Scientific Journal of Informatics
Vol. 11, No. 1, Feb 2024

p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Comparative Analysis of LSTM Neural Network and SVM for USD Exchange Rate Prediction: A Study on Different Training Data Scenarios

Yesy Diah Rosita[1]*, Lady Silk Moonlight[2]
[1]Study Program of Informatics Engineering, Faculty of Informatics, Institut Teknologi Telkom Purwokerto, Indonesia
[2]Aeronautical Communication, Politeknik Penerbangan Surabaya, Indonesia

Abstract.
Purpose: This paper aims to investigate and compare the performance of LSTM Neural Network and Support Vector Machines (SVM) in predicting the USD exchange rate using three different training data scenarios: 45%, 55%, and 75%. The study employs a dataset from the Indonesian Central Bureau of Statistics (BPS) for the period of January 1 to June 30, 2021, encompassing attributes USD Selling Rate.
Methods: The methods involve implementing LSTM and SVM algorithms within the Python programming language using Google Colaboratory. Three distinct training data scenarios are explored to evaluate the models' robustness. Performance metrics, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared, are employed for evaluation.
Result: Results reveal that LSTM demonstrates superior prediction accuracy compared to SVM across all scenarios, even though it incurs a longer training time. Notably, in the 75% training data scenario, LSTM achieves an MAE of 49.52, RMSE of 63.08, and R-squared of 0.37906, outperforming SVM with MAE of 138.33, RMSE of 161.58, and R-squared of 0.34277.
Novelty: This study innovatively compares LSTM Neural Network and Support Vector Machines (SVM) for USD exchange rate prediction across different training scenarios (45%, 55%, and 75%). Unlike previous research focusing on individual models, this study systematically evaluates both methods, highlighting the nuanced balance between prediction accuracy and training time. The findings offer novel insights into LSTM and SVM applicability in currency forecasting, providing valuable guidance for researchers and practitioners in model selection based on specific predictive task requirements.

Keywords: Predictive modeling, Time series forecasting, Financial data analysis
Received December 2023 / **Revised** February 2024 / **Accepted** February 2024

INTRODUCTION

In the rapidly evolving landscape of economic globalization, currency exchange rates emerge as pivotal determinants of a country's economic dynamics [1], [2]. Policy decisions, industrial competitiveness, and investment strategies hinge significantly upon the fluctuations in currency values, notably the United States Dollar (USD), which serves as the global reserve currency, exerting substantial influence over the global financial market [3], [4]. Consequently, the development of predictive models that not only accurately forecast currency movements but also adapt to the dynamic nature of market conditions becomes imperative to facilitate informed decision-making among market participants, investors, and financial institutions [5], [6].

However, the landscape of predictive modeling for currency exchange rates is vast and diverse, encompassing various methodologies ranging from traditional time series models to advanced machine learning techniques. Despite this diversity, there remains a lack of consensus on the most effective approach, particularly in addressing the challenges posed by the dynamic nature of financial markets. Hence, a comprehensive review of existing literature and an identification of research gaps are essential to guide the selection of appropriate predictive models.

*Corresponding author.
Email addresses: yesy.diah@ittelkom-pwt.ac.id (Rosita)
DOI: 10.15294/sji.v11i1.49975

Against this backdrop, this research aims to address the aforementioned gap by focusing on two prominent predictive models: the LSTM (Long Short-Term Memory) neural network and the SVM (Support Vector Machine). These models are chosen for their distinct capabilities and strengths in handling time series data and capturing complex patterns. The LSTM neural network, a type of recurrent neural network, is renowned for its ability to model long-term dependencies and intricate temporal relationships [7], [8]. Conversely, the SVM, initially developed for classification tasks, has proven to be proficient in regression problems and excels in capturing nonlinear patterns within high-dimensional data [9], [10]. While both models offer promising avenues for predicting exchange rates, it is imperative to acknowledge their respective limitations. The LSTM neural network, despite its ability to capture long-term dependencies, may suffer from vanishing or exploding gradients, leading to challenges in training with prolonged sequences of data. Similarly, the SVM, while effective in capturing nonlinear patterns, may struggle with large datasets and requires careful selection of appropriate kernel functions to optimize performance.

Through a rigorous examination of the strengths and weaknesses of the LSTM neural network and SVM, this research seeks to provide a nuanced understanding of their applicability in predicting exchange rates. By conducting controlled empirical experiments and scrutinizing relevant literature, this study endeavors to offer insights into the comparative performance of these models under varying market conditions. Ultimately, the findings of this research are anticipated to inform decision-makers in selecting suitable predictive models and contribute to advancing both theoretical understanding and practical applications in navigating the complexities of currency exchange rate dynamics within the global market [4], [6].

METHODS

A more thorough literature review is warranted to provide a comprehensive understanding of the research landscape and to identify specific gaps in knowledge. Previous studies in the field of currency exchange rate prediction have addressed various predictive models, including autoregressive integrated moving average (ARIMA), neural networks, support vector machines (SVM), and long short-term memory (LSTM) networks, among others [11], [12], [13], [14], [15], [16], [17], [18], [19]. However, a significant gap exists in the comparative analysis of different models, particularly in the context of dynamic financial markets characterized by rapid fluctuations and complex interactions [20].

