Jurnal Teknik Informatika (JUTIF) Vol. 6, No. 4, August 2025, Page. 2331-2347
P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5057

Stacking Ensemble RNN-LSTM Models for Forecasting the IDR/USD Exchange Rate with Nonlinear Volatility

Windy Ayu Pratiwi[1], I Made Sumertajaya*[2], Khairil Anwar Notodiputro [3]

[1,2,3]School of Data Science, Mathematics, and Informatics, IPB University, Bogor, Indonesia

Email: [2]imsjaya@apps.ipb.ac.id

Received : Jul 8, 2025; Revised : Aug 1, 2025; Accepted : Aug 3, 2025; Published : Aug 19, 2025

Abstract

Abstract - Predicting exchange rates with high volatility and nonlinear patterns presents a critical challenge in financial analysis. Deep learning models such as RNN and LSTM are widely used for their ability to capture temporal dependencies, yet each has limitations when applied individually. This study aims to enhance the prediction accuracy of the Indonesian Rupiah (IDR) to US Dollar (USD) exchange rate by implementing a stacking ensemble approach that combines RNN and LSTM models. The dataset consists of 522 weekly observations from January 2015 to December 2024, sourced from the official website of Bank Indonesia (bi.go.id). In the proposed framework, RNN and LSTM serve as base learners, while linear regression acts as the meta-learner. Model performance is evaluated using RMSE, MAPE, and MSE. The results indicate that the stacking ensemble consistently outperforms the individual models, achieving an RMSE of 117.91, a MAPE of 0.01, and an MSE of 13,901.67. The model effectively captures historical patterns and delivers stable and accurate predictions. In conclusion, the stacking ensemble approach developed in this study contributes to the advancement of ensemble learning techniques in computer science and offers practical value for financial decision-makers, particularly in managing complex and dynamic exchange rate scenarios.

Keywords : *deep learning, exchange rate prediction, LSTM, RNN, stacking ensemble, time series analysis.*

1. INTRODUCTION

Forecasting refers to a structured approach used to anticipate future outcomes based on historical data, aiming to deliver precise estimations of upcoming conditions or influencing variables [1]. A commonly analyzed data format in forecasting is time series data, which consists of sequential observations collected at consistent time intervals on a specific entity or phenomenon [2]. These data often reveal recognizable patterns such as long-term trends, seasonal behaviors, and irregular fluctuations, all driven by temporal relationships between observations. The core components trend, seasonality, and residuals are essential for decomposing time series data to gain insights into its intrinsic structure [3], [4], [5], [6]. Effective forecasting models must be capable of learning these historical structures while also adapting to ongoing and complex changes.

With recent technological progress, deep learning has emerged as a promising solution for modeling nonlinear time dependencies. Two of the most widely used architectures in this field are RNNs and LSTM networks [7], [8]. RNNs are tailored for processing sequential data like time series, using internal memory to capture long-term dependencies. However, RNNs are affected by the vanishing gradient problem, which hampers their ability to model complex, long-range patterns [9]. Research such as RNN applications in rainfall forecasting [10] and comparative analysis of RNN and ARIMA in gold price prediction[11], demonstrates that while RNNs can be effective, their success largely depends on how the training data is configured and its specific characteristics.

To overcome these limitations, the LSTM architecture emerged as a more sophisticated alternative. LSTM is widely adopted for forecasting due to its ability to produce consistent long-term

Vol. 6, No. 4, August 2025, Page. 2331-2347
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5057

predictions. Unlike traditional RNNs, which suffer from vanishing gradient issues during backpropagation and struggle to learn long-term dependencies, LSTM incorporates a gated memory system for selective information retention and update over time. An LSTM unit consists of three critical gates input, forget, and output that regulate how information enters, persists in, or exits the memory cell, enabling effective modeling of long-range dependencies [12], [13]. Despite its advantages, LSTM's complex architecture leads to slower training and inference, increased memory and computational demands, and a heightened risk of overfitting, especially when training data is limited or lacks diversity [7], [14], [15].

Numerous studies have highlighted LSTM's effectiveness in forecasting complex and volatile time series data. For example, Septiani et al.[16]. found that LSTM delivered the lowest RMSE when compared to models like GARCH, GRU, and CNN for forecasting the Rupiah–USD exchange rate. Similarly, Magfirrah et al. [17]. reported that LSTM outperformed ARIMA in predicting peak wind speeds in Kupang, and Danis et al. [18] confirmed its performance by achieving a 1.53% MAPE in predicting the Yen–Rupiah exchange rate based on daily data.

Additionally, Wang et al [19] proposed a hybrid forecasting model integrating Empirical Mode Decomposition (EMD), Principal Component Analysis (PCA), Random Forest (RF), and LSTM to forecast wind energy. Their approach enhanced accuracy by minimizing noise, reducing dimensionality, and extracting vital features. Compared to benchmark models such as BP, SVM, and traditional LSTM, the EMD-PCA-RF-LSTM model demonstrated superior performance with an RMSE of 2.69576, MAE of 1.73981, and R^2 of 0.9699, indicating its capability in modeling complex environmental time series.

Given the individual strengths and weaknesses of RNNs and LSTMs, ensemble learning offers a promising strategy for improved forecasting. Ensemble models combine predictions from several base learners to boost both accuracy and generalization [20]. Among these, stacking where a meta-learner synthesizes outputs from various models into a final prediction is particularly effective [21]. Empirical results show that stacking tends to reduce forecast errors, especially in applications like electricity consumption forecasting, when compared to single-model approaches [1].

Another study that utilized an ANN ensemble to estimate ice thickness in Canadian lakes demonstrated significant improvements in generalization. The stacking method, in particular, was found to be more effective than simple averaging in producing accurate predictions [22]. Guo et al further confirmed the benefits of stacking in short-term load forecasting by integrating LSTM-based base learners and using Lasso regression as a meta-learner, resulting in lower MAPE values. In the domain of health analytics, a study applied a stacking ensemble combined with SMOTE-ENN for diabetes classification and achieved an accuracy of 97.3%, outperforming individual classifiers [23]. Moreover, Wibowo and Pratama, implemented a stacking model for sentiment classification of hotel reviews using multiple base classifiers and achieved an accuracy of 98%, demonstrating the robustness of stacking in handling complex text data. Despite these promising outcomes, the majority of stacking applications are concentrated in domains such as classification, regression, and structured prediction tasks. These findings collectively reinforce the applicability and superiority of stacking ensemble models across a variety of domains, including time series forecasting, classification, and sentiment analysis [24].

In this study, the dataset comprises the exchange rate of the Indonesian Rupiah (IDR) against the United States Dollar (USD), a time series known for its high volatility due to both global and domestic economic dynamics. This dataset was considered because it aligns with the targets of the Sustainable Development Goals (SDGs), particularly Goal 8 on *Decent Work and Economic Growth*, which emphasizes inclusive and sustainable economic growth, employment, and productivity improvements [25]. Exchange rate instability can impact labor markets, investment confidence, and income inequality factors directly relevant to SDG 8 targets on productive employment and economic resilience. Furthermore, this dataset exhibits characteristics such as short-term fluctuations, non-stationarity, and

Vol. 6, No. 4, August 2025, Page. 2331-2347
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5057

complex nonlinear patterns, making it a challenging benchmark for accurate forecasting and for modeling macroeconomic indicators linked to SDG achievement [26].

