Journal of Quality Measurement and Analysis *JQMA* 21(3) 2025, 191-205
e-ISSN: 2600-8602 *http://www.ukm.my/jqma*
https://doi.org/10.17576/jqma.2103.2025.12

OPTIMISING LSTM AND BILSTM MODELS FOR TIME SERIES FORECASTING THROUGH HYPERPARAMETER TUNING

(*Mengoptimumkan Model LSTM dan BiLSTM untuk Ramalan Siri Masa Melalui Penyesuaian Hiperparameter*)

NUR HAIZUM ABD RAHMAN*, QUAY PIN YIN & HANI SYAHIDA ZULKAFLI

ABSTRACT

Long Short-Term Memory (LSTM) and Bidirectional Long Short-Term Memory (BiLSTM) are the emerging Recurrent Neural Networks (RNN) widely used in time series forecasting. The performance of these neural networks relies on the selection of hyperparameters. A random selection of the hyperparameters may increase the forecasting error. Hence, this study aims to optimise the performance of LSTM and BiLSTM in time series forecasting by tuning one of the essential hyperparameters, the number of hidden neurons. LSTM and BiLSTM with 32, 64, and 128 hidden neurons and various combinations of other hyperparameters are formed in this study through grid searching. The models are evaluated and compared based on the Mean Squared Error (MSE) and Mean Absolute Error (MAE). The results from real data analysis revealed that 128 hidden neurons are the optimum choice of hidden neurons with the lowest error values. This study investigates whether BiLSTM, performs better than LSTM in forecasting. Thus, the performance of these two neural networks in forecasting time series data was compared, and the Wilcoxon-Signed Rank Test was conducted. Results revealed a significant difference in the performance of these two neural networks, and BiLSTM outperformed LSTM in forecasting time series data. Hence, BiLSTM with 128 hidden neurons is encouraged to be chosen over LSTM in time series forecasting. Since these findings have implications for future practice, the combination of model and hyperparameter should be chosen wisely to obtain more accurate predictions in time series forecasting.

Keywords: LSTM; deep learning; time series; forecasting; BiLSTM

ABSTRAK

Memori Jangka Pendek Panjang (LSTM) dan Memori Jangka Pendek Panjang Dwiarah (BiLSTM) ialah Rangkaian Neural Berulang (RNN) digunakan secara meluas dalam peramalan siri masa. Prestasi rangkaian neural ini bergantung pada pemilihan hiperparameter. Pemilihan rawak hiperparameter boleh meningkatkan ralat ramalan. Oleh itu, kajian ini bertujuan untuk mengoptimumkan prestasi LSTM dan BiLSTM dalam peramalan siri masa dengan menala salah satu hiperparameter penting, bilangan neuron tersembunyi. LSTM dan BiLSTM dengan 32, 64, dan 128 neuron tersembunyi dan pelbagai kombinasi hiperparameter lain dibentuk dalam kajian ini melalui carian grid. Model-model tersebut dinilai dan dibandingkan berdasarkan Ralat Purata Kuasa Dua (MSE) dan Ralat Mutlak Purata (MAE). Keputusan daripada analisis data sebenar mendedahkan bahawa 128 neuron tersembunyi adalah pilihan optimum neuron tersembunyi dengan nilai ralat terendah. Kajian ini mengkaji samada BiLSTM memberikan prestasi yang lebih baik berbanding LSTM dalam peramalan. Prestasi kedua-dua rangkaian neural ini dalam peramalan data siri masa dibandingkan, dan Ujian Peringkat Bertanda Wilcoxon (Wilcoxon-Signed Rank Test) dilaksanakan. Hasil kajian menunjukkan perbezaan ketara dalam prestasi kedua-dua rangkaian neural ini, dan BiLSTM mengatasi LSTM dalam peramalan siri masa. Oleh itu, BiLSTM dengan 128 neuron tersembunyi disarankan untuk dipilih berbanding LSTM dalam peramalan siri masa. Memandangkan penemuan ini mempunyai implikasi untuk amalan masa depan, kombinasi model dan hiperparameter harus dipilih dengan bijak untuk mendapatkan ramalan yang lebih tepat dalam peramalan siri masa.

Kata kunci: LSTM; pembelajaran mendalam; siri masa; ramalam; BiLSTM

1. Introduction

A Recurrent Neural Network (RNN) is a type of artificial neural network capable of processing the sequence of inputs using their internal memory due to their recurrently connected units (Ding *et al.* 2025). However, RNN is affected by the vanishing gradient problem, which happens when the gradient that contains information used in the parameter update decreases gradually. Eventually, the parameter update becomes insignificant. To overcome the vanishing gradient problem, several variations of RNN have been developed. According to Cruz-Victoria *et al.* (2024), one of the variations of RNN is unidirectional Long Short-Term Memory, also known as LSTM. It was first introduced by Hochreiter and Schmidhuber (1997) to tackle the problem of RNN with long-term dependency. The components that make up a typical LSTM unit are an input gate, a forget gate, an output gate, and a memory cell. Furthermore, to allow the LSTM network to reset its state, Gers *et al.* (2000) developed the forget gate, which was not initially a component of LSTM (Van Houdt *et al.* 2020). The three gates regulate the information flow inside the memory cell, which stores value over a period of time.

Bidirectional Long Short-Term Memory (BiLSTM) is another variation of RNN. The BiLSTM model comprises two layers. One layer processes the data sequence in the forward direction, while the other does the action backward (Althelaya *et al.* 2018). It was first introduced by Graves and Schmidhuber (2005), and they applied it to framewise phoneme classification. Notably, LSTM and BiLSTM have gained significant attention in recent years due to the rise of Artificial Intelligence (AI). They are also utilised to perform various tasks, such as language modelling (Rahman *et al.* 2021), speech recognition (Feng 2024), text classification (Duan & Raga 2024), and time series forecasting (Li *et al.* 2024).