This research aims to address this gap by conducting a comprehensive comparative analysis of two prominent predictive models: the LSTM neural network and the SVM. While previous studies have explored the predictive capabilities of individual models, the novelty of this research lies in its comparative approach, which offers a more robust framework for model evaluation and selection [21], [22], [23].

Furthermore, by leveraging insights from previous studies on the limitations and challenges associated with each model, this research aims to identify strategies for enhancing predictive accuracy and adaptability. Through a systematic analysis of empirical data and rigorous statistical methods, this study endeavors to contribute to advancing both theoretical understanding and practical applications in the field of currency exchange rate forecasting [24], [25].

The comparison of the LSTM neural network and SVM models represents a novel contribution to the literature, as previous research has primarily focused on the performance of individual models without direct comparison [26], [27], [28]. By conducting a head-to-head comparison of these two models, this research seeks to provide valuable insights into their relative strengths and weaknesses, thus offering a more informed basis for model selection in currency exchange rate prediction.

In summary, while existing literature has provided valuable insights into the predictive performance of individual models, a comprehensive comparative analysis remains lacking. This research aims to fill this gap by providing a rigorous evaluation of the LSTM neural network and SVM, thus offering valuable insights for informed decision-making in the dynamic realm of currency exchange rate prediction.

The LSTM Neural Network is specifically designed to handle time series data with long-term memory, enabling an understanding of complex temporal patterns [7], [8]. In the context of predicting the USD selling value involving historical data, the ability of LSTM to capture long-term relationships can provide an advantage in handling exchange rate fluctuations associated with historical factors. SVM, although initially known in the context of classification, is also capable of addressing regression problems. The strength of SVM lies in its ability to handle high-dimensional data and find the best hyperplane to separate classes or model regression relationships [9], [10], [29]. In the context of exchange rate prediction, the reliability of SVM in handling regression problems can provide an interesting contrast to the recurrent LSTM approach.

LSTM has the ability to recognize more complex temporal patterns due to its recursive structure [7], [30]. This model can understand and leverage long-term historical information, which is crucial in dealing with dynamic changes in currency exchange rates. SVM has good generalization ability and can provide reliable solutions even for complex data. Additionally, SVM is often considered more interpretable, providing clarity in understanding how the model makes decisions or determines relationships between variables [9], [31].

The use of LSTM and SVM in the context of exchange rate prediction can provide advantages. Combining the recursive capabilities of LSTM in capturing long-term temporal patterns with the regression and classification capabilities of SVM can result in a robust and reliable model. Both methods are chosen due to their relevance to specific challenges associated with exchange rate prediction. In a rapidly changing financial market environment, LSTM and SVM represent two different approaches and can provide valuable insights into the performance of models in dynamic situations.

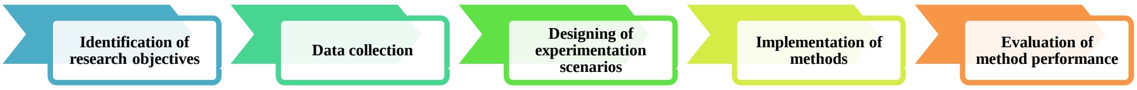

Identification of research objectives
Data collection
Designing of experimentation scenarios
Implementation of methods
Evaluation of method performance

Figure 1. The stages of the research

Identification of research objectives

The primary objective of this research is to compare the performance of two prediction methods, namely Long Short-Term Memory (LSTM) Neural Network and Support Vector Machine (SVM), in forecasting the selling value of the United States Dollar (USD). The main focus of performance measurement is the USD selling attribute. This study aims to provide in-depth insights into the capabilities of both methods in anticipating changes in currency exchange rates.

Data collection

The data used in this research is sourced from the Central Statistics Agency (Badan Pusat Statistik or BPS) and covers the time period from January 1 to June 30, 2021. The dataset consists of several attributes, including Date, USD Selling, Inflation, Money Supply (in Billion Rupiah), Import Value (in Million Dollars), and Export Value (in Million Dollars). This data will serve as the foundation for evaluating the performance of the LSTM Neural Network and SVM prediction methods.

Designing of experimentation scenarios

Before implementing the prediction methods, a scenario testing plan will be formulated. This involves selecting parameters for LSTM and SVM. In the scenario testing plan phase, this research will implement three different scenarios for the division of data between training and testing data. These scenarios were informed by best practices in machine learning experimentation [32], [33]. The division of data into these two sets has a significant impact on the formation of predictive models, and therefore, variations need to be made to understand the performance of the methods comprehensively.

In the LSTM neural network method in this study, there is an architecture specification defined in Table-1, which consists of the LSTM model architecture specifications, model compilation, and early stopping.

Table 1. The architecture of the LSTM neural network method (data source: compiled by authors)

Categories	Descriptions
Model Architecture Specifications	▪ Layer: LSTM ▪ Dense (Output Layer): 1 ▪ Number of neuron units: 50 ▪ Activation Function: ReLU ▪ Input Shape: (X_train.shape[1], 1)
Model Compilation Configuration	▪ Optimizer: Adam ▪ Loss Function: Mean Squared Error (MSE)
Early Stopping Configuration	▪ Monitor: val_loss ▪ Patience: 5 ▪ Restore Best Weights: True

This LSTM is a specialized type of layer in recurrent neural networks (RNN) designed to handle long-term dependency issues. LSTM is used to understand temporal patterns in time series data [7], [8]. On the other hand, the Dense (Output Layer) with 1 neuron is a fully connected layer that summarizes the results from the LSTM layer into a single output value. With one neuron, it is suitable for regression tasks to predict a single numerical value, as this is a regression task to predict currency exchange rates, and one unit is sufficient to generate one output value. This LSTM layer has 50 neuron units that determine the model's complexity and its ability to capture patterns in time series data. The ReLU activation function introduces non-linearity to the LSTM layer, allowing the model to learn from more complex data. The input shape accepted by this LSTM layer indicates the number of timesteps and features at each timestep. This aligns with the structure of the time series data used in the model.

