International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Optimization of Stock Movements Based on Historical Data for Stock Index Prediction Using Deep Learning Models

Bernadectus Yudi Dwiandiyanta[1], Rudy Hartanto*[2], Ridi Ferdiana [3]

Submitted:11/03/2024 **Revised**: 26/04/2024 **Accepted**: 03/05/2024

Abstract: Investors, analysts, and intellectuals are consistently known for predicting stock movements. Therefore, this research focuses on the importance of simplicity, relying solely on stock data that includes open, high, low, close, and volume prices. The objective is to forecast stock movements on the Indonesia Stock Exchange (IDX). To achieve the desired result, three well-known Deep Learning architectures were used namely, Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Recurrent Neural Networks (RNN). Furthermore, the architectures assisted in creating a series of prediction models based on the datasets. The dataset was sourced from the index of the Indonesian Stock Exchange, known as the Jakarta Composite Index. Through experimentation, efficiency, reliability, and susceptibility to fluctuations in data for each architecture were assessed. Consequently, the results showed that historical data alone could be used to create a stock prediction model, particularly when approached correctly. Among the three architectures explored, there was an observation that RNN achieved the highest level of prediction accuracy as the research signified the importance of simplicity in modeling. Based on the findings, further research could develop streamlined and effective stock prediction models that rely on minimal data.

Keywords: *Deep Learning Models, Stock Index Prediction, Jakarta Composite Index, Historical Data*

1. Introduction

Stock represents ownership in a company and is considered a volatile financial instrument. Despite including high risk, significant returns are offered when managed carefully in a portfolio [1]. By acquiring stock, investors become partial owners of the respective company, even when the ownership stake is minuscule. The stock market serves as the platform where the stock is traded, exchanged, and distributed [2].

Investing in stock provides several advantages which firstly include the potential for capital growth. This implies that the stock price of a growing company can increase in no time. Secondly, certain stocks yield dividends, signifying the earnings distribution of the company to its shareholders. Additionally, investing in stock facilitates portfolio diversification [3], which helps to mitigate risk. However, aside from these benefits, there are inherent risks associated with stock investments. [4]. For example, stock prices can show instability in short durations, influenced by various external factors, including macroeconomic conditions, political dynamics, and global events [5][6][7]. Consequently, the ability to forecast stock price trajectories is crucial for stock market investments [4].

Investors, irrespective of experience levels in the stock market, consistently pursue ways to improve their chances of profit. In the recent digital age, signified by numerous data and technological advancements, strategies centered on data and using artificial intelligence and machine learning are achieving popularity. However, before embarking on investments in the stock market, evaluating available options and identifying approaches that promise better returns is essential [8].

In recent decades, predicting stock movements has become an engaging activity, driven by the potential for monetary profits as well as its complex intellectual challenges. Forecasting stocks is a demanding task, prompting computer experts to use artificial intelligence to project future stock trajectories [9][10]. In addition, modern technology, particularly artificial intelligence and machine learning, has introduced innovative tools capable of addressing these challenges in exceptional ways.

Indonesia Stock Exchange (IDX) is a dynamic stock market experiencing evolution with distinctive characteristics. Movements reflect economic growth and geopolitical conditions in the region. Therefore, a well-crafted stock prediction model for IDX can offer strategic advantages to investors and major stakeholders. In this context, maximizing profits and improving the market value of company remain fundamental business objectives [11].

A widely recognized statistical method used in time series forecasting, including predictions for stocks, is ARIMA

[1] *Department of Electrical Engineering and Information Technology, Universitas Gadjah Mada, Indonesia*
Department of Informatics, Universitas Atma Jaya Yogyakarta, Indonesia
ORCID ID : 0009-0003-0963-849X
[2] *Department of Electrical Engineering and Information Technology, Universitas Gadjah Mada, Indonesia*
ORCID ID : 0000-0003-1126-2340
[3] *Department of Electrical Engineering and Information Technology, Universitas Gadjah Mada, Indonesia*
ORCID ID : 0000-0001-9961-5205
** Corresponding Author Email: rudy@ugm.ac.id*

(Auto-Regressive Integrated Moving Average). ARIMA combines the strengths of two models, namely Auto-Regressive (AR) and Moving Average (MA), to show the statistical dependency structures inherent in data. While ARIMA has proven effective in many instances and is widely used in the financial sector, it does have certain limitations. Specifically, challenges arise when exchanging stock data characterized by substantial noise, non-stationarity, market downturns, and susceptibility to various external influences.

Through the advent of the big data era and technological progression, methods based on machine learning are achieving projections in predicting stock outcomes [12]. Machine learning is a subset of artificial intelligence that enables computers to learn from data without precise programming. Furthermore, algorithms such as Random Forest, Gradient Boosting, and Neural Networks have been used for predicting stocks, yielding encouraging outcomes. Deep learning, which is a specialized branch of machine learning, shows its capability in modeling difficult time series data, primarily through architectures such as RNN [13][14], GRU [15][16], and LSTM [17][18].