To address these challenges, this research proposes stacking ensemble model that combines RNN and LSTM architectures, aiming to deliver more accurate, stable, and adaptive predictions. Although stacking ensemble shows strong potential, its application in forecasting highly volatile and nonlinear time series remains limited. In particular, applications of stacking in exchange rate forecasting especially for developing economies such as Indonesia are still relatively scarce in the literature. This highlights a research gap in exploring how hybrid deep learning ensembles can perform under extreme volatility and nonlinear dynamics. Therefore, this study seeks to demonstrate the effectiveness of stacking ensemble and contribute to the advancement of deep learning-based forecasting approaches for handling complex and unstable time series. The objective of this research is to evaluate the performance of a stacking ensemble of RNN and LSTM models in forecasting the IDR/USD exchange rate, compare it against individual deep learning models (RNN and LSTM) and a conventional statistical model (ARIMA), and provide empirical evidence of its effectiveness in improving forecasting accuracy in the presence of high volatility and nonlinear structure.

2. METHOD

This study implements a stacking ensemble model by combining two deep learning architectures: Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM). The stacking model is designed to taking advantage of RNN for short-term patterns and LSTM for long-term sequence modeling, while using a linear regression model as the meta-learner to combine predictions.

2.1. Data and Variables

This study uses time series data representing the daily exchange rate of the Indonesian Rupiah (IDR) against the United States Dollar (USD), covering the period from January 2015 to December 2024. The data were obtained from the official website of Bank Indonesia (https://www.bi.go.id), specifically from the "Kurs Transaksi Bank Indonesia" page, which provides daily exchange rate records. The data were accessed via manual download in Excel format and further processed using Python. To reduce noise and better capture temporal trends, the data were resampled into weekly frequency by extracting the highest exchange rate value (selling rate) from each week. This conversion aims to capture the peak volatility of each week while minimizing random fluctuations that may occur on a daily basis. As a result, a total of 522 weekly observations were compiled and used for modeling and evaluation. The primary variable in this study is the weekly IDR/USD exchange rate, which serves as the univariate target to be forecasted. No external or exogenous variables are included; the models rely solely on the temporal patterns within the data itself.

2.2. Recurrent Neural Network (RNN)

RNN represent a category of a type of neural network built to handle time-related or sequential data inputs. They incorporate a built-in memory system that stores prior inputs to inform future output decisions. The RNN functions by transmitting information from one input to the next within a sequence. Each incoming input is stored in memory and then used in processing the next one, continuing in this way until the entire sequence has been processed [27]. Within the RNN architecture, the units in the hidden layers are interconnected, meaning they do not operate in isolation. At each time step, the hidden layer receives input not only receiving inputs not only from the input layer but also from the previous hidden state. Theoretically, RNNs are capable of handling input sequences of any length [28].

Vol. 6, No. 4, August 2025, Page. 2331-2347
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5057

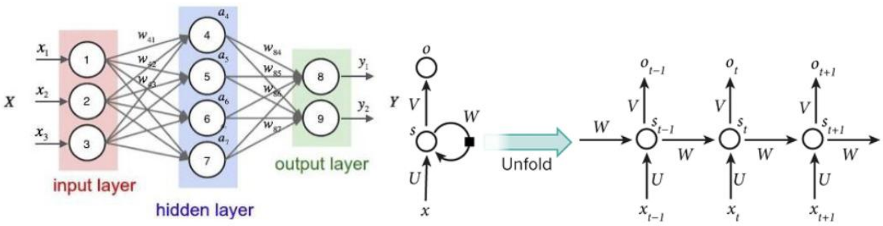

Figure 1. RNN Hidden Layer Structure

Figure 1 illustrates the basic structure of an RNN's hidden layer. In this model, each neuron receives input not only from the input layer but also from the hidden state of the previous time step. This architecture helps RNNs retain recent information by utilizing earlier input data. The unrolled structure on the right side of the figure shows how RNNs handle sequences over time, where the hidden state from the previous time step (s_{t-1}) contributes to forming the current hidden state (s_t), This process continues across each time step. This recursive process happens at each time step, enabling the output to reflect both current input and past memory. Such temporal dependency makes RNNs particularly suited for handling data with temporal sequences, such as time series.

Previous empirical research has confirmed RNNs' effectiveness, for example, Achmad et al [29]. Developed a machine translation system for Indonesian–Sentani Papua using RNN-GRU, achieving a BLEU score of 35.3, which indicates strong performance in handling low-resource language sequences. Meanwhile, Larasati et al. [30] compared RNN and LSTM for forecasting daily turnover in a motorcycle spare parts store. Although LSTM achieved higher predictive accuracy. maintained consistent accuracy (MAE = 0.092), highlighting its ability to model short-term seasonality in time series.

2.3. Long Short-Term Memory (LSTM)

The LSTM is a model designed to determine which information should be retained and which should be discarded during processing a task handled by individual neuron units. LSTM is widely used for modeling ordered data formats like time series, video, and text [31], [32]. It can be viewed as an extension of the RNN, as both share a foundational architecture composed of input, hidden, and output layers. The key innovation in LSTM lies in its hidden layer, which introduces memory cells and specialized gate mechanisms that address the vanishing gradient issue commonly found in traditional RNNs. These gate mechanisms, including the forget gate, input gate, and output gate, regulate the flow of information using sigmoid and tanh activation functions[31]. The forget gate selects which earlier data to remove, the input gate identifies which new data to retain, and the output gate directs the relevant information to the next time step [32], [33]. This architecture enables LSTM to preserve relevant information across long sequences and handle nonlinear temporal dependencies effectively.

LSTM's ability to retain relevant information over long sequences and manage nonlinear temporal dependencies effectively has been validated by several studies. Budiprasetyo et al. [34] applied a multi-layer LSTM model to forecast stock prices of sharia-compliant companies in Indonesia and achieved high predictive accuracy, with MAPE values below 3% across different sectors. Similarly, Hussein and Azhar [33] demonstrated the robustness of LSTM in predicting global oil prices, showing its ability to outperform conventional methods in capturing fluctuating economic conditions. To provide deeper insight into how LSTM works internally, Figure 2 presents a detailed illustration of the model's internal architecture.

Figure 2 presents the basic framework of an LSTM network. This model manages information flow through three core gates: the forget gate, input gate, and output gate. The forget gate decides which parts of the previous memory should be discarded. The input gate regulates which new data is added to the memory. The output gate decides what memory content is passed on to the next time step. These

Vol. 6, No. 4, August 2025, Page. 2331-2347
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5057

operations rely on sigmoid and tanh activation functions to adjust memory content selectively. This structure allows LSTM to retain essential information over extended sequences and ignore less relevant inputs, making it highly effective for time series analysis involving complex and long-term dependencies.