In time series forecasting, such as financial time series, LSTM has proven to forecast with higher accuracy (Yang 2021). Liu *et al.* (2018) built two different LSTM models, one single-layer LSTM and one three-layer LSTM, to predict stock transactions. It was discovered that the three-layer LSTM has a better prediction accuracy than the single-layer LSTM. However, they concluded that computing resources and training time should be considered in choosing the proper number of stacked layers in the LSTM model to obtain prediction results with higher accuracy. Similarly, Fang *et al.* (2023) applied LSTM models to three datasets with a one-step-ahead forecasting approach. The datasets are the Shanghai Stock Exchange 180, the Standard & Poor's 500 Index and the China Securities Index 300. Their results demonstrated that LSTM enhances the robustness of forecasting outcomes. Yadav *et al.* (2020) compared stateless and stateful LSTM models for predicting Indian stock prices and found that the differences in predictive performance were statistically insignificant. After tuning the hidden layers in the LSTM model, they concluded that a single hidden layer provided the best accuracy based on RMSE values. While LSTM has shown considerable success in financial time series forecasting, further research is needed to identify the optimal model architecture. Additionally, future studies should explore state-of-the-art deep learning modules to enhance forecasting performance and adaptability.

BiLSTM represents an advancement in deep learning, particularly in time series forecasting. Several studies have compared the performance of BiLSTM and LSTM in predictive tasks. According to Siami-Namini *et al.* (2019), BiLSTM requires a longer training time and processes more data batches before reaching equilibrium compared to LSTM. Their study, which forecasted stock prices using autoregressive integrated moving average (ARIMA), LSTM, and BiLSTM, found that BiLSTM outperformed both ARIMA and LSTM models in predictive accuracy. Additionally, LSTM and BiLSTM have also been applied to daily price predictions of various foreign exchange rates (García *et al.* 2024). The results indicated that the BiLSTM network significantly reduced prediction errors, as measured by multiple error evaluation metrics. Comparative studies beyond just financial data also demonstrate the efficiency of BiLSTM, with observed error reductions between approximately 7% and 47%.

To train and test a neural network, it is essential to establish the network's optimal structure and parameters. The determination of the structure of the network, such as the size of hidden layers, and the structure of the learning parameters, such as the learning rate, is required in training and testing a neural network (Ilemobayo *et al.* 2024). This type of parameter is usually known as a hyperparameter since it must be set prior to the actual training of the neural network model. Note that hyperparameters can highly affect the performance of a model since they are used to construct and build different aspects of the learning algorithm (Claesen & Moor 2015). Hyperparameter tuning involves choosing the optimal hyperparameters for a learning algorithm. However, hyperparameter tuning is the main issue in machine learning since it is a decisive factor in determining whether a trained model is state-of-the-art or simply average (Jomaa *et al.* 2019). Therefore, identifying the optimal hyperparameters is essential for maximising the model's performance.

Methods to tune the hyperparameter include Bayesian optimisation, grid search, and random search (Quan 2024). There are many studies related to the tuning methods mentioned. Bergstra and Bengio (2012) performed grid search and random search methods in hyperparameter optimisation for neural networks. According to Yuanyuan *et al.* (2017), the simulation result of the study suggested that the grid search method is better than the Genetic Algorithm (GA) method in hyperparameter optimisation with less time consumed. Meanwhile, Wu *et al.* (2021) compared the performance of the methods of Bayesian optimisation and grid search in hyperparameter optimisation of machine learning models in their study. The paper highlighted that both Bayesian optimisation and grid search have similar performance, and the hyperparameter optimisation algorithms have significantly improved the accuracy of the models. These findings are further supported by Quan (2024), who demonstrated that grid, random, and Bayesian search yield similar tuning performance. Despite the availability of more adaptive tuning methods, grid search is still widely used because it follows a clear, systematic process that checks every possible combination of hyperparameters within a set range. The grid search provides a clear and interpretable comparison of different hidden neuron configurations, minimising the variability introduced by other hyperparameters and making it suitable for this study.

Several hyperparameters influence the performance of neural network architecture, including batch size, number of epochs, and activation function. The number of hidden neurons is another crucial hyperparameter in constructing the architecture. Therefore, selecting the optimum number of hidden neurons may significantly boost the performance of the neural network architecture. However, there are no clear guidelines for selecting the optimal number of hidden neurons for forecasting time series data. Tuning the number of hidden neurons should be performed to identify the best structure for LSTM and BiLSTM models. Specifically, choosing the number of hidden neurons is crucial for the model's predictive performance (Qu & Zhao 2019). They conducted a study to forecast the foreign exchange price by training LSTM models with 5, 10, 15, and 20 hidden neurons. The results suggested ten hidden neurons are the best hyperparameter choice since the model generated the lowest error. They stated that the model may be unable to extract the features of the historical data well if the number of hidden neurons chosen is too small. In contrast, if the number of hidden neurons is too large, the model may perform poorly in the testing set despite performing well in the training set.

Numerous studies examine LSTM and BiLSTM models for time series forecasting, as these models effectively capture long-term dependencies. However, existing research lacks consensus on the optimal number of hidden neurons for time series forecasting with LSTMs and BiLSTMs. Focusing on this crucial aspect of model tuning will enhance the reliability and efficiency of these two models (LSTM and BiLSTM) in forecasting practice.

2. Materials and Methods

2.1. *Long Short-Term Memory*

The LSTM-based models improve RNNs by providing a clean solution to the vanishing gradient problem. LSTMs have a longer memory and can learn from inputs separated from one another by longer time lags. The memory of LSTM is known as a gated cell due to the ability to decide whether to keep or ignore the memory information. Additionally, the weight values of the gate assigned to the information during the training process help decide whether the information should be deleted or preserved. A standard LSTM unit consists of a cell and three gates (forget gate, input gate, output gate). The three gates function together to regulate the flow of information across the cell. Figure 1 below illustrates the architecture of LSTM.