The model compilation process is a necessary step before training or evaluating a model in deep learning. Compilation configures the model by selecting the optimizer, loss function, and evaluation metrics. Some key elements of model compilation are:

1. Optimizer: The optimization algorithm that will be used to adjust the model weights based on the training data. 'Adam' is one of the commonly used optimizers because it is efficient and effective.
2. Loss Function: The loss function that will be minimized during the training process. For regression problems, such as predicting currency exchange rates in this case, Mean Squared Error (MSE) is often used.
3. Evaluation Metric (Optional): Metrics that will be evaluated during training and testing. While optional, these metrics provide additional insights into the model's performance.

Early Stopping is a technique used during model training to prevent overfitting and expedite the training process. The main function of Early Stopping is:

1. Monitor: Monitor a specific metric on the validation data, in this case, 'val_loss' or the loss on the validation data. If there is no improvement in that metric, the training will be stopped.
2. Patience: Indicate how long (how many epochs) we are willing to wait without improvement in the monitored metric before stopping the training. This helps prevent the model from learning noise in the training data.
3. Restore Best Weights: If set to True, it will revert the model to the best weights found during training. This helps ensure that we have a model with the best performance even if the training is stopped early.

With model compilation, the model configuration is used for learning data using the appropriate optimizer and loss function. On the other hand, Early Stopping is a strategy to control the training duration and prevent the model from overlearning, which can negatively impact performance on new data.

The basic equations detailing the operations within one Long Short-Term Memory (LSTM) cell are as follows:

1. Forget Gate (f_t)
$$f_t = \sigma\big(W_f \times [h_{t-1}, x_t]\big) + b_f \tag{1}$$
2. Input Gate (i_t)
$$i_t = \sigma(W_i \times [h_{t-1}, x_t]) + b_i \tag{2}$$
3. Candidate Cell State ($\tilde{C}_t$)
$$\tilde{C}_t = tanh(W_C \times [h_{t-1}, x_t]) + b_C \tag{3}$$
4. Cell State (C_t)
$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \tag{4}$$
5. Output Gate (o_t)
$$o_t = \sigma(W_o \times [h_{t-1}, x_t]) + b_o \tag{5}$$
6. Hidden State (h_t)
$$h_t = o_t \times tanh(C_t) \tag{6}$$

Notation:

- h_t : hidden state at time *t*
- x_t : input state at time *t*
- C_t : cell state at time t

- σ : sigmoid functions
- $tanh$: hyperbolic tangent function
- W_f, W_i, W_C, W_o : learned weight values
- b_f, b_i, b_C, b_o : learned bias values

The LSTM utilizes gates to control the information that is stored or ignored in the memory cell. The equations above reflect the specific operations that occur within an LSTM cell during one time step. In each subsequent time step, these values are updated using information from the previous time step and the current input.

Meanwhile, in the SVM method, it is a machine learning algorithm used for classification and regression problems [9], [10], [29]. The SVM model in this research uses Support Vector Regression (SVR) with a linear kernel. SVM can also employ various types of kernels (linear, polynomial, radial basis function, etc.), and in this case, a linear kernel is chosen for specific reasons:

1. A linear kernel has a relatively straightforward interpretation. The decision function produced by SVM with a linear kernel can be explained in the original feature space, aiding in understanding the relationship between features and the target.
2. A linear kernel is suitable when the relationship between features and the target can be approximated linearly. If the data can inherently be described by a straight line, then a linear kernel can provide good results.
3. SVM with a linear kernel is typically computationally more efficient compared to non-linear kernels. Training the model and making predictions with a linear kernel can be faster, especially on large datasets.
4. A linear kernel tends to provide strong regularization to the model. This can help avoid overfitting, especially when the data is limited or tends to be noisy.

For SVM in the context of regression, such as Support Vector Regression (SVR), the underlying equation is as follows:

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b \tag{7}$$

Notation:
$f(x)$: predicted value
α_i : Lagrange coefficients
y_i : target label
$K(x_i, x)$: kernel function that computes the similarity between two vectors x_i and $, x$
b : bias term

For SVM with a linear kernel, the kernel function (K) is calculated as the simple dot product between two feature vectors. For other kernels, such as polynomial or Radial Basis Function (RBF) kernels, the kernel function is adjusted according to the type of kernel used.

Scenario-1 (45:55)

- 45% of the data is utilized as part of the model training process.
- 55% of the data is set aside as test data.
- The selection of this proportion emphasizes training the model to recognize patterns within the previous dataset.

Scenario-2 (55:45)

- The 55%-45% data split aims to achieve a better balance between the training and testing phases.
- With a larger allocation of test data, the model is given the opportunity to assess the generalization of prediction outcomes.