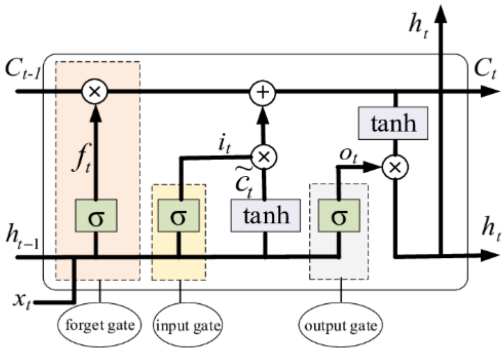

Figure 2 Architecture of the Long Short-Term Memory (LSTM) Model

2.4. Stacking Ensemble

According to Silfiani and Suhartono (2012), stacking is a technique used to combine multiple predictors through the construction of a linear combination, with the aim of improving forecasting accuracy. In this method, each predictor is assigned a specific coefficient weight, such that the resulting linear combination can produce more accurate predictions [35]. The concept of stacking itself originates from the idea of stacked generalization introduced by Wolpert (1992), which proposes using a meta-learner to learn from the outputs of base models and improve generalization performance [36]. Stacking ensemble is further developed as an approach within ensemble learning, which involves training several base models in parallel and independently. The outputs of these base models are then combined through a meta-learning algorithm to produce a more robust final prediction. In other words, stacking integrates the predictions from multiple base learners into a higher-level model (meta-learner) to enhance predictive accuracy and generalization[37].

In this research, RNN and LSTM are utilized as level-0 base learners, where RNN is responsible for identifying short-term patterns, while is intended to learn long-range patterns in time series datasets. Linear regression serves as the meta-model that merges the prediction outputs of RNN and LSTM, selected for its ability to form a weighted combination of the predictions from base learners, thereby producing more stable and accurate forecasts. Gao et al. (2025) demonstrated that stacking deep learning-based base learners using a residual learning scheme significantly improved prediction accuracy for moderate-sized time series datasets. Their proposed SE-DRVFL, a deep ensemble framework utilizing random vector functional links model achieved lower error rates compared to traditional ensembles and individual models, showing that the residual-based architecture helped better exploit the strengths of each base learner and enhance generalization [38].This supports the use of stacking ensembles that combine RNN and LSTM in this study, particularly when dealing with nonlinear and volatile patterns in time series data.

The stacking ensemble was implemented in two stages. First, the RNN and LSTM models were trained separately using the training dataset. After training, each model generated predictions on a validation subset (15% of the training data), which were stored as features for the meta-model. These predictions formed the input for the level-1 meta-learner linear regression. The linear regression model was then trained to learn the optimal weights for combining the base model outputs. During final prediction, the full training dataset was used to retrain RNN and LSTM. Their predictions on the test set were then passed through the trained meta-model to generate the final forecast. This approach aligns with the findings of Ismail et al [39], who proposed a GA-Stacking ensemble that combines diverse base

Vol. 6, No. 4, August 2025, Page. 2331-2347
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5057

learners through a linear regression meta-model. Their results demonstrated that stacking significantly enhances forecasting accuracy, especially when applied to complex and volatile time series data such as COVID-19 cases. The formulation used by the meta-learner to generate the final forecast based on the predictions of RNN and LSTM can be represented by the following equation:

$$\hat{Y} = \alpha + b_1 \times \hat{Y}_{RNN} + b_1 \times \hat{Y}_{LSTM} + \varepsilon \quad (1)$$

- $\hat{Y}_{RNN}$ and $\hat{Y}_{LSTM}$ are the predicted outputs from the base models (RNN and LSTM),
- α is the intercept (bias),
- b_1 and b_2 are the weights learned by the meta-model (linear regression),
- ε is the error term.

formulation 1 allows the stacking model to leverage the strengths of both RNN and LSTM, improving forecasting accuracy through model ensemble. The meta-model integrates the base model outputs by learning optimal weights that minimize prediction error. This structure is further illustrated in Figure 3, which shows how the predictions from RNN and LSTM are aggregated using a meta-level linear regression model to generate the final forecast

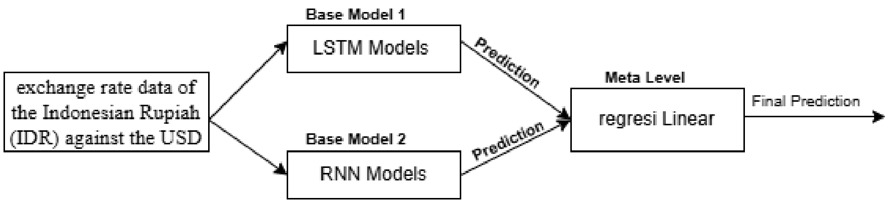

Figure 3 Architecture of the Stacking Ensemble Model

Figure 3 depicts the basic architecture of the stacking ensemble model. In this structure, multiple base learners such as RNN and LSTM are trained in parallel using the same input data. Each base learner produces its own prediction, which is then combined through a meta-learner algorithm to form the final forecast. This process involves a level-1 model that aggregates the outputs from all base learners using a linear regression approach. Through this stacking mechanism, the combined predictions from diverse base models are expected to improve accuracy, enhance generalization, and yield more stable and reliable forecasting results compared to single models.

This research adopts a stacking ensemble strategy that combines outputs from both RNN and LSTM models, with a linear regression algorithm serving as the meta-learner. Linear regression was selected due to its simplicity, clarity, and capability to assign optimal weights to the base models' outputs. The ensemble technique aims to minimize individual model errors and enhance the model's ability to generalize. Model performance is evaluated using three commonly used forecasting error metrics: MSE, RMSE, and MAPE.

2.5. Evaluation Metrics

Three error-based metrics were used to assess model accuracy:

1. Mean Squared Error (MSE): Calculates the average of the squared differences between predicted and actual values, heavily penalizing larger errors.

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

2. Root Mean Squared Error (RMSE): Represents the error in the same scale as the original data, making it more interpretable for practical use.

$$RMSE = \left(\frac{\sum_{t=1}^{n}((y)_t - \hat{y}_t)^2}{n}\right)^{1/2} \quad (3)$$

Vol. 6, No. 4, August 2025, Page. 2331-2347
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5057

3. Mean Absolute Percentage Error (MAPE): Measures the prediction error as a percentage of the actual values, providing a standardized evaluation metric.

$$MAPE = \frac{\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right|}{n} \times 100 \quad (4)$$

These metrics provide a quantitative evaluation of prediction accuracy. MSE gives greater weight to larger errors by squaring them, RMSE presents the error in the same unit as the target, enhancing interpretability, and MAPE expresses error as a percentage, making it easier to compare across different datasets. Together, these metrics offer a thorough evaluation of the model's predictive performance and demonstrate the reliability of the proposed stacking ensemble framework.

