & Yanuar, F. (2023). The soft computing FFNN method for adjusting heteroscedasticity on the time series model of currency exchange rate. *Frontiers in Applied Mathematics and Statistics*, 9, 1045218. <https://doi.org/10.3389/fams.2023.1045218>
- Du, K.-L., Jiang, B., Lu, J., Hua, J., & Swamy, M. N. S. (2024). Exploring kernel machines and support vector machines: Principles, techniques, and future directions. *Mathematics*, 12(24), 3935. <https://doi.org/10.3390/math12243935>
- Farizawani, A. G., Puteh, M., Marina, Y., & Rivaie, A. (2020). A review of artificial neural network learning rule based on multiple variant of conjugate gradient approaches. *Journal of Physics: Conference Series*, 1529(2), 022040. <https://doi.org/10.1088/1742-6596/1529/2/022040>
- Galena, M., Sediono, S., Mardianto, M., & Pusporani, E. (2025). Prediction Analysis of Jakarta Composite Index Movement Using Support Vector Regression Method. *G-Tech: Jurnal Teknologi Terapan*, 9(1), 68–77. <https://doi.org/10.70609/gtech.v9i1.5879>
- Halder, T., Ganguly, S., Mukhopadhyay, S., et al. (2024). A Hybrid-SFLA-Based Hyper-Parameter Tuning Algorithm in SVR for Exchange Rate Prediction. *SN Computer Science*, 5(3), 285. <https://doi.org/10.1007/s42979-023-02598-z>

- Hu, J., Liu, C., & Gao, J. (2021). An effective hybrid approach for forecasting currency exchange rates. *Sustainability*, 13(5), 2761. <https://doi.org/10.3390/su13052761>
- Jazayeri, K. (2024). Predicting multi-horizon currency exchange rates using a stacked ensemble of random forest and SVR. *Intelligent Decision Technologies*, 18(1), 297–325. <https://doi.org/10.3233/IDT-230337>
- Johnson, D. W. (2024). N. Gregory Mankiw (1958–). In R. A. Cord (Ed.), *The Palgrave Companion to Harvard Economics* (pp. 1039–1064). Springer International Publishing. https://doi.org/10.1007/978-3-031-52053-2_42
- Marwadi, M., & Rokhim, R. (2016). Jakarta Interbank Spot Dollar Rate (JISDOR) as The Reference Rate: Is It Effective? *Indonesian Capital Market Review*, 8(2), 5. <https://doi.org/10.21002/icmr.v8i2.5012>
- Mubarok, F., Purnamasari, I., & Yuniarti, D. (2024). Peramalan Kurs Rupiah Terhadap Dolar Amerika Menggunakan Model Autoregressive Integrated Moving Average. *Journal of Mathematics: Theory and Applications*, 6(2), 225–236. <https://doi.org/10.31605/jomta.v6i2.4072>
- Muchtadi-Alamsyah, I., Viltoriano, R., Harjono, F., Nazaretha, M., Susilo, M., Bayu, A., Josaphat, B., Hakim, A., & Syuhada, K. (2024). Support vector regression-based heteroscedastic models for cryptocurrency risk forecasting. *Applied Soft Computing*, 162, 111792. <https://doi.org/10.1016/j.asoc.2024.111792>
- Ningrum, A. F., Purwanto, D., & Kusuma Wardani, A. (2025). Evaluation of Deep Learning Optimizers for Predicting JISDOR Exchange Rates Using LSTM Networks. *Journal Of Data Insights*, 3(1), 32–41. <https://doi.org/10.26714/jodi.v3i1.726>
- Pratiwi, D. M., Firdaniza, & Kusuma, D. A. (2025). Forecasting the Rupiah Exchange Rate Influenced by Several Factors Using the Improve Grey Model (1,3). *Jambura Journal of Mathematics*, 7(1), 1–7. <https://doi.org/10.37905/jjom.v7i1.27954>
- Purnama, D. I., & Hendarsin, O. P. (2020). Peramalan Jumlah Penumpang Berangkat Melalui Transportasi Udara di Sulawesi Tengah Menggunakan Support Vector Regression (SVR). *Jambura Journal of Mathematics*, 2(2), 49–59. <https://doi.org/10.34312/jjom.v2i2.4458>
- Pusporani, E., Fariz Fadillah Mardianto, M., Sediono, 'iffah, A., Firdausy, A. R., Widyatama, E., & Huda, M. F. (2022). Prediction Of National Strategic Commodity Prices Based On Multivariate Nonparametric Time Series Analysis. *Communications in Mathematical Biology and Neuroscience*, 2022, Article 60. <https://doi.org/10.28919/cmbn/7712>
- Pusporani, E., Suhartono, & Prastyo, D. D. (2019). Hybrid multivariate generalized space-time autoregressive artificial neural network models to forecast air pollution data at Surabaya. *AIP Conference Proceedings*, 2194, 020073. <https://doi.org/10.1063/1.5139822>
- Saputra, G. H., Wigena, A. H., & Sartono, B. (2019). Penggunaan support vector regression dalam pemodelan indeks saham syariah indonesia dengan algoritme grid search. *Indonesian Journal of Statistics and Its Applications*, 3(2), 148–160. <https://doi.org/10.29244/ijsa.v3i2.172>
- Sari, A. N., Zuleika, T., Mardianto, M. F. F., & Pusporani, E. (2025). Application of Support Vector Regression in Time Series Analysis of Dior Stock Prices. *Zeta - Math Journal*, 10(1), 51–60. <https://doi.org/10.31102/zeta.2025.10.1.51-60>
- Wang, Z., Xu, H., Xia, L., Zou, Z., & Soares, C. G. (2020). Kernel-based support vector regression for nonparametric modeling of ship maneuvering motion. *Ocean Engineering*, 216, 107994. <https://doi.org/10.1016/j.oceaneng.2020.107994>
- Zhang, F., & O'Donnell, L. J. (2020). Support vector regression. In *Machine learning* (pp. 123–140). Elsevier. <https://doi.org/10.1016/B978-0-12-815739-8.00007-9>
- Zhang, X., Zhong, C., & Abualigah, L. (2022). Foreign exchange forecasting and portfolio optimization strategy based on hybrid-molecular differential evolution algorithms. *Soft Computing*, 27(7), 3921–3938. <https://doi.org/10.1007/s00500-022-07309-3>