Scenario-3 (75:25)

- The 75-25 scenario gives the highest weight to training data (75%) and reserves 25% for test data.

- This approach emphasizes training the model by providing limited room to test the model on data not involved in the training process.

Implementation of methods

The LSTM Neural Network was implemented using TensorFlow [34] , a widely used deep learning framework, with specific configurations as detailed in Table 1. The choice of TensorFlow aligns with its reputation for efficient handling of sequential data and complex neural network architectures [35]. The SVM implementation utilized the Scikit-learn library [36], a popular machine learning toolkit in Python, known for its ease of use and robust implementation of SVM algorithms [37]. The programming language used was Python, and all experiments were conducted through the Google Colaboratory platform, leveraging the flexibility of cloud computing for efficient analysis and model training. Additionally, tool specifications were a critical factor to ensure the continuity and accuracy of the analysis process. The tool used was Google Colab, a Python-based platform running in the Google Cloud environment. The device specifications used to run Google Colab are presented in Table 2.

Table 2. Specifications of the tools used (data source: compiled by authors)

No.	Components	Description
1	Operating System (OS) Name	Microsoft Windows 11 Home Single Language
2	OS Version	10.0.22621 Build 22621
3	System Manufacturer	AUSTek COMPUTER INC
4	System Model	VivoBook_ASUSLaptop X430FN_S430FN
5	System Type	X64-based PC
6	Processor	Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz, 1800 Mhz, 4 Core(s), 8 Logical Processor(s)
7	BIOS Version/Date	American Megatrends Inc. X430FN.308, 5/28/2019
8	Installed Physical Memory (RAM)	8.00 GB
9	Total Physical Memory	7.85 GB
10	Available Physical Memory	912 MB

Evaluation of method performance

To evaluate the performance of the LSTM Neural Network and SVM, several metrics were employed, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared. These metrics are standard in assessing the accuracy and goodness-of-fit of predictive models [27]. The choice of these metrics was informed by their relevance to regression tasks and their widespread use in evaluating machine learning models [28], [34].

Table 3. Metrics evaluation

No.	Metrics	Formula		Function
1	MAE	$MAE = \frac{1}{n}\sum_{i=1}^{n} \lvert y_i - \hat{y}_i \rvert$	(8)	MAE (Mean Absolute Error) provides information about the average of the absolute differences between predictions and actual values. The lower the MAE value, the better the model performance [38], [39], [40].
2	RMSE	$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \lvert y_i - \hat{y}_i \rvert}$	(9)	RMSE (Root Mean Squared Error) measures the magnitude of the average prediction error in a form consistent with the original data. Like MAE, a lower RMSE value indicates better predictions [38], [39], [40].
3	R-Squared	$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$	(10)	R-squared presents the extent to which the variation in the dependent variable can be explained by the model. Higher values indicate a better model in explaining the data variation [38], [39], [40].

Notation:
$\hat{y}$: estimated value
y : actual values
n : number of data points
i : index of data point

RESULTS AND DISCUSSIONS

This section illustrates and analyzes the experimental results of this research, focusing on the three data split scenarios previously described: 45:55, 55:45, and 75:25. Additionally, the discussion will involve the architectures of the implemented LSTM and SVM methods, as well as the tool specifications used in this study.

The exploration of data split scenarios aims to evaluate the reliability and accuracy of predictions from the LSTM Neural Network and Support Vector Machine (SVM) models in the context of predicting the USD exchange rate. As a complement, this chapter also provides details about the specific architectures used in the LSTM and SVM models, offering a deeper understanding of how they operate in this prediction task.

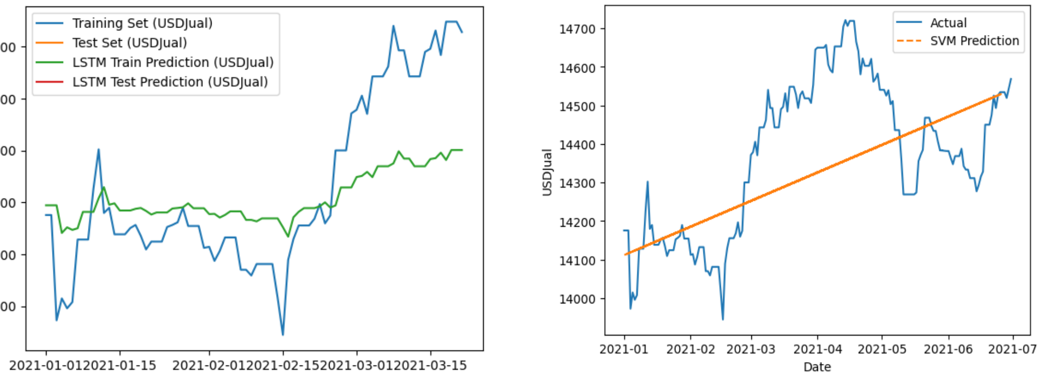
Scenario-1 (45:55)

In the data split scenario with a training proportion of 45%, the experimental results show a performance comparison between the LSTM Neural Network and Support Vector Machines (SVM) models. The time required by the LSTM model is approximately 7.49 seconds, while SVM only takes about 0.43 seconds. These results indicate a significant advantage of SVM in terms of training speed. However, when looking at the performance evaluation metrics, it is evident that the LSTM model shows a Mean Absolute Error (MAE) of 204.64, Root Mean Squared Error (RMSE) of 225.03, and R-Squared of around 1.99. On the other hand, the SVM model has an MAE of 132.50, RMSE of 162.81, and R-Squared of around 0.33 (see Table 4).