2.6. Steps of Analysis

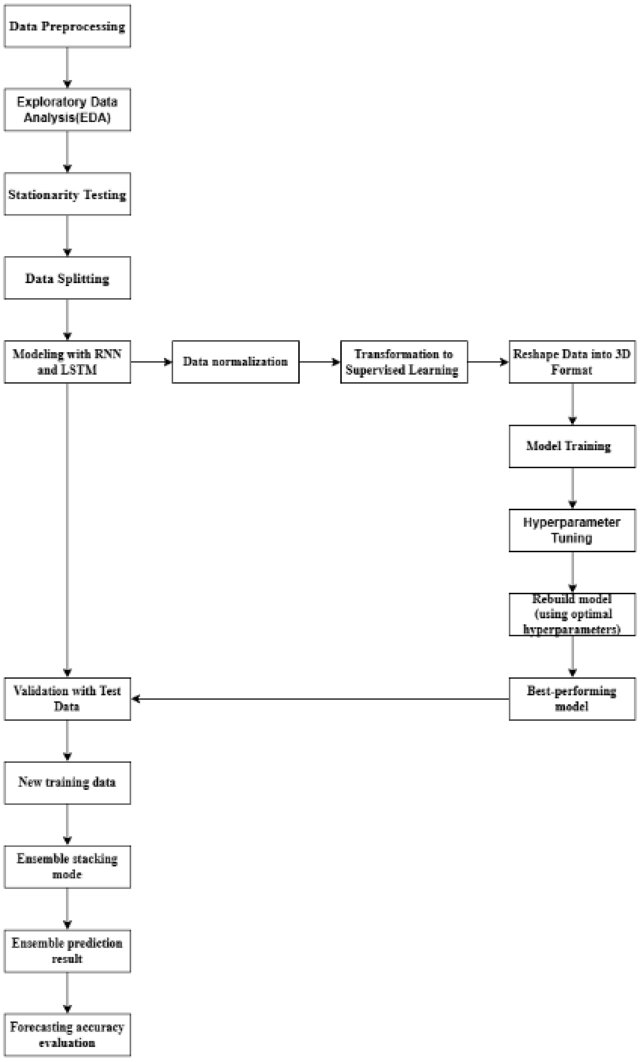

Figure 4. Workflow For Building the Stacking Ensemble Model

This study employs stacking ensemble method applied to RNN and LSTM models. Data analysis was conducted using Python software with TensorFlow/Keras and Scikit-learn libraries. The study follows these main stages of analysis:

1. Data Preprocessing: Daily exchange rate data are transformed into weekly data by selecting the highest value each week. Missing values are handled, and the data are normalized and reshaped to fit sequential input formats suitable for RNN and LSTM architectures.

Vol. 6, No. 4, August 2025, Page. 2331-2347
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5057

2. Exploratory Data Analysis (EDA): Initial exploration is conducted to identify general trends, patterns, and variability. Stationarity is examined using the autocorrelation function (ACF), and temporal structures are visualized.
3. Data Splitting: The data is divided into two parts, training and testing. The training portion covers 85% of the data, ranging from January 2015 to June 2023, while the remaining 15% from July 2023 to December 2024 is allocated for testing.
4. Hyperparameter Tuning: A grid search approach is employed to fine-tune the parameters of the RNN and LSTM models. Parameters such as the number of neurons, learning rate, batch size, dropout rate, and number of epochs are tuned based on validation performance.
5. Individual Model Training (RNN and LSTM): Each of the RNN and LSTM models is trained separately using the training dataset. The predictions produced on the 15% validation portion are saved and later used as features for the stacking meta-model.
6. Stacking Ensemble Training: The meta-learner (linear regression) is trained on predictions generated by RNN and LSTM. During testing, final predictions are made by first generating RNN and LSTM outputs on the test data, which are then passed through the trained meta-model.
7. Model Evaluation and Conclusion Drawing The evaluation of RNN, LSTM, and stacking ensemble models is carried out using metrics including MSE, RMSE, and MAPE. The model with the best performance on the test set is selected and applied to both simulated and empirical exchange rate data for comparison.

3. RESULT

The results and discussion are presented in a structured and detailed manner. Model performance is reported using tables and graphs to illustrate accuracy and comparative results. Each figure and table is numbered, titled, and referred to in the text for clarity. Metrics such as MSE, RMSE, and MAPE are used to evaluate forecasting performance. All visual elements, including equations, graphs, and tables, follow consistent formatting to ensure readability and coherence throughout the manuscript.

3.1. Data Preprocessing

This research utilizes exchange rate data of the Indonesian Rupiah (IDR) against the United States Dollar (USD), specifically targeting the weekly peak values of the IDR selling rate. The dataset spans from January 2015 to December 2024 and is retrieved from the official website of Bank Indonesia (bi.go.id). The data conversion process involves selecting the maximum selling rate of the Rupiah against the US Dollar for each week. This step aims to provide a more stable representation of the currency trend by minimizing the impact of extreme daily fluctuations, resulting in a total of 522 weekly observations used for analysis.

Table 1. Weekly Exchange Rate (Idr/USD): First Ten Observations

Period	Selling Exchange Rate (IDR/USD)
04 January 2015	12.536
11 January 2015	12.796
18 January 2015	12.680
25 January 2015	12.722
01 February 2015	12.688
08 February 2015	12.764
15 February 2015	12.858
22 February 2015	12.913
01 March 2015	12.951
08 March 2015	12.536

Table 1 presents the first ten weekly observations of the IDR/USD selling exchange rate, starting from 04 January 2015 to 08 March 2015. The exchange rate fluctuates within a narrow range, from

Jurnal Teknik Informatika (JUTIF) Vol. 6, No. 4, August 2025, Page. 2331-2347
P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5057

12.536 to 12.951, indicating relatively stable currency movement during the early part of the observed period.

3.2. Exploratory Data Analysis (EDA)

During the data exploration phase, the initial step is to divide the dataset into training and testing sets. This separation is crucial to ensure the model is evaluated on previously unseen data, helping to prevent overfitting and provide a more accurate reflection of its performance. In this study, the training data spans from January 2015 to June 2023 (85%), while the test data covers the period from July 2023 to December 2024 (15%). This split allows the model to be trained on a long historical dataset and then tested on the most recent data, which was not used during training, to assess its predictive accuracy.

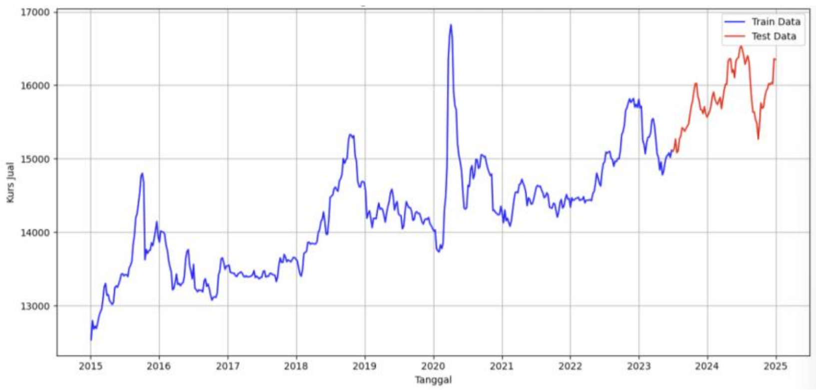

Figure 5. Plot of Training and Test Data Split

Figure 5 illustrates how the dataset is divided into training and testing segments. The training data, shown in blue, covers January 2015 to June 2023 (85%) and is used to build the model. In contrast, the test data, highlighted in red, spans from July 2023 to December 2024 (15%) and is used to assess the model's performance on recent data. The model is trained using the historical dataset, while the test dataset helps evaluate how well it performs on new or unseen data points. This clear distinction allows the model to learn from the past and be tested on future outcomes. Its predictions are then compared with the actual test results, and performance is measured using metrics such as RMSE, MSE, and MAPE.

3.3. Stationarity Analysis

After exploring the data and dividing it into training and testing sets, a crucial step in time series analysis is to assess its stationarity. Stationarity refers to a statistical property where the mean, variance, and autocorrelation of a time series stay stable over time. Checking for stationarity is important for choosing the right modeling approach. One common method to do this is the Autocorrelation Function (ACF), which measures the correlation between observations at different time intervals. If significant autocorrelation appears at specific lags, it suggests the series is non-stationary, indicating recurring patterns or trends. If autocorrelation weakens and approaches zero, the series is likely stationary.