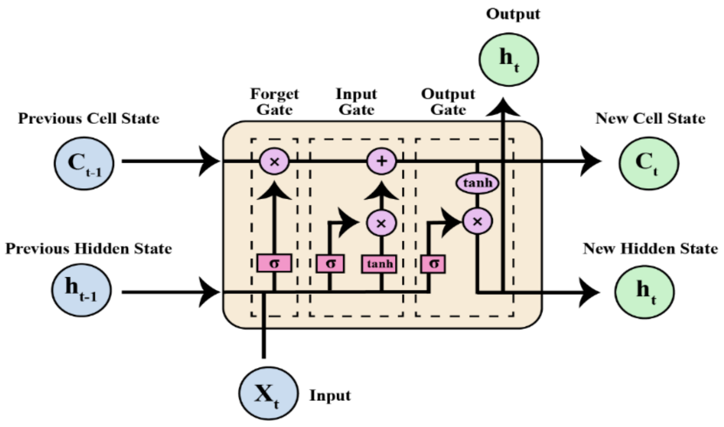


Figure 1: Architecture of LSTM

From the figure above, the previous hidden state and the current input are passed through the activation function, which is the sigmoid function in the forget gate. The process is necessary to decide which information should be retained or discarded. The previous hidden state and the current input are also passed to the sigmoid function and the hyperbolic tangent (Tanh) function of the input gate. The outputs from both functions at the input gate are multiplied together. The forget and input gate outputs are utilised to update the cell state. The previous cell state undergoes a pointwise multiplication with the forget vector, which is then pointwise added to the output from the input gate to derive the new cell state. Finally, the output gate defines the new hidden state. It processes the previous hidden state and current input into a sigmoid function, while the updated cell state passes through a tanh function. The results from the sigmoid and tanh are multiplied to generate the new hidden state. Both the updated cell state and the new hidden state are forwarded to the next time step. Generally, the three gates serve unique roles. The forget gate assesses whether to retain or discard existing information, the input gate manages how much new input is integrated into memory, and the output gate regulates if the current cell value should contribute to the output.

2.2. *Bidirectional Long Short-Term Memory*

The BiLSTM employs two hidden layers to process input data in both forward and backward directions and links both hidden layers to a single output layer, as illustrated in Figure 2.

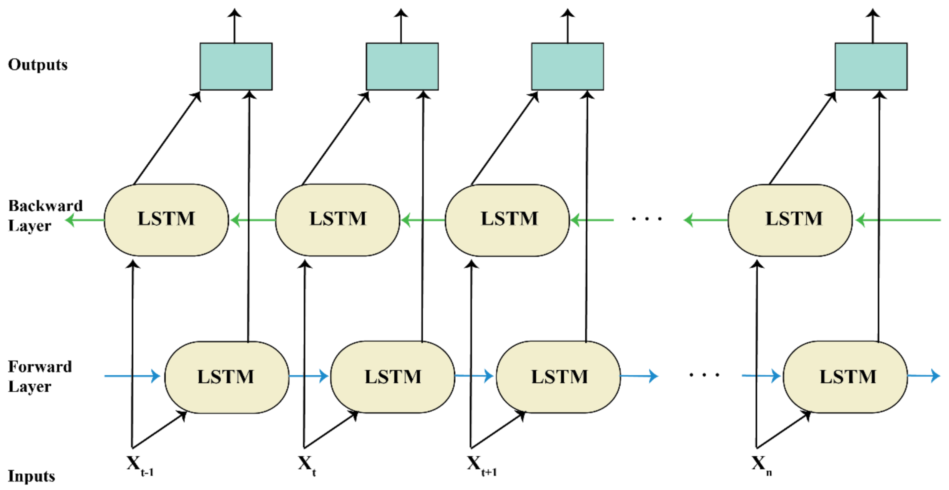


Figure 2: Architecture of BiLSTM

The backward layer processes the input data in a backward direction, while the forward layer processes the input data in the forward direction. The backward direction processing enables the model to capture the input data's pattern and hidden characteristics, which was generally neglected by LSTM (Yildirim 2018). Thus, applying the LSTM twice enhances the theoretical predictive performance of BiLSTM compared to the basic unidirectional LSTM (Yang & Wang 2022). The training algorithms, including activation functions, learning algorithm, and optimisation algorithm used in the BiLSTM in this paper, are similar to those used in the LSTM model.

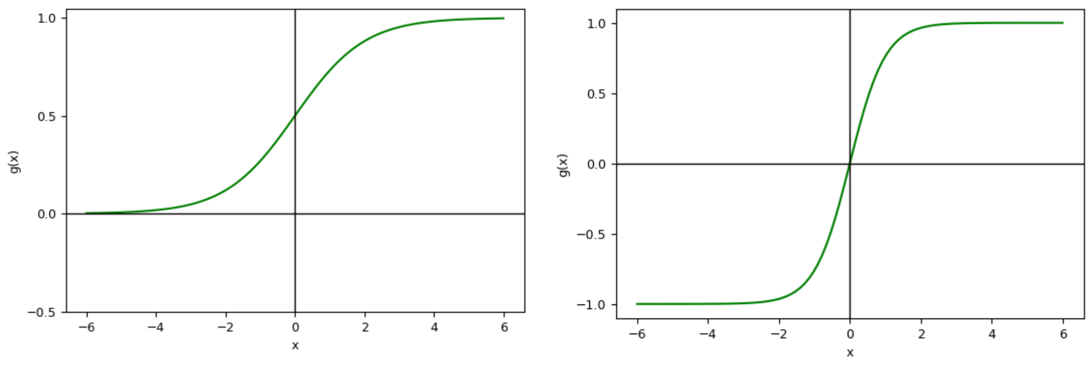
2.3. *Hyperparameter*

2.3.1. *Hidden Neuron*

The hidden layer is between a neural network's input and output layers. The number of neurons in this layer is an essential hyperparameter in configuring LSTM and BiLSTM since they carry the information and data from layer to layer. The number of hidden neurons should be tuned to optimise the performance of LSTM and BiLSTM. Similar to the study conducted by Sunny *et al.* (2020) and Qu and Zhao (2019), LSTM and BiLSTM models will be trained with hidden neurons varying from 32, 64, and 128 in this study. The performance of the models will be compared. It is to be determined which number of hidden neurons enable the best performance of LSTM and BiLSTM models. The other hyperparameters are kept constant. Since the developed model is used to address a regression problem, the output neuron is fixed at 1. The number of epochs used were 20, 50, and 100. The batch size selected in the Yadav *et al.* (2020) study is 64. However, Masters and Luschi (2018) suggested that a smaller batch size, such as 32, can enable the neural network to perform better in terms of training stability and generalisation. Hence, to compare the effect of different batch sizes on the model's performance, batch sizes of 32, 64, and 128 are utilised in this study. The Adam optimiser with its default learning rate of 0.001 and a dropout rate of 0.2 is employed to train the LSTM and BiLSTM models, and the selected loss function is Mean Square Error (MSE).