Table 4. Performance evaluation results of the methods in scenario-1 (data source: compiled by authors)

No.	Method	Duration (seconds)	MAE	RMSE	R-Squared
1	LSTM	7.48855	204.63687	225.02692	(1.98866)
2	SVM	0.43251	132.49984	162.81235	0.32560

Although the training time of SVM is shorter, the LSTM model appears to produce more accurate predictions based on lower MAE and RMSE values. The R-Squared value approaching 2 for LSTM indicates the model's ability to explain data variation better compared to SVM, which has a relatively low R-Squared value. Therefore, while SVM excels in time efficiency, the LSTM model provides more accurate and reliable results in the context of predicting the USD exchange rate in this data split scenario. In model selection, the trade-off between time and accuracy needs to be carefully considered depending on specific applicative needs (see Figure 1).



(a) (b)

Figure 1. Visualization of the experimental results in scenario-1: (a) LSTM; (b) SVM (data source: compiled by authors)

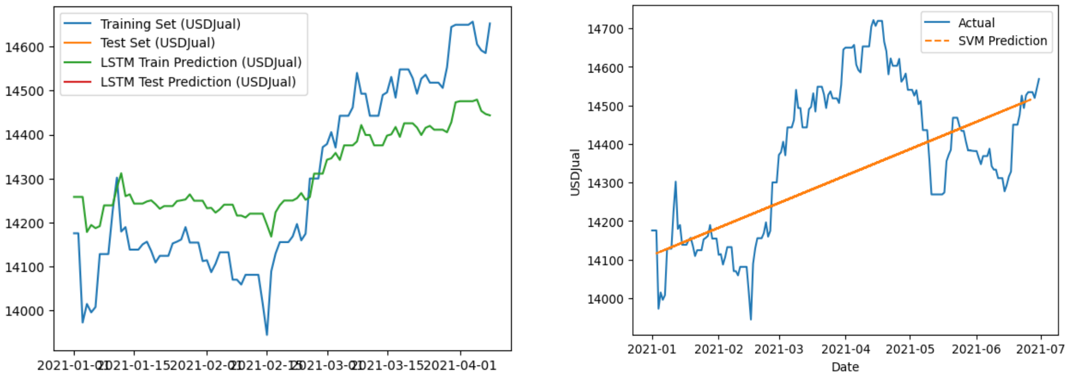
Scenario-2 (55:45)

Experimental results using 55% of the data as training data show a significant difference between the performance of the LSTM Neural Network and Support Vector Machines (SVM) models. The time required by the LSTM Neural Network model (5.73909) is much longer than that of SVM (0.60398), indicating higher complexity in the LSTM model training process. Nevertheless, the performance of the LSTM Neural Network, as measured by Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and coefficient of determination (R-squared), is better than SVM. The lower MAE in LSTM (84.72266) indicates that its predictions have a more minor average error than SVM (134.67929). A similar thing is seen in RMSE, where a lower value in LSTM (104.92248) indicates a better prediction error rate than SVM (168.23663). In addition, the higher coefficient of determination (R-squared) in LSTM (0.35128) suggests that this model is better able to explain variations in test data than SVM (0.28888) (see Table 5).

Table 5. Performance evaluation results of the methods in scenario-2 (data source: compiled by authors)

No.	Metode	Duration (seconds)	MAE	RMSE	R-Squared
1	LSTM	5.73909	84.72266	104.92248	0.35128
2	SVM	0.60398	134.67929	168.23663	0.28888

Despite the longer training time of LSTM, these results illustrate that the model complexity and time invested in LSTM Neural Network can be offset by superior prediction performance compared to SVM in this data-sharing scenario (see Figure 2).



(a) (b)

Figure 2. Visualization of the experimental results in scenario-2: (a) LSTM; (b) SVM (data source: compiled by authors)

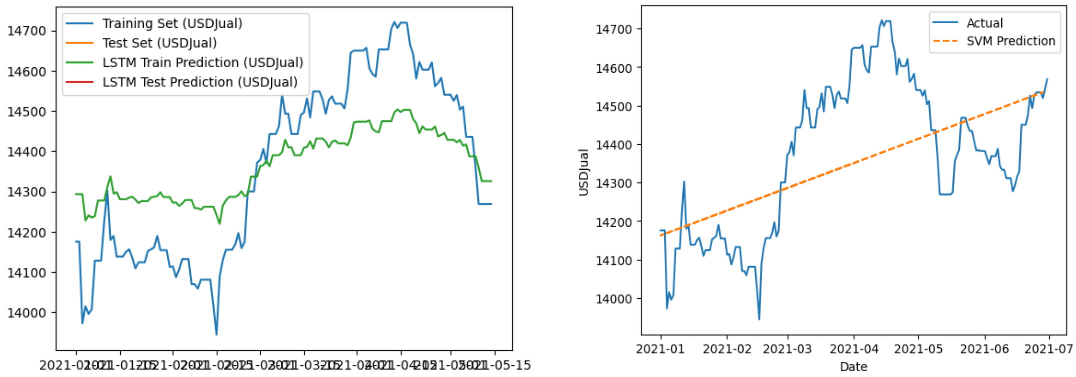
Scenario-3 (75:25)

The results of experiments with a 75% data-sharing scenario for training show a performance comparison between the LSTM Neural Network and Support Vector Machines (SVM) methods in predicting USD currency exchange rates. Regarding the time required to train the model, LSTM Neural Network shows a more significant time, around 11.52 seconds, while SVM only takes 0.30 seconds. The relatively faster time on SVM is attributed to its essential characteristics, which are more computationally efficient, especially when dealing with large data sets. However, keep in mind that faster training times do not necessarily indicate better prediction quality (see Table 6).