Figure 6 displays the Autocorrelation Function (ACF) plot, which illustrates the correlation between observations at various lag intervals. In this plot, we observe that the autocorrelation is relatively high at the initial lags but decreases significantly after several lags. This indicates short-term dependencies within the time series. Although the data shows significant autocorrelation at the early lags suggesting some level of non-stationarity this does not pose a critical issue when applying deep learning methods such as LSTM and RNN. According to Lin and Feng (2022) and Long et al. (2023), these models are capable of handling non-stationary data directly, without the need for prior transformations to induce stationarity [40], [41]. Therefore, techniques such as differencing or other transformations are not required prior to modeling, as LSTM and RNN can still perform effectively on

Vol. 6, No. 4, August 2025, Page. 2331-2347
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5057

raw non-stationary data. Based on this, the stacking ensemble model leverages the strengths of base models, also referred to as level-0 models. In this study, two base models are employed, namely LSTM and RNN, each of which is trained independently on the same input data. By combining the distinct perspectives offered by these models, the stacking framework aims to improve generalization and model robustness compared to relying on a single model. The following section provides a more detailed explanation of the level-0 models.

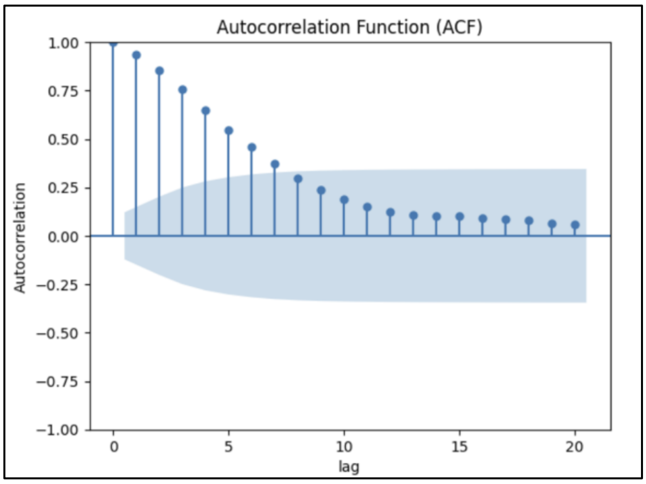

Figure 6. Autocorrelation function (ACF) plot

3.4. Base Models Individual Model Training

3.4.1. Architecture and Hyperparameters

The level-0 models used in this study consist of two neural network architectures, LSTM and RNN. This model analysis aims to assess the performance of two neural network architectures: LSTM and RNN in time series forecasting. These models were selected due to their proven effectiveness in processing time-dependent data and capturing temporal dependencies across sequences.

Table 2. Hyperparameters For LSTM and RNN Architectures

Hyperparameter	Value
Optimizer	Adam
Neuron	32,56, 64, 100, 200
Batch size	56
Learning Rate	0.01, 0.001, 0.0001
Epoch	100, 200, 500
Layer	1
Dropout	0.2

Table 2 presents the hyperparameters used for LSTM and RNN models. The Adam optimizer was chosen for its efficiency in convergence. To test model complexity, hidden layer sizes of 32, 56, 64, 100, and 200 neurons were explored. A batch size of 56 was used to balance stability and computation. The learning rate was tuned at 0.01, 0.001, and 0.0001, and training was run for 100, 200, and 500 epochs. Both models used one hidden layer to reduce overfitting. A dropout rate of 0.2 was applied only to the LSTM model to enhance generalization.

Jurnal Teknik Informatika (JUTIF) Vol. 6, No. 4, August 2025, Page. 2331-2347
P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5057

3.4.2. LSTM Model Results

In this study, The LSTM architecture consists of a single hidden layer and a single output layer. The model's parameters are optimized using the Adam algorithm, which is recognized for its ability to dynamically adjust the learning rate throughout the training process. During the tuning phase, various combinations of hyperparameters are evaluated to determine the best-performing configuration. The optimal hyperparameter settings used in the final training stage of the LSTM model are listed in the table below.

Table 3 outlines the optimal hyperparameter setup for the LSTM configuration, incorporating the Adam optimizer and a 0.001 learning rate, 32 hidden layer neurons, and 500 training epochs. Using this configuration, the LSTM model achieved a RMSE of 180.752, a MAPE of 0.00943, and MSE of 32,671.2844. These relatively low error values demonstrate that the model performs effectively in predicting the test data with high accuracy.

Table 3. Optimal Hyperparameter Configuration For the LSTM Model

Model	**Hyperparameter**				**RMSE**	**MAPE**	**MSE**
	Optimizer	**Learning Rate**	**Neuron**	**Epoch**			
LSTM	Adam	0.001	32	500	180.752	0.00943	32671.2844

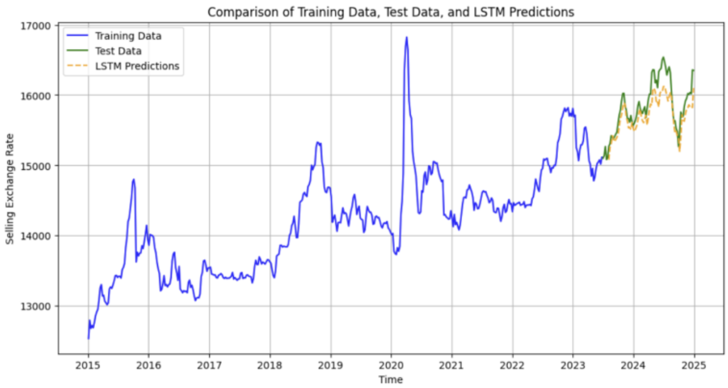

Figure 7. Comparison Of Training Data, Test Data, and LSTM Predictions

Figure 7 presents a comparison between the training data, test data, and the predicted values generated by the LSTM model. The training data from 2015 to mid-2023 show considerable fluctuations, including a significant spike around early 2020, likely caused by global disruptions such as the COVID-19 pandemic. This confirms the volatile and non-stationary nature of the IDR/USD exchange rate. The test data, covering the second half of 2023 to 2024, continue the upward trend with short-term volatility. The LSTM model effectively follows the general trend in the test data, particularly in capturing local peaks and troughs. Although there are minor discrepancies such as slight overestimations or underestimations during sudden shifts the overall prediction closely mirrors the actual pattern. This indicates that the LSTM is capable of learning complex time series behavior, making it suitable for medium-term forecasting.

3.4.3. RNN Model Results

The RNN model applied in this research features a structure with two processing layers and a single output layer. Similar to the LSTM model, parameter optimization is carried out using the Adam algorithm. Hyperparameter tuning is conducted to determine the optimal parameter combination that yields the best predictive performance for time series data. The table below presents the optimal hyperparameter configuration applied during the training of the RNN model.

Table 4. Optimal Hyperparameter Configuration For the RNN Model

Model	Hyperparameter				RMSE	MAPE	MSE
	Optimizer	Learning Rate	Neuron	Epoch			
LSTM	Adam	0.001	56	200	129.082	0.63%	16662.251

Table 4 shows the best hyperparameter configuration for the RNN model, which uses the Adam optimizer, learning rate of 0.001, 56 neurons, and 200 epochs. Under this setup, the RNN achieved an RMSE of 129.082, MAPE of 0.63%, and MSE of 16,662.251. Compared to LSTM, RNN delivered slightly lower predictive accuracy with marginally higher error values.