2.3.2. *Activation Function*

The activation functions are functions used in artificial neural networks to convert input signals into output signals, fed as input to the subsequent layer in the stack. There are linear activation functions and the more widely used non-linear activation functions. Non-linear activation functions facilitate the adaption of the model with a variety of data and enable it to differentiate between outputs (Feng & Lu 2019). Meanwhile, the non-linear activation functions used in LSTM and BiLSTM models are the sigmoid function and hyperbolic tangent function (Tanh). The sigmoid function is also known as the logistic function, which is the most widely used activation function since it is non-linear. It is a smooth S-shaped function. A hyperbolic tangent function can be defined as the ratio of the sine and cosine functions. Moreover, it is similar to the sigmoid function, but it is symmetric about zero. The sigmoid curve and hyperbolic tangent function curve are illustrated in Figure 3.



(a) (b)

Figure 3: Activation function (a) Sigmoid, (b) Tanh

The sigmoid function takes the input value and transforms it into output ranging from 0 to 1. The sigmoid function is a continuous function, and it is continuously differentiable. This function is not symmetric about zero, causing the signs of all output values of neurons to be the same. The equation of the sigmoid function is written as below:

$$g(x) = \frac{1}{1 + e^{-x}}. \tag{1}$$

The tanh function produces output ranging from -1 to 1, and can be computed as follows:

$$g(x) = \frac{sinh\ sinh\ (x)}{cosh\ cosh\ (x)} = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}. \tag{2}$$

Since the hyperbolic tangent function is symmetric about the origin, the outputs are more likely, on average, close to zero. The gradient of the hyperbolic tangent function is steeper than that of the sigmoid function. Therefore, the hyperbolic tangent functions are more preferred when compared to sigmoid functions.

2.4. *Forecast accuracy measures*

It is crucial to check the performance accuracy of the models after obtaining the predicted value from the experiment. MSE is often utilised in studies to measure accuracy. MSE measures the differences between the actual values and predicted values. MSE measures the errors' variability. Hence, the smaller the MSE, the better the prediction performance. One drawback of MSE is that it is sensitive to data transformations or changes in scale, and it penalises extreme errors. Therefore, another measure that will be used in this study is MAE. MAE does not heavily penalise extreme errors, as the measurement calculates the average absolute difference between predicted and actual values. A lower MAE indicates better forecasting accuracy. The formula to compute MSE and MAE is as follows:

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2, \tag{3}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t|, \tag{4}$$

where y_t is the actual value at timestep t and $\hat{y}_t$ is the predicted value at timestep t.

3. Results and Discussion

Real-life stock price data used in this study is gathered from Yahoo Finance. The raw dataset of Google stock price covered 1,260 observations for five years, from 25 May 2017 to 25 May 2022. The data of adjusted closing value is used to conduct the analysis in this study, with no missing values. The data is divided into an 80:20 ratio, with 80% allocated for training and the remaining 20% reserved for testing. Hence, the testing datasets will be 252 data. The training dataset is used to train and run through LSTM and BiLSTM models. The models in this study were built using Python 3.7, and the Keras library was used as the main machine-learning framework. Besides, Pandas library also used for reading the data files, handling missing values, and manipulating data, NumPy for numerical operations and reshaping arrays. Matplotlib is used for visualising the data, while Scikit-learn is employed for data normalisation using MinMaxScaler and for evaluating model performance through MSE and MAE. Specifically, the Keras library with a TensorFlow is used to build and train LSTM and BiLSTM, incorporating dropout for regularisation and the Adam optimizer for efficient training. The trained models were then used to forecast the stock price and evaluate the forecast accuracies.

Before analysing the dataset with LSTM and BiLSTM models, data transformation was performed to enhance the models' learning capabilities. Normalisation using min-max was applied to rescale the data within a range of 0 to 1, ensuring consistency in input values and improving training stability. To account for the stochastic nature of neural networks and ensure result reliability, each configuration, based on different hyperparameter combinations, was performed 20 times. This iterative approach helped mitigate variations caused by random initialisation and training dynamics, leading to a more robust evaluation. The MSE and MAE were then computed to assess prediction accuracy and error magnitude. The performance of the real data analysis forecast accuracy is provided in Table 1.

Table 1: MSE and MAE of LSTM and BiLSTM.

Model	**Hidden Neurons**	**Batch Size**	**MSE**				**MAE**			
			Epochs							
			20	**50**	**100**	**150**	**20**	**50**	**100**	**150**
LSTM	32	32	31.7869	19.3176	13.6255	**12.9086**	4.3450	3.3855	2.8357	**2.7416**
		64	43.3789	27.9336	17.0904	15.3419	4.9719	4.0263	3.1436	3.0236
		128	45.8412	40.1011	23.8870	17.2883	5.0756	4.7655	3.7253	3.1582
	64	32	28.0949	18.7536	11.6758	**10.4011**	4.0709	3.3605	2.6549	**2.4542**
		64	37.3382	23.7548	15.0127	11.5444	4.6567	3.7728	3.0017	2.6408
		128	44.3155	30.2102	18.9988	15.7130	4.9841	4.1634	3.3254	3.0477
	128	32	27.3478	15.5972	9.8734	**8.4748**	4.0216	3.0565	2.4258	**2.2192**
		64	30.8474	20.1610	13.2749	9.6646	4.2948	3.5028	2.8382	2.4203
		128	39.2927	28.2479	18.8277	11.7920	4.7052	4.1018	3.3193	2.6499
BiLSTM	32	32	17.8243	13.5414	12.1220	**10.3749**	3.2670	2.8605	2.7424	**2.5044**
		64	23.3622	14.0980	12.6782	10.7790	3.6799	2.9075	2.7433	2.5391
		128	29.7172	18.7634	13.6095	11.5996	4.1257	3.2947	2.8106	2.5992
	64	32	17.5221	11.4504	10.4568	**9.4258**	3.2383	2.6182	2.5364	**2.3900**
		64	20.5757	13.6797	11.1715	9.7077	3.4599	2.8648	2.5827	2.4037
		128	27.2694	16.9565	12.3308	10.9602	3.9238	3.1514	2.6958	2.5254
	128	32	15.8564	11.4043	9.8734	**7.4766**	3.1070	2.5978	2.4258	**2.1031**
		64	19.0703	13.0986	10.5710	9.0791	3.3812	2.8508	2.5274	2.3455
		128	25.1989	14.4455	11.3258	10.0351	3.7601	2.9440	2.5651	2.4384