In the historical data analysis for stock prediction, the different methods used relied on extensive datasets and various features such as technical indicators, fundamentals, news, and market sentiment. However, the utilization of these strategies often introduces complications in both interpretation and computation. In the current research, an alternative approach was adopted by focusing on the fundamentals and narrowing the scope to historical stock data, specifically open, high, low, close, and volume prices. Despite this deviation, concerns arise about potential omissions, suggesting that essential information within historical data may not have been thoroughly exploited.

This current exploration adopts an unconventional stance, describing the significance of historical stock data. To assess its potential as a foundation for constructing a strong stock prediction model for IDX, Deep Learning architectures, including LSTM, GRU, and RNN, are leveraged. The primary contribution of the research lies in providing an experiential understanding of the most suitable architecture for stock predictions found in historical data, which is adjusted for IDX by comparing the three architectures. Through this method, the research aims to bridge existing knowledge gaps in stock prediction and to also introduce fresh perspectives on the untapped potential of historical data. This approach addresses the limitations of traditional strategies and also opens avenues for innovation in advancing stock prediction models.

2. Related Work

In the current digital landscape, predictions of financial indices through machine learning and deep learning techniques have gathered significant interest from the intellectual community. Numerous explorations have wanted to use the capabilities of neural networks for forecasting trajectories in the stock market.

Lin et al., present a groundbreaking approach to forecasting stock prices by introducing the use of RNN. The aim is to predict the opening and closing prices, as well as the difference between prices. Moreover, a significant feature of this research is the importance of data pre-processing and the adoption of normalized first-order difference method. The main focus is on fundamental attributes of stock data, particularly the Zero Crossing Rate (ZCR). Additionally, the application of this method to indices such as the S&P 500 and Dow Jones shows a significant improvement in predictive precision compared to previous techniques [14].

Jarrar and Salim explore the effectiveness of screening using the Discrete Wavelet Transform (DWT) and RNN for forecasting stock price trajectories in the Saudi Arabian context. Recognizing the stock market as a multifaceted and dynamic environment, the research aims to improve predictive accuracy by using DWT to eliminate noise from the dataset. Leveraging RNN training facilitate by the Back Propagation Through Time (BPTT) technique. In addition to this, the examiners achieve superior predictive outcomes, particularly when compared with conventional forecasting algorithms such as ARIMA [13].

To improve the precision of predictions for stock indices, Gupta et al., propose a method that combines deep learning modalities with data augmentation to address the challenges of overfitting. The team introduces a GRU-anchored StockNet framework, divided into two principal modules. The first module, named the Injection module, is carefully created to counteract overfitting. Meanwhile, the second module, named the Investigation module, is modified for forecasting stock indices. The paradigm experiences validation in the Indian stock market domain, specifically CNX-Nifty. This scenario shows a marked decrease in test loss compared to alternative models lacking an anti-overfitting apparatus. The results of this intellectual inquiry show the importance of combining advanced techniques in deep learning with judicious data augmentation tactics to refine the quality of predictions for stock indices [15].

Through the evolution of technological paradigms, the Long Short-Term Memory (LSTM) technique has become a main point of academic analysis. Baek and Kim (2018) worked more than the introduction of the standard LSTM method, adding complexity through a data augmentation strategy. Subsequently, the conceived ModAugNet framework aims to improve the accuracy of stock price forecasts by enriching the training dataset [19]. Nguyen and Yoon explored the potential of Deep Neural Networks, with a focus on LSTM, in forecasting short-term stock price fluctuations. The findings of the team signify that LSTM possesses superior capability in discerning temporal patterns, thereby leading to outstanding improvements in predictive accuracy [10]. Similarly, Ji et al. advance the communication by introducing a deep learning-centric method modified specifically for stock price forecasting. The technique distinguishes the situation by an outstanding performance compared to existing benchmarking paradigms. It is crucial to be aware that the investigation further confirms the preeminence of deep learning modalities [12].

Manurung et al. focus on the salience of time series analysis in decoding stock market dynamics. Through the algorithms and paradigms the dynamics formulated, the exploration explains the efficacious deployment of LSTM across diverse market scenarios. Furthermore, LSTM shows both adaptability and flexibility to fluctuations in data curation [1]. Borovkova and Tsiamas made a significant contribution through the joint approach of LSTM neural networks. By joining insights from diverse LSTM paradigms, the method confers distinct benefits in predicting volatility. Moreover, this approach provides a more deep understanding of market perturbations [20]. Zhang et al. combine two distinct technological frameworks, specifically RNN and DBN, concluding in a hybrid DBN-RNN model that shows superior efficacy in its performance [3]. Luo et al. present an artificial intelligence forecasting model for stock market returns, using the capabilities of LSTM and strengthening it with a Shuffled Frog Leaping Algorithm (SFLA) that emulates amphibian behavior. This model is further supported with a mutation and crossover rectification method and has been subjected to empirical testing on stock market datasets [21]. Kim et al. (2019) explain the way integrating multiple economic paradigms can improve the effectiveness of LSTM [22]. Subsequent research, including [23] and [7], offers essential contributions, by explaining the effective use of LSTM in stock price forecasting.