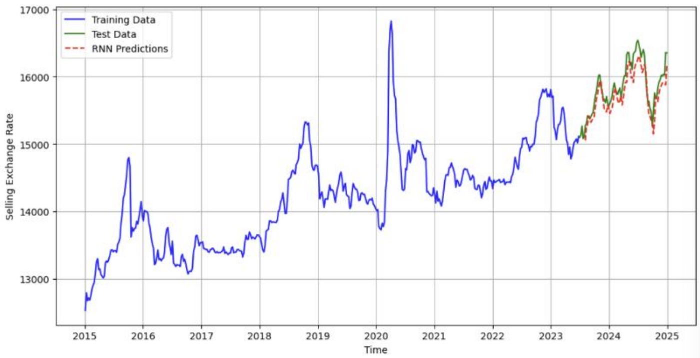

Figure 8. Comparison Of Training Data, Test Data, and RNN Predictions

Figure 8 shows the comparison between the training data, test data, and the predicted values from the RNN model. The training data from 2015 to mid-2023 exhibit high volatility with several significant peaks most notably around early 2020 reflecting the impact of major global events such as the COVID-19 pandemic. In the test period (mid-2023 to 2024), the IDR/USD exchange rate shows an overall upward trend with short-term fluctuations that challenge predictive stability.

The RNN model manages to capture the general direction of the test data but shows larger deviations than the LSTM model, particularly during periods with sharp reversals or high volatility (e.g., early and mid-2024). In some segments, the RNN predictions lag slightly or miss the intensity of the fluctuations, suggesting that RNN struggles with long-range dependencies. This highlights the model's limitations in learning from complex, nonlinear time series, especially when compared to deeper recurrent architectures like LSTM.

3.5. Level-1 Model (Ensemble Stacking)

The stacking ensemble model outperformed the individual RNN and LSTM models, demonstrating superior forecasting accuracy. From the evaluation on the test dataset, it achieved an RMSE of 117.91, which is lower than that of the RNN (129.08) and LSTM (180.75), along with a MAPE of 0.01 and MSE of 13,901.67. These results indicate the ensemble approach's effectiveness in capturing the complex, volatile behavior of the IDR/USD exchange rate.

In this section, the level-1 model also referred to as the ensemble stacking model—is presented as the final stage in the predictive framework. The stacking ensemble was built by integrating the output predictions of RNN and LSTM using Linear Regression as the meta-learner. This design allows the model to combine the strengths of both base learners, improving overall robustness and generalization. Table 5 above displays the predicted selling exchange rate values for the next 20 periods based on the stacking ensemble model.

Table 5 shows that the forecasted exchange rates exhibit a gradual upward trend over the 20 forecast periods. This movement reflects a stable pattern and is consistent with the historical trend

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 4, August 2025, Page. 2331-2347
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5057

previously observed in the data analysis. The predictive output suggests that the model anticipates a steady increase in the exchange rate in the upcoming months.

Table 5 Forecast for the Next 20 Periods Using the Best Model

No	**Forecast Date**	**Predicted Selling Rate**	No	**Forecast Date**	**Predicted Selling Rate**
1	2025-01-05	16342.1477	11	2025-03-16	16772.6179
2	2025-01-12	16389.1968	12	2025-03-23	16811.0898
3	2025-01-19	16435.3084	13	2025-03-30	16848.7951
4	2025-01-26	16480.5012	14	2025-04-06	16885.7492
5	2025-02-02	16524.794	15	2025-04-13	16921.9669
6	2025-02-09	16568.2034	16	2025-04-20	16957.4630
7	2025-02-16	16610.748	17	2025-04-27	16992.2519
8	2025-02-23	16652.4454	18	2025-05-04	17026.3476
9	2025-03-02	16693.3118	19	2025-05-11	17059.7639
10	2025-03-09	16733.3638	20	2025-05-18	17092.5143

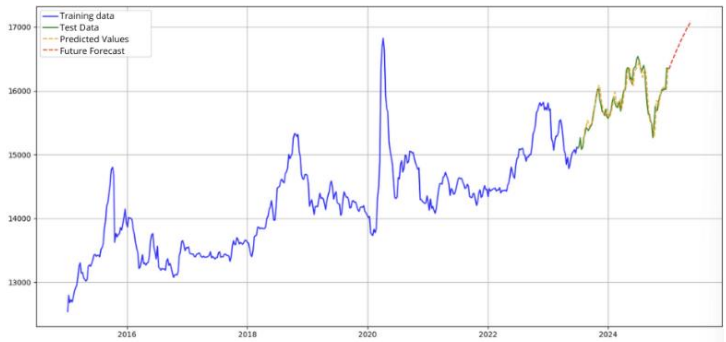

Figure 9. Comparison of Training, Test, Ensemble Predictions, and Forecast

Figure 9 displays the comparison between training data, test data, stacking ensemble predictions, and the 20-period ahead forecast. The historical trend from 2015 to mid-2023 shows high volatility with a sharp spike around 2020, followed by a gradual upward movement. The test data (mid-2023 to 2024) continue this upward trend with noticeable fluctuations. The ensemble predictions closely follow the actual test data, capturing both trend direction and local variability more accurately than individual models.

What stands out is the forecast segment, which extends into early 2025. The stacking ensemble produces a smooth and upward-sloping forecast, consistent with the recent trend in the test data. This indicates that the ensemble model effectively generalizes the temporal pattern and minimizes overfitting. Compared to RNN and LSTM individually, the stacking approach shows better stability and alignment, especially in volatile periods, suggesting its strength in integrating short-term and long-term dependencies for more reliable forecasting.

4. DISCUSSIONS

This study demonstrates the effectiveness of a stacking ensemble model that integrates RNN and LSTM networks for forecasting time series data characterized by high volatility and nonlinear patterns, as exemplified by the weekly IDR/USD exchange rate. The results indicate that while RNN models are adept at capturing short-term temporal dependencies and exhibit relatively fast training times, they fall short when modeling long-term relationships due to the vanishing gradient problem. In contrast, LSTM models excel at learning long-term dependencies and nonlinear dynamics but tend to require more computational resources and longer training durations. The combination of these two architectures

P-ISSN: 2723-3863
https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5057

through a stacking ensemble approach enables the model to exploit both short- and long-term structures in the data.

The performance evaluation reveals that the stacking ensemble model consistently outperforms both individual models, achieving lower RMSE, MSE, and MAPE values. This aligns with the fundamental principle of ensemble learning, which seeks to improve predictive accuracy and generalization by integrating multiple base learners. The effectiveness of stacking in this study is supported by [42], who found that stacking ensembles were particularly effective when the base models had similar predictive performance, outperforming both weighted averaging and gradient boosting approaches in daily time series forecasting. This demonstrates stacking's unique strength in adaptively combining base models, which is especially useful in handling complex, nonlinear, and non-stationary time series data. The use of a linear regression meta-learner in the ensemble further contributes to its robustness by reducing overfitting and enhancing stability in forecast outputs.