Based on Table 1, the LSTM model that generated the lowest MSE and MAE from the analysis is the one with 128 hidden neurons, 150 epochs, and a batch size of 32, where the values for MSE and MAE are 8.4748 and 2.6499, respectively. Similarly, the BiLSTM model that generated the lowest MSE and MAE from the analysis is the one with 128 hidden neurons, 150 epochs, and a batch size of 32, where the values for MSE and MAE are 7.4766 and 2.1031, respectively. It can be observed from the table that both LSTM and BiLSTM models with 128 hidden neurons produced the lowest MSE and MAE when compared to other numbers of hidden neurons. The MSE and MAE decrease with the increased hidden neurons from 32 to 64 and 128. This indicates a negative relationship between the number of hidden neurons and the error values. However, batch size has a positive relationship with the MSE and MAE values. The error values increase with the increase of batch size from 32,64 to 128. Furthermore, it can also be observed that the MSE and MAE for both models decrease when the epochs increase from 20 to 150. By taking 32 hidden neurons and a batch size of 32 as an example, Figure 4 below displays the effect of increasing epochs on the forecast accuracy of the LSTM model.

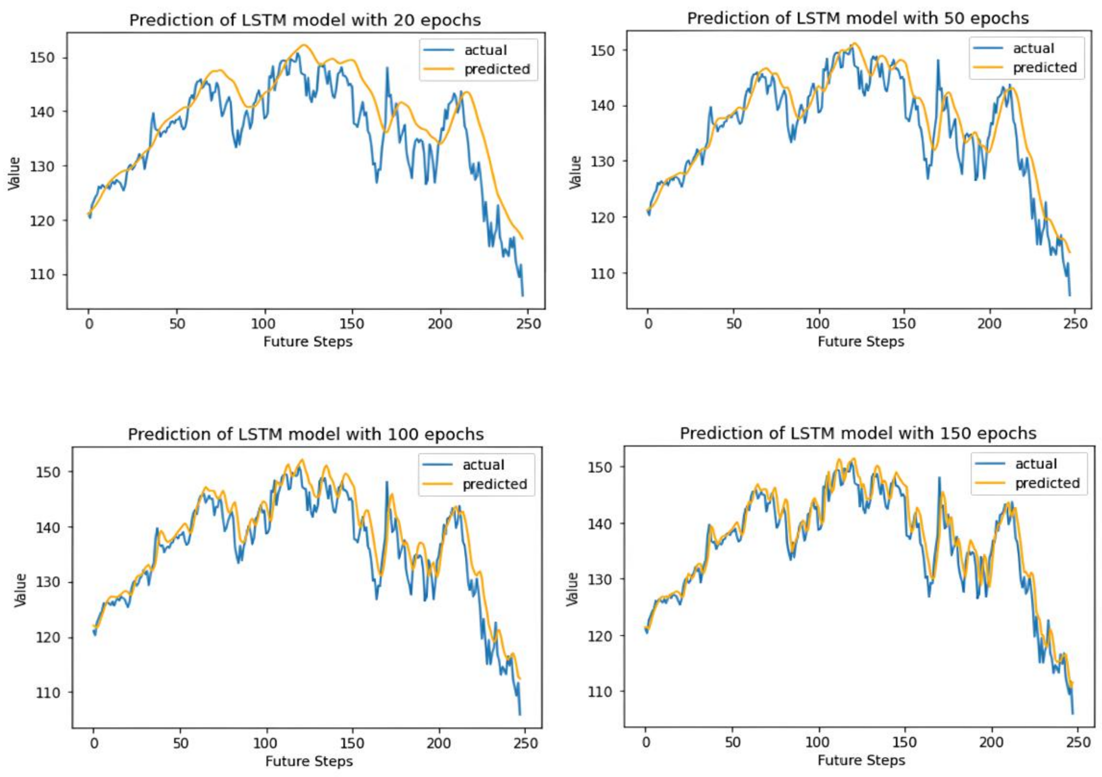


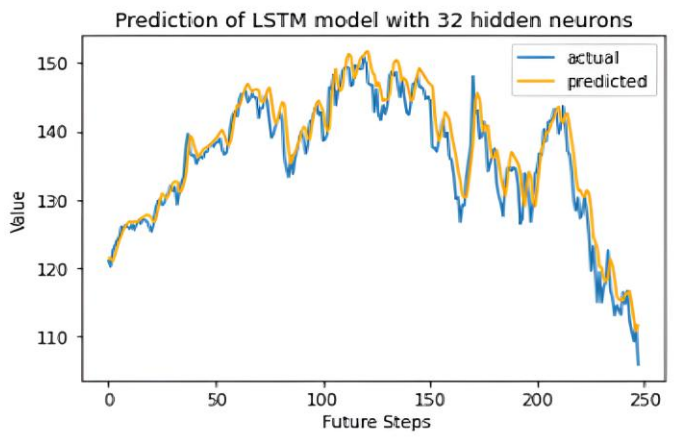
Figure 4: Stock price prediction of LSTM with different epochs