Table 6. Performance evaluation results of the methods in scenario-3 (data source: compiled by authors)

No.	Method	Duration (seconds)	MAE	RMSE	R-Squared
1	LSTM	11.51935	49.52471	63.08182	0.37906
2	SVM	0.29688	138.32589	161.58138	0.34277

Regarding performance evaluation, the LSTM Neural Network shows a Mean Absolute Error (MAE) of around 49.52 and a Root Mean Squared Error (RMSE) of about 63.08. On the other hand, SVM has an MAE of about 138.33 and an RMSE of about 161.58. The lower MAE and RMSE values in the LSTM Neural Network indicate that this model tends to provide predictions that are more accurate and close to the true value, especially when compared with the performance of SVM. However, the low R-squared for both, namely 0.37906 for LSTM and 0.34277 for SVM, indicates that neither model thoroughly explains the variation in the data.



(a) (b)

Figure 3. Visualization of the experimental results in scenario-3: (a) LSTM; (b) SVM (data source: compiled by authors)

Critically, these results show that although LSTM requires a longer time for training, it tends to provide more accurate predictions compared to SVM in the context of currency exchange data. Thus, the decision between these two methods must be carefully considered, considering the trade-off between training time and desired prediction quality.

The author should enrich the additional results by considering the implications of the findings for practical applications and further research. For example, the author could discuss how the superior performance of the LSTM Neural Network in predicting the USD exchange rate could benefit financial institutions, investors, and policymakers in making more informed decisions. Additionally, the author could explore potential extensions of the research, such as investigating the performance of other machine learning algorithms or incorporating additional features into the prediction models.

Furthermore, this segment should be enriched with comparisons to other studies to provide context for the results. By comparing the findings to existing literature, the author can assess whether the results are consistent with previous research or if there are any discrepancies that warrant further investigation. This comparative analysis could help validate the robustness of the findings and contribute to the advancement of knowledge in the field.

CONCLUSION

The conclusion drawn from the results of the three data-sharing scenarios for training reveals several significant findings. Firstly, when utilizing 75% of the data for training, the LSTM Neural Network exhibits a longer training time (11.52 seconds) compared to SVM (0.30 seconds). However, the LSTM Neural Network yields predictions with superior performance, with an MAE of approximately 49.52 and an RMSE of about 63.08, whereas SVM shows an MAE of around 138.33 and an RMSE of approximately 161.58. Despite these results, both methods still exhibit a need for higher R-squared values, indicating a partial explanation of the data variation.

Secondly, in the scenario using 55% of the data for training, the LSTM Neural Network maintains commendable performance, with a training time of around 5.74 seconds, an MAE of about 84.72, and an RMSE of 104.92. Despite a faster training time for SVM (0.60 seconds), its prediction accuracy is inferior, with an MAE around 134.68 and an RMSE around 168.24.

Lastly, with 45% of the data used for training, the LSTM Neural Network demonstrates a training time of around 7.49 seconds, accompanied by improved prediction performance compared to the previous scenario, with an MAE around 204.64 and an RMSE around 225.03. SVM, on the other hand, requires a very short training time (0.43 seconds), but its prediction performance remains suboptimal, with an MAE of approximately 132.50 and an RMSE of around 162.81.

The practical implications of these findings are significant for the finance sector. By demonstrating the superior predictive performance of the LSTM Neural Network, this research provides valuable insights for currency traders, financial institutions, and policymakers. Accurate exchange rate forecasts enable traders to make informed decisions and mitigate risks, while financial institutions can optimize their hedging strategies and minimize exposure to currency fluctuations. Policymakers can utilize these forecasts to formulate effective monetary policies and stabilize currency markets.

Furthermore, the research contributes to the advancement of predictive modeling techniques in finance. By showcasing the effectiveness of the LSTM Neural Network, the study encourages further exploration and development of deep learning algorithms for financial forecasting. Future research could explore the integration of additional data sources and refining model architectures to enhance predictive accuracy further.

In conclusion, the findings of this research offer practical benefits for decision-making and risk management in the finance sector. By providing accurate exchange rate predictions, the LSTM Neural Network presents a valuable tool for navigating the complexities of global financial markets. As such, this research contributes to the ongoing development and innovation in financial forecasting methodologies.

REFERENCES

[1] G. Pan, “The Impact of Exchange Rate Fluctuation on Toyota Financial Performance and Its Hedging Strategy,” BCP Business & Management, vol. 43, pp. 270–276, Mar. 2023, doi: 10.54691/bcpbm.v43i.4647.

[2] U. S. Al-Azizah, Y. Daulay, and N. Krisnanto, “The Effect of USD/IDR Exchange Rate, Interest Rate, and World Oil Price To Jakarta Composite Index (JCI),” Agregat: Jurnal Ekonomi dan Bisnis, vol. 3, no. 2, p. 191, Nov. 2019, doi: 10.22236/agregat_vol3/is2pp191-204.