The practical implications of this research are significant. Financial institutions, policymakers, and analysts dealing with exchange rate prediction and other volatile economic indicators can benefit from the proposed method, which offers improved accuracy and adaptability. Beyond its applied value, this study contributes to the field of computer science by advancing ensemble deep learning techniques demonstrating the efficacy of stacking RNN and LSTM architectures for forecasting highly volatile and nonlinear time series data. This hybrid framework not only enhances predictive performance but also offers a scalable and generalizable architecture for other complex forecasting tasks in computational finance and related domains. In conclusion, the study validates that a stacking ensemble of RNN and LSTM outperforms individual models, offering a robust solution for time series modeling and a promising direction for future research in intelligent forecasting systems.

5. CONCLUSION

This study evaluated the effectiveness of a stacking ensemble model that integrates LSTM and RNN for forecasting nonlinear and volatile time series data. The results showed that the ensemble outperformed individual models, achieving lower RMSE (117.91), MAPE (0.01), and MSE (13,901.67). The ensemble successfully combined LSTM's strength in capturing long-term patterns with RNN's responsiveness to short-term variations, leading to more accurate and stable forecasts. These findings suggest that the proposed model is well-suited for real-world forecasting tasks in highly dynamic environments such as financial markets. Beyond practical implications, the study advances ensemble deep learning techniques in computer science by demonstrating an adaptive, scalable framework for complex sequential data. Future research may explore alternative meta-learners, incorporate external features, or apply the approach to other domains like energy, climate, or health forecasting.

REFERENCES

[1] N. O. Syamsiah and I. Purwandani, "Penerapan Metode Stacking Untuk Meningkatkan Akurasi Hasil Peramalan Konsumsi Listrik," *Journal of System and Computer Engineering (JSCE) ISSN*, vol. 4, no. 1, pp. 15–25, Jan. 2023, doi: https://doi.org/10.47650/jsce.v4i1.665.

[2] K. A. Notodiputro, Y. Angraini, and L. N. A. Mualifah, *Analisis Data Deret Waktu dengan Python*, 1st ed. Bogor: PT Penerbit IPB Pres, 2025.

[3] P. Das and S. Barman, "Perspective Chapter: An Overview of Time Series Decomposition and Its Applications," *Applied and Theoretical Econometrics and Financial Crises*, pp. 1–15, Feb. 2025, doi: 10.5772/intechopen.1009268.

[4] J. Hao and F. Liu, "Improving long-term multivariate time series forecasting with a seasonal-trend decomposition-based 2-dimensional temporal convolution dense network," *Sci Rep*, vol. 14, no. 1, pp. 1–13, Dec. 2024, doi: 10.1038/s41598-024-52240-y.

Vol. 6, No. 4, August 2025, Page. 2331-2347
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5057

[5] A. Casolaro, V. Capone, G. Iannuzzo, and F. Camastra, "Deep Learning for Time Series Forecasting: Advances and Open Problems," Nov. 01, 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/info14110598.

[6] S. B. Chudo and G. Terdik, "Modeling and Forecasting Time-Series Data with Multiple Seasonal Periods Using Periodograms," *Econometrics*, vol. 13, no. 2, p. 14, Mar. 2025, doi: 10.3390/econometrics13020014.

[7] M. Waqas and U. W. Humphries, "A critical review of RNN and LSTM variants in hydrological time series predictions," *MethodsX*, vol. 13, p. 102946, 2024, doi: https://doi.org/10.1016/j.mex.2024.102946.

[8] F. F. Mojtahedi, N. Yousefpour, S. H. Chow, and M. Cassidy, "Deep Learning for Time Series Forecasting: Review and Applications in Geotechnics and Geosciences," *Computational Methods in Engineering*, pp. 1–31, Aug. 2025, doi: 10.1007/s11831-025-10244-5.

[9] I. Cahya and T. Ningsih, "Sistem Prediksi Berbasis Recurrent Neural Network untuk Klasifikasi Penyakit Kanker," *JURNAL ILMU KOMPUTER DAN TEKNOLOGI*, vol. 1, no. 1, pp. 7–11, Jan. 2025, [Online]. Available: https://jurnal.kolabs.id/index.php/JIKT/

[10] I. Farisi, J. Shadiq, W. Priyadi, D. Maulana, and S. F. Gusril, "Penerapan Model Recurrent Neural Network (RNN) untuk Prediksi Curah Hujan Berbasis Data Historis," *INFORMATION SYSTEM FOR EDUCATORS AND PROFESSIONALS*, vol. 9, no. 2, pp. 217–226, 2024.

[11] W. A. Pratiwi, A. F. Rizki, K. A. Notodiputro, Y. Angraini, and L. N. A. Mualifah, "THE COMPARISON OF ARIMA AND RNN FOR FORECASTING GOLD FUTURES CLOSING PRICES," *BAREKENG: Journal of Mathematics and Its Applications*, vol. 19, no. 1, pp. 0397–0406, Mar. 2025.

[12] A. Casolaro, V. Capone, G. Iannuzzo, and F. Camastra, "Deep Learning for Time Series Forecasting: Advances and Open Problems," Nov. 01, 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/info14110598.

[13] B. Ghojogh and A. Ghodsi, "Recurrent Neural Networks and Long Short-Term Memory Networks: Tutorial and Survey," Ithaca, NY, arXiv:2304.11461, Apr. 2023. doi: https://arxiv.org/abs/2304.11461.

[14] G. Shankarlingam and K. Thirupathireddy, "DRAWBACKS OF LSTM ALGORITHM: A CASE STUDY," Dec. 2023. doi: 10.2139/ssrn.5080605.

[15] G. G. Ghiffary, E. D. D. Yanuari, K. A. Notodiputro, Y. Angraini, and L. N. A. Mualifah, "Comparative Performance Of Sarimax And Lstm Model In Predicting Import Quantities Of Milk, Butter, And Eggs," *BAREKENG: J. Math. & App*, vol. 19, no. 1, pp. 407–0418, Mar. 2025, doi: 10.30598/barekengvol19iss1pp0407-0418.

[16] A. V. Septiani, F. M. Afendi, and A. Kurnia, "Perbandingan Metode GARCH, LSTM, GRU, dan CNN pada Peramalan Volatilitas Kurs," *Limits: Journal of Mathematics and Its Applications*, vol. 22, no. 1, pp. 147–167, Mar. 2025, doi: 10.12962/limits.v22i1.3384.

[17] I. Magfirrah, M. Ilma, K. A. Notodiputro, Y. Angraini, and L. N. A. Mualifah, "Comparative Analysis of ARIMA and LSTM for Forecasting Maximum Wind Speed in Kupang City, East Nusa Tenggara," *Jambura Journal of Mathematics*, vol. 6, no. 2, pp. 169–175, Aug. 2024, doi: 10.37905/jjom.v6i2.25834.

[18] Anggi Adrian Danis, Yenni Kurniawati, Nonong Amalita, and Fadhilah Fitri, "Forecasting the Exchange Rate of Yen to Rupiah Using the Long Short-Term Memory Method," *UNP Journal of Statistics and Data Science*, vol. 1, no. 5, pp. 449–455, Nov. 2023, doi: 10.24036/ujsds/vol1-iss5/114.

[19] D. Wang, X. Cui, and D. Niu, "Wind Power Forecasting Based on LSTM Improved by EMD-PCA-RF," *Sustainability (Switzerland)*, vol. 14, no. 12, Jun. 2022, doi: 10.3390/su14127307.