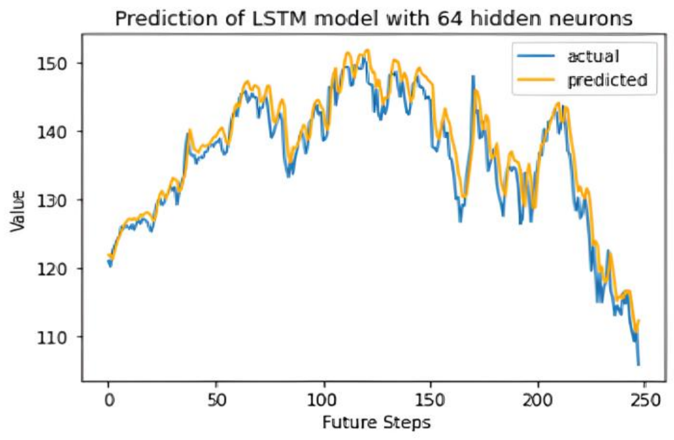
A bias-variance trade-off analysis for both LSTM and BiLSTM models with varying numbers of hidden neurons (32, 64, and 128) was conducted to enhance the reliability of the model selection further. The analysis is shown in Table 2. The results concentrate on a batch size of 32 and an epoch of 150, as noted in Table 1, indicating the optimal batch size and epoch for this study.

Table 2: MSE and MAE of LSTM and BiLSTM.

Model	Hidden Neurons	MSE		MAE	
		Train	Test	Train	Test
LSTM	32	2.8102	12.9086	1.1866	2.7416
	64	2.8189	10.4011	1.2591	2.4542
	128	**1.8934**	**8.4748**	**0.9859**	**2.2192**
BiLSTM	32	1.9623	10.3749	0.9818	2.5044
	64	**1.7208**	9.4258	**0.9150**	2.3900
	128	1.8818	**7.4766**	0.9469	**2.1031**

From Table 2, the LSTM model with 128 neurons achieves the lowest training error (1.893 MSE) and the lowest test error (8.4748 MSE), suggesting a good balance between bias and variance. Similarly shown in MAE results. Besides, the result of BiLSTM is also shown in Table 2. The BiLSTM initially achieves the lowest training error at 64 neurons (MSE = 1.7208, MAE = 0.9150), indicating strong learning capability. However, generalisation to unseen data shows the lowest test error with 128 neurons (MSE = 7.4766, MAE = 2.1031). This suggests that BiLSTM benefits from additional complexity, leveraging its bidirectional structure to capture more temporal dependencies, leading to better overall performance.





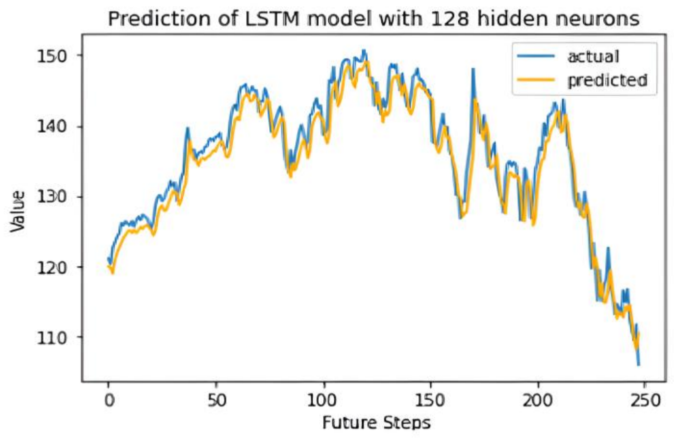
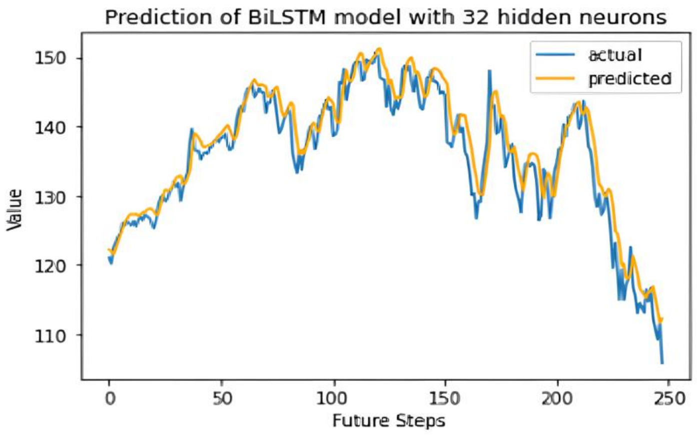
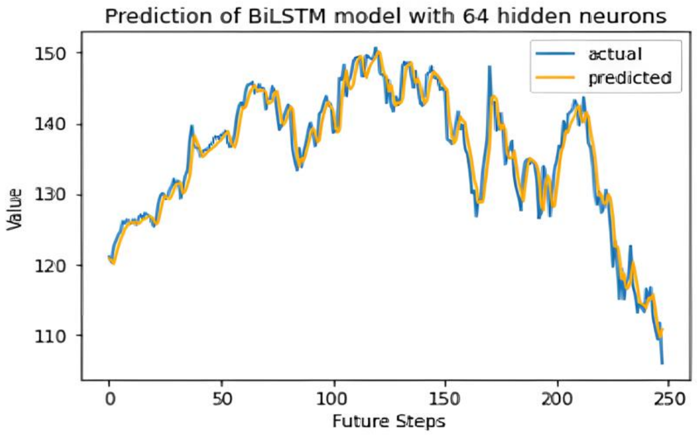


Figure 5: The actual and stock price prediction of LSTM with different hidden neurons





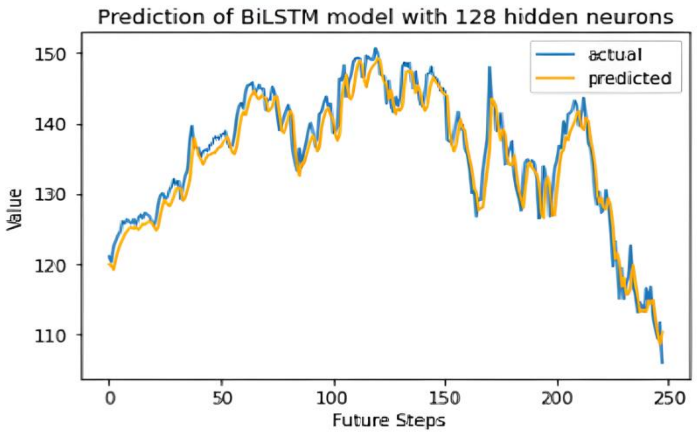


Figure 6: The actual and stock price prediction of BiLSTM with different hidden neurons