[3] E. Y. Matsumoto, E. Del-Moral-Hernandez, C. E. Yoshinaga, and A. de Campos Pinto, “Forecasting US dollar exchange rate movement with computational models and human behavior,” Expert Syst Appl, vol. 194, p. 116521, May 2022, doi: 10.1016/j.eswa.2022.116521.

[4] F. Ayiah-Mensah, P. Arthur-Nunoo, J. Acquah, and J. A. Addor, “Trend Analysis of Exchange Rate of the Ghana Cedi against the US Dollar Using Time Series,” Open J Stat, vol. 13, no. 05, pp. 734–745, 2023, doi: 10.4236/ojs.2023.135035.

[5] A. Vadivel, S. Veeramani, and C. Raghutla, “Exchange rate (USD/INR) pass-through and wholesale price index: A flexible least square approach,” J Public Aff, vol. 20, no. 3, Aug. 2020, doi: 10.1002/pa.2087.

[6] P. Hasanaj and B. Kuqi, “Analysis of Financial Statements,” Humanities and Social Science Research, vol. 2, no. 2, p. p17, Jun. 2019, doi: 10.30560/hssr.v2n2p17.

[7] E. M. C. Wattimena, A. Annisa, and I. S. Sitanggang, “CO and PM10 Prediction Model based on Air Quality Index Considering Meteorological Factors in DKI Jakarta using LSTM,” Scientific Journal of Informatics, vol. 9, no. 2, pp. 123–132, Oct. 2022, doi: 10.15294/sji.v9i2.33791.

[8] D. Hopp, “Performance of LSTM Neural Networks in Nowcasting during the COVID-19 Crisis,” in UNCTAD Research Paper, 2021. [Online]. Available: https://github.com/dhopp1/NowcastLSTM.jl

[9] W. Sholihah and A. Silvia Handayani, “Revolutionizing Healthcare: Comprehensive Evaluation and Optimization of SVM Kernels for Precise General Health Diagnosis,” Scientific Journal of Informatics, vol. 10, no. 4, p. 445, 2023, doi: 10.15294/sji.v10i4.46430.

[10] A. Fahmi Limas, R. Rosnelly, and A. Nursie, “A Comparative Analysis on the Evaluation of KNN and SVM Algorithms in the Classification of Diabetes,” Scientific Journal of Informatics, vol. 10, no. 3, p. 251, 2023, doi: 10.15294/sji.v10i3.44269.

[11] G. Pan, “The Impact of Exchange Rate Fluctuation on Toyota Financial Performance and Its Hedging Strategy,” BCP Business & Management, vol. 43, pp. 270–276, Mar. 2023, doi: 10.54691/bcpbm.v43i.4647.

[12] U. S. Al-Azizah, Y. Daulay, and N. Krisnanto, “The Effect of USD/IDR Exchange Rate, Interest Rate, and World Oil Price To Jakarta Composite Index (JCI),” Agregat: Jurnal Ekonomi dan Bisnis, vol. 3, no. 2, p. 191, Nov. 2019, doi: 10.22236/agregat_vol3/is2pp191-204.

[13] E. Y. Matsumoto, E. Del-Moral-Hernandez, C. E. Yoshinaga, and A. de Campos Pinto, “Forecasting US dollar exchange rate movement with computational models and human behavior,” Expert Syst Appl, vol. 194, p. 116521, May 2022, doi: 10.1016/j.eswa.2022.116521.

[14] F. Ayiah-Mensah, P. Arthur-Nunoo, J. Acquah, and J. A. Addor, “Trend Analysis of Exchange Rate of the Ghana Cedi against the US Dollar Using Time Series,” Open J Stat, vol. 13, no. 05, pp. 734–745, 2023, doi: 10.4236/ojs.2023.135035.

[15] A. Vadivel, S. Veeramani, and C. Raghutla, “Exchange rate (USD/INR) pass-through and wholesale price index: A flexible least square approach,” J Public Aff, vol. 20, no. 3, Aug. 2020, doi: 10.1002/pa.2087.

[16] P. Hasanaj and B. Kuqi, “Analysis of Financial Statements,” Humanities and Social Science Research, vol. 2, no. 2, p. p17, Jun. 2019, doi: 10.30560/hssr.v2n2p17.

[17] D. Hopp and A. Statistician, “Performance of LSTM Neural Networks in Nowcasting during the COVID-19 Crisis,” 2021. [Online]. Available: https://github.com/dhopp1/NowcastLSTM.jl

[18] L. Setiyani and W. H. Utomo, “Arabica Coffee Price Prediction Using the Long Short Term Memory Network (LSTM) Algorithm,” Scientific Journal of Informatics, vol. 10, no. 3, 2023, doi: 10.15294/sji.v10i3.44162.

[19] B. A. Ardhani, N. Chamidah, and T. Saifudin, “Sentiment Analysis Towards Kartu Prakerja Using Text Mining with Support Vector Machine and Radial Basis Function Kernel,” Journal of Information Systems Engineering and Business Intelligence, vol. 7, no. 2, p. 119, Oct. 2021, doi: 10.20473/jisebi.7.2.119-128.