[20] Y. Bai, J. Sun, J. Luo, and ZhangXiaobin, "Forecasting Financial Time Series with Ensemble Learning," *International Symposium on Intelligent Signal Processing and Communication System (ISPACS 2010) December 6-8*, no. 3, pp. 1–4, 2010.

Vol. 6, No. 4, August 2025, Page. 2331-2347
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5057

[21] M. Leutbecher and T. N. Palmer, "Ensemble forecasting," *J Comput Phys*, vol. 227, no. 7, pp. 3515–3539, Mar. 2008, doi: 10.1016/j.jcp.2007.02.014.

[22] I. Zaier, C. Shu, T. B. M. J. Ouarda, O. Seidou, and F. Chebana, "Estimation of ice thickness on lakes using artificial neural network ensembles," *J Hydrol (Amst)*, vol. 383, no. 3–4, pp. 330–340, Mar. 2010, doi: 10.1016/j.jhydrol.2010.01.006.

[23] N. K. Majid, C. Supriyanto, and A. Marjuni, "Peningkatan Keberagaman Data untuk Klasifikasi Penyakit Diabetes Berbasis Stacking Ensemble Learning," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 10, no. 1, pp. 1–10, Jan. 2025, doi: 10.30591/jpit.v10i1.7375.

[24] E. Wibowo and I. Pratama, "Analisis Sentimen Terhadap Ulasan Hotel Melalui Platform Google Review Menggunakan Metode Stacking," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 6, no. 4, pp. 774–784, Oct. 2024, doi: 10.47233/jteksis.v6i4.1475.

[25] M. N. Azimi, M. M. Rahman, and T. Maraseni, "Advancing sustainable development goal 8 Targets: The role of institutional Quality, economic Complexity, and state fragility in G20 nations (2000–2023)," *Research in Globalization*, vol. 10, Jun. 2025, doi: 10.1016/j.resglo.2025.100278.

[26] H. Kreinin and E. Aigner, "From 'Decent work and economic growth' to 'Sustainable work and economic degrowth': a new framework for SDG 8," *Empirica*, vol. 49, no. 2, pp. 281–311, May 2022, doi: 10.1007/s10663-021-09526-5.

[27] C. Alkahfi, A. Kurnia, and A. Saefuddin, "Perbandingan Kinerja Model Berbasis RNN pada Peramalan Data Ekonomi dan Keuangan Indonesia," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 4, pp. 1235–1243, Jul. 2024, doi: 10.57152/malcom.v4i4.1415.

[28] A. Waladi, Y. Perdana, U. Jambi, F. Sains dan Teknologi, and N. Raisa Hanum, "Stock Price Prediction Using Machine Learning-Based on RNN Algorithms," *Media Journal of Information System and Informatics*, vol. 1, no. 1, pp. 1–8, Nov. 2024, [Online]. Available: http://www.mase.or.id

[29] R. Achmad, Y. Tokoro, J. Haurissa, and A. Wijanarko, "Recurrent Neural Network-Gated Recurrent Unit for Indonesia-Sentani Papua Machine Translation," *Journal of Information Systems and Informatics*, vol. 5, no. 4, pp. 1449–1460, Dec. 2023, doi: 10.51519/journalisi.v5i4.597.

[30] L. Larasati, S. Saadah, and P. E. Yunanto, "Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) Methods to Forecast Daily Turnover at BM Motor Ngawi," *Indonesian Journal of Artificial Intelligence and Data Mining*, vol. 7, no. 1, p. 141, Jan. 2024, doi: 10.24014/ijaidm.v7i1.27643.

[31] A. A. Ningrum, I. Syarif, A. I. Gunawan, E. Satriyanto, and R. Muchtar, "Algoritma Deep Learning-LSTM untuk Memprediksi Umur Transformator," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 8, no. 3, pp. 539–548, Jun. 2021, doi: 10.25126/jtiik.2021834587.

[32] R. Firdaus, H. Mukhtar, and Awaluddin, "Prediksi Indeks Harga Produsen Pertanian Karet Di Indonesia Menggunakan Metode LSTM," *Jurnal Fasilkom*, vol. 13, no. 1, pp. 01–06, Apr. 2023, [Online]. Available: https://www.bappebti.go.id/.

[33] M. Hussein and Y. Azhar, "Prediksi Harga Minyak Dunia Dengan Metode Deep Learning," *Fountain of Informatics Journal*, vol. 6, no. 1, pp. 2548–5113, 2021, doi: 10.21111/fij.v6i1.4446.

[34] G. Budiprasetyo, M. Hani'ah, and D. Z. Aflah, "Prediksi Harga Saham Syariah Menggunakan Algoritma Long Short-Term Memory (LSTM)," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 8, no. 3, pp. 164–172, Jan. 2023, doi: 10.25077/teknosi.v8i3.2022.164-172.

[35] R. Arum Pitaloka and R. Rahmawati, "PERBANDINGAN METODE ARIMA BOX-JENKINS DENGAN ARIMA ENSEMBLE PADA PERAMALAN NILAI IMPOR PROVINSI JAWA TENGAH," *Jurnal Gaussian*, vol. 8, no. 2, pp. 194–207, 2019, [Online]. Available: http://ejournal3.undip.ac.id/index.php/gaussian

Vol. 6, No. 4, August 2025, Page. 2331-2347
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.4.5057

[36] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992, doi: 10.1016/S0893-6080(05)80023-1.

[37] N. K. Majid, C. Supriyanto, and A. Marjuni, "Peningkatan Keberagaman Data untuk Klasifikasi Penyakit Diabetes Berbasis Stacking Ensemble Learning," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 10, no. 1, pp. 1–10, Jan. 2025, doi: 10.30591/jpit.v10i1.7375.

[38] R. Gao, M. Hu, R. Li, X. Luo, P. N. Suganthan, and M. Tanveer, "Stacked Ensemble Deep Random Vector Functional Link Network With Residual Learning for Medium-Scale Time-Series Forecasting," *IEEE Trans Neural Netw Learn Syst*, vol. 36, no. 6, pp. 10833–10843, 2025, doi: 10.1109/TNNLS.2025.3529219.

[39] W. N. Ismail, H. A. Alsalamah, and E. Mohamed, "GA-Stacking: A New Stacking-Based Ensemble Learning Method to Forecast the COVID-19 Outbreak," *Computers, Materials and Continua*, vol. 74, no. 2, pp. 3945–3976, 2023, doi: 10.32604/cmc.2023.031194.

[40] S. Lin and Y. Feng, "Research on Stock Price Prediction Based on Orthogonal Gaussian Basis Function Expansion and Pearson Correlation Coefficient Weighted LSTM Neural Network," *Advances in Computer, Signals and Systems*, vol. 6, no. 5, 2022, doi: 10.23977/acss.2022.060504.

[41] B. Long, F. Tan, and M. Newman, "Forecasting the Monkeypox Outbreak Using ARIMA, Prophet, NeuralProphet, and LSTM Models in the United States," *Forecasting*, vol. 5, no. 1, pp. 127–137, Mar. 2023, doi: 10.3390/forecast5010005.

[42] P. Cawood and T. van Zyl, "Evaluating State of the Art, Forecasting Ensembles- and Meta-learning Strategies for Model Fusion," Jul. 2022, [Online]. Available: http://arxiv.org/abs/2203.03279