In addition, Figures 5 and 6 above illustrate the actual versus forecasted values of the best-performing LSTM and BiLSTM models with different numbers of hidden neurons. The graphs suggest that the gap between the actual and predicted values decreases when the number of hidden neurons increases from 32, 64, to 128 for both models. Hence, it can be concluded that among the three models with different numbers of hidden neurons, the LSTM and BiLSTM models with 128 hidden neurons perform better with the lowest MSE and MAE. The performance of LSTM and BiLSTM models with different numbers of hidden neurons in forecasting real-time series data is being compared, and the results are summarised in Table 3 below. In comparing the reduction percentage, the MSE values range between 9.35% and 19.61% and range from 2.62% to 8.65% for MAE values of reduction for BiLSTM over LSTM, indicating that the error values observed for BiLSTM are lower than that of the LSTM model.

Table 3: Comparison MSE and MAE between LSTM and BiLSTM.

Measurement	**Hidden Neurons**	**LSTM**	**BiLSTM**	**Percentage of Reduction (BiLSTM over LSTM)**
MSE	32	12.9060	10.3749	19.61
	64	10.4011	9.4258	9.35
	128	8.4748	7.4766	11.78
MAE	32	2.7416	2.5044	8.65
	64	2.4542	2.3900	2.62
	128	2.2192	2.1031	5.23

Wilcoxon-signed rank test was then conducted to further confirm that the BiLSTM model performed better than the LSTM model in forecasting the time series data. Wilcoxon signed-rank test is a frequently used non-parametric test for paired samples (Flores 1989; Wang *et al.* 2020). One of the advantages of using this test is that it does not assume the data is normally distributed and does not require any assumptions about the shape of the distribution. Both the MSE and MAE values obtained for each model were used to perform the Wilcoxon signed-rank test to ensure consistency of the result. Table 4 below provides test results for both models for MSE and MAE values.

Table 4: Test results of Wilcoxon Signed-Rank.

Type of Values Used	***p*-value**
MSE	$2.588e{-}07$
MAE	$4.695e{-}06$

Based on Table 4, the tests' p-values using MSE and MAE are less than the 0.05 significance level. It is clearly revealed that there is a significant difference between the LSTM and BiLSTM in predicting the data used in this study. In addition, MSE and MAE values observed for BiLSTM models are relatively lower than those for LSTM models.

BiLSTM consistently outperforms LSTM due to the input of processes in both directions. Thus, the model can capture more temporal patterns compared to LSTM, which only learns from past data (Graves & Schmidhuber 2005). This ability makes BiLSTM particularly effective in financial time series forecasting, where multiple interdependent factors influence price movements. Additionally, the findings align with previous studies highlighting BiLSTM’s advantages in various forecasting tasks beyond financial markets, such as speech recognition and medical diagnostics (Siami-Namini *et al.* 2019).

4. Conclusion

This research conducted experimental studies on applying machine learning to time series data. Grid search was used to optimise the hyperparameters of LSTM and BiLSTM models to enhance the performance of the neural network. The primary hyperparameter adjusted in the experiment is the number of hidden neurons, specifically 32, 64, and 128 neurons were utilised. The results from the experiment indicated that the MSE and MAE values decrease with the increase in the number of hidden neurons. Accordingly, 128 hidden neurons are the optimum number of hidden neurons with the highest prediction accuracy. Hidden neurons are artificial neurons that receive a set of weighted inputs and generate an output using an activation function. While a small number of hidden neurons may improve generalisation, it may also cause the neural network to lose the ability to solve the problem (Thomas *et al.* 2015). Therefore, determining the optimal number of hidden neurons through analysis with various datasets is essential as it depends on the network architecture, the noise level, and the function's complexity.

An analysis of real data is conducted to compare the consistency of the findings. A set of real stock price data was used to analyse real data. The results revealed that 128 hidden neurons are the optimum number of hidden neurons, and the LSTM and BiLSTM models perform better. Furthermore, the Wilcoxon signed-rank test was conducted, and the results indicated a significant difference between the performance of LSTM and BiLSTM, with BiLSTM producing lower error values. This result is compatible with the study conducted by Siami-Namini *et al.* (2018), which highlighted that BiLSTM outperformed the regular unidirectional LSTM in the context of forecasting financial time series data. Hence, it can be concluded that the BiLSTM performs better hence more accurate than the LSTM model in forecasting.

This study has limitations and centres on actual stock price data. Future research could explore various datasets to enhance the generalisation of hyperparameter optimisation for both models. In addition, the study could be extended using different hybrid models of LSTM in forecasting time series data rather than the basic LSTM and BiLSTM models. Hybrid models of LSTM are widely discussed in the deep learning literature and have been utilised in different areas of study. In addition to neural networks, this study can also be further expanded through the perspective of time series, which can be expanded to forecast future multivariate and seasonal time series data.

Acknowledgments

The authors would like to thank the Ministry of Higher Education for providing financial support under the Fundamental Research Grants Scheme - Early Career Research (FRGS-EC) No. FRGS-EC/1/2024/STG06/UMPSA/02/1 (University reference RDU243711) and Universiti Malaysia Pahang Al-Sultan Abdullah for additional financial support under the Fundamental Research Grant with registration number RDU230396 and Tabung Persidangan Dalam Negara.

References

Althelaya K.A., El-Alfy E.S.M. & Mohammed S. 2018. Evaluation of bidirectional LSTM for short-and long-term stock market prediction. *2018 9th International Conference on Information and Communication Systems (ICICS)*, pp. 151–156.

Bergstra J. & Bengio Y. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research* **13**: 281–305.

Claesen M. & De Moor B. 2015. Hyperparameter search in machine learning. arXiv preprint arXiv:1502.02127.