[20] E. M. C. Wattimena, A. Annisa, and I. S. Sitanggang, “CO and PM10 Prediction Model based on Air Quality Index Considering Meteorological Factors in DKI Jakarta using LSTM,” Scientific Journal of Informatics, vol. 9, no. 2, pp. 123–132, Oct. 2022, doi: 10.15294/sji.v9i2.33791.

[21] W. Sholihah and A. Silvia Handayani, “Revolutionizing Healthcare: Comprehensive Evaluation and Optimization of SVM Kernels for Precise General Health Diagnosis,” Scientific Journal of Informatics, vol. 10, no. 4, p. 445, 2023, doi: 10.15294/sji.v10i4.46430.

[22] A. Fahmi Limas, R. Rosnelly, and A. Nursie, “A Comparative Analysis on the Evaluation of KNN and SVM Algorithms in the Classification of Diabetes,” Scientific Journal of Informatics, vol. 10, no. 3, p. 251, 2023, doi: 10.15294/sji.v10i3.44269.

[23] T. Soni Madhulatha and M. Atheeq, “Foreign Exchange Rates Prediction for Time-series Data using Advanced Q-sensing Model ⋆,” 2022. [Online]. Available: http://ceur-ws.org

[24] T. E. Putri, A. A. Firdaus, and W. I. Sabilla, “Short-Term Forecasting of Electricity Consumption Revenue on Java-Bali Electricity System using Jordan Recurrent Neural Network,” Journal of Information Systems Engineering and Business Intelligence, vol. 4, no. 2, p. 96, Oct. 2018, doi: 10.20473/jisebi.4.2.96-105.

[25] A. Biswas and Md. S. Islam, “A Hybrid Deep CNN-SVM Approach for Brain Tumor Classification,” Journal of Information Systems Engineering and Business Intelligence, vol. 9, no. 1, pp. 1–15, Apr. 2023, doi: 10.20473/jisebi.9.1.1-15.

[26] D. S. K. Karunasingha, “Root mean square error or mean absolute error? Use their ratio as well,” Inf Sci (N Y), vol. 585, pp. 609–629, Mar. 2022, doi: 10.1016/j.ins.2021.11.036.

[27] D. Chicco, M. J. Warrens, and G. Jurman, “The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation,” PeerJ Comput Sci, vol. 7, p. e623, Jul. 2021, doi: 10.7717/peerj-cs.623.

[28] A. G. Priya Varshini, K. Anitha Kumari, D. Janani, and S. Soundariya, “Comparative analysis of Machine learning and Deep learning algorithms for Software Effort Estimation,” J Phys Conf Ser, vol. 1767, no. 1, p. 012019, Feb. 2021, doi: 10.1088/1742-6596/1767/1/012019.

[29] A. Biswas and M. S. Islam, “A Hybrid Deep CNN-SVM Approach for Brain Tumor Classification,” Journal of Information Systems Engineering and Business Intelligence, vol. 9, no. 1, Apr. 2023, doi: 10.20473/jisebi.9.1.1-15.

[30] T. E. Putri, A. A. Firdaus, and W. I. Sabilla, "Short-Term Forecasting of Electricity Consumption Revenue on Java-Bali Electricity System using Jordan Recurrent Neural Network," Journal of Information Systems Engineering and Business Intelligence, vol. 4, no. 2, p. 96, Oct. 2018, doi: 10.20473/jisebi.4.2.96-105.

[31] E. Rainarli and A. Romadhan, "Perbandingan Simple Logistic Classifier dengan Support Vector Machine dalam Memprediksi Kemenangan Atlet," Journal of Information Systems Engineering and Business Intelligence, vol. 3, no. 2, p. 87, Oct. 2017, doi: 10.20473/jisebi.3.2.87-91.

[32] G. James, D. Witten, T. Hastie, and R. Tibshiran, An Introduction to Statistical Learning with Application R, vol. 103. New York: Springer, 2013.

[33] J. Brownlee, Better Deep Learning: Train Faster, Reduce Overfitting, and Make Better Predictions. San Fransisco: machinelearningmastery.com, 2018.

[34] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," in OSDI'16: Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation, Mar. 2016. [Online]. Available: http://arxiv.org/abs/1603.04467

[35] A. Gulli and S. Pal, Deep Learning with Keras. Packt Publishing, 2017.

[36] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, Jan. 2012, [Online]. Available: http://arxiv.org/abs/1201.0490

[37] L. Buitinck et al., "API design for machine learning software: experiences from the scikit-learn project," in European Conference on Machine Learning and Principles and Practices of Knowledge Discovery in Databases, Sep. 2013. [Online]. Available: http://arxiv.org/abs/1309.0238

[38] D. S. K. Karunasingha, "Root mean square error or mean absolute error? Use their ratio as well," Inf Sci (N Y), vol. 585, pp. 609–629, Mar. 2022, doi: 10.1016/j.ins.2021.11.036.

[39] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," PeerJ Comput Sci, vol. 7, p. e623, Jul. 2021, doi: 10.7717/peerj-cs.623.

[40] A. G. Priya Varshini, K. Anitha Kumari, D. Janani, and S. Soundariya, "Comparative analysis of Machine learning and Deep learning algorithms for Software Effort Estimation," J Phys Conf Ser, vol. 1767, no. 1, p. 012019, Feb. 2021, doi: 10.1088/1742-6596/1767/1/012019.