Cruz-Victoria J.C., Netzahuatl-Muñoz A.R. & Cristiani-Urbina E. 2024. Long short-term memory and bidirectional long short-term memory modeling and prediction of hexavalent and total chromium removal capacity kinetics of Cupressus lusitanica bark. *Sustainability*, **16**(7): 2874.

Ding H., Hou H., Wang L., Cui X., Yu W. & Wilson D.I. 2025. Application of convolutional neural networks and recurrent neural networks in food safety. *Foods* **14**(2): 247.

Duan A. & Raga R.C. 2024. BiLSTM model with attention mechanism for multi-label news text classification. *2024 4th International Conference on Neural Networks, Information and Communication Engineering (NNICE)*, pp. 566–569.

Fang Z., Ma X., Pan H., Yang G. & Arce G.R. 2023. Movement forecasting of financial time series based on adaptive LSTM-BN network. *Expert Systems with Applications* **213**: 119207.

Feng J. & Lu S. 2019. Performance analysis of various activation functions in artificial neural networks. *Journal of Physics: Conference Series* **1237**(2): 022030.

Feng Y. 2024. Intelligent speech recognition algorithm in multimedia visual interaction via BiLSTM and attention mechanism. *Neural Computing and Applications* **36**(5): 2371–2383.

Flores B.E. 1989. The utilization of the Wilcoxon test to compare forecasting methods: A note. *International Journal of Forecasting* **5**(4): 529–535.

García F., Guijarro F., Oliver J. & Tamošiūnienė R. 2024. Foreign exchange forecasting models: LSTM and BiLSTM comparison. *Engineering Proceedings* **68**(1): 19.

Gers F.A., Schmidhuber J. & Cummins F. 2000. Learning to forget: Continual prediction with LSTM. *Neural Computation* **12**(10): 2451–2471.

Graves A. & Schmidhuber J. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* **18**(5-6): 602–610.

Hochreiter S. & Schmidhuber J. 1997. Long short-term memory. *Neural Computing* **9**(8): 1735–1780.

Ilemobayo J.A., Durodola O.I., Alade O., Awotunde O.J., Adewumi T.O., Falana O., Ogungbire A., Osinuga A., Ogunbiyi D., Ifeanyi A., Odezuligbo I.E. & Edu O.E. 2024. Hyperparameter tuning in machine learning: A comprehensive review. *Journal of Engineering Research and Reports* **26**(6): 388–395.

Jomaa H.S., Grabocka J. & Schmidt-Thieme L. 2019. Hyp-RL: Hyperparameter optimization by reinforcement learning. arXiv preprint arXiv: 1906.11527v1.

Li F., Liu S., Wang T. & Liu R. 2024. Optimal planning for integrated electricity and heat systems using CNN-BiLSTM-Attention network forecasts. *Energy* **309**: 133042.

Liu S., Liao G. & Ding Y. 2018. Stock transaction prediction modeling and analysis based on LSTM. *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA),* pp. 2787–2790.

Masters D. & Luschi C. 2018. Revisiting small batch training for deep neural networks. arXiv preprint arXiv: 1804.07612v1.

Qu Y. & Zhao X. 2019. Application of LSTM neural network in forecasting foreign exchange price. *Journal of Physics: Conference Series* **1237**(4): 042036.

Quan S.J. 2024. Comparing hyperparameter tuning methods in machine learning based urban building energy modeling: A study in Chicago. *Energy and Buildings* **317**: 114353.

Rahman M.M., Watanobe Y. & Nakamura K. 2021. A Bidirectional LSTM language model for code evaluation and repair. *Symmetry* **13**(2): 247.

Siami-Namini S., Tavakoli N. & Namin A.S. 2019. The performance of LSTM and BiLSTM in forecasting time series. *2019 IEEE International Conference on Big Data (Big Data)*, pp. 3285–3292.

Siami-Namini S., Tavakoli N. & Siami Namin A. 2018. A comparison of ARIMA and LSTM in forecasting time series. *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1394–1401.

Sunny M.A.I., Maswood M.M.S. & Alharbi A.G. 2020. Deep learning-based stock price prediction using LSTM and Bi-Directional LSTM model. *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES),* pp. 87–92.

Thomas A.J., Petridis M., Walters S.D., Gheytassi M.S. & Morgan R.E. 2015. On predicting the optimal number of hidden nodes. *2015 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 565-570.

Van Houdt G., Mosquera C. & Nápoles G. 2020. A review on the long short-term memory model. *Artificial Intelligence Review* **53**(8): 5929–5955.

Wang B., Jiang T., Zhou X., Ma B., Zhao F. & Wang Y. 2020. Time-series classification based on fusion features of sequence and visualization. *Applied Sciences* **10**(12): 4124.

Wu X., Liu Z., Yin L., Zheng W., Song L., Tian J., Yang B. & Liu S. 2021. A haze prediction model in Chengdu based on LSTM. *Atmosphere* **12**(11): 1479.

Yadav A., Jha C.K. & Sharan A. 2020. Optimizing LSTM for time series prediction in Indian stock market. *Procedia Computer Science* **167**: 2091–2100.

Yang M. & Wang J. 2022. Adaptability of financial time series prediction based on BiLSTM. *Procedia Computer Science* **199**: 18–25.

Yang S. 2021. A novel study on deep learning framework to predict and analyze the financial time series information. *Future Generation Computer Systems* **125**: 812–819.

Yildirim Ö. 2018. A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification. *Computers in Biology and Medicine* **96**: 189–202.

Yuanyuan S., Yongming W., Lili G., Zhongsong M. & Shan J. 2017. The comparison of optimizing SVM by GA and grid search. *2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI),* pp. 354–360.

Centre for Mathematical Sciences
Universiti Malaysia Pahang Al-Sultan Abdullah
Lebuh Persiaran Tun Khalil Yaakob
26300 Gambang
Pahang, MALAYSIA
Email: haizum@umpsa.edu.my[*]

Department of Mathematics and Statistics
Faculty of Science
Universiti Putra Malaysia
43400 UPM Serdang
Selangor, MALAYSIA
E-mail: 202975@student.upm.edu.my, hsyahida@upm.edu.my

Received: 23 August 2024
Accepted: 7 April 2025

[*]Corresponding author