3. Materials and Methods

3.1. Data

The research used stock data from the Indonesia Stock Exchange (IDX) which included information about the opening, highest, lowest, and closing prices of stocks, as well as the trading volume throughout a specific period. The source of the stock trading data was Yahoo Finance, providing data on a daily, weekly, and monthly basis. For this finding, the focus was on the Jakarta Composite Index (JCI), representing the stock trading index of IDX. The time range for the trading data was from January 1st, 2007, to December 31st, 2020. Moreover, this dataset included five metrics namely opening price, highest price, lowest price, closing price, and trading volume. Figure 1 showed the closing values of JCI from January 1st, 2000, to December 31st, 2022.

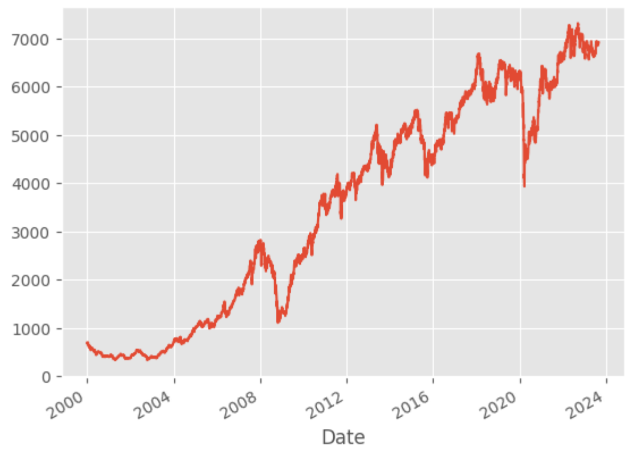


Fig. 1. JCI close price on January 1, 2000 – December 31, 2022

3.2. Data Preprocessing

Before starting the model training, there was a need to preprocess the stock data, including both cleaning and normalizing the data.

3.2.1. Data Cleaning

Data cleaning performed a crucial role in ensuring the quality of the prediction model. Dataset was carefully examined for any incomplete data, which were subsequently removed. Following this, gaps in data were common during stock exchange downtimes on holidays.

3.2.2. Normalization

Normalization is a technique used to scale the data in a range, which assisted in convergence and stability during training. Typically, the aim was to scale the values between 0 and 1 for optimal performance in learning models. In a stock dataset, attributes such as open, high, low, close, and volume had different value ranges. These scale differences could unintentionally prioritize features, potentially affecting model training. To ensure the importance of all features during training, normalization was applied as a result that these attributes influenced the final predictions. In the research, the Min-Max Normalization technique was used, adjusting data to be in the range of [0, 1] through a specific formula.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

3.3. Deep Learning Architecture

This research used three principal deep learning architectures namely, LSTM, GRU, and RNN.

3.3.1. LSTM

LSTM architecture was fitted to the category of networks (RNN), a type of artificial neural network. Introduced in 1997 by Hochreiter and Schmidhuber, LSTM addressed the vanishing gradient problem encountered in RNN during training with data sequences [24]. In this context, LSTM proved particularly useful for tasks including data, such as datasets related to stocks, due to its ability to retain information throughout sequences.

The basic structure of an LSTM comprised memory cells along with three gates namely, the input gate, the forget gate, and the output gate [25]. The input gate determined the information to be stored in the memory cell, while the forget gate decided what information should be discarded, and the output gate controlled which value should be passed on to the cell. In addition, these three gates worked together harmoniously, allowing LSTM to intelligently store, modify, or erase information based on the context of each data sequence [26]. Figure 2 showed the structure of LSTM model.

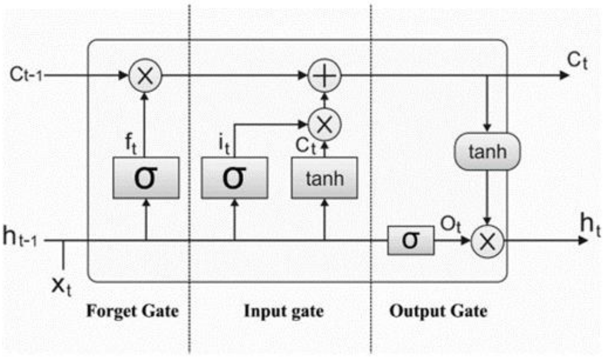


Fig. 2. LSTM structure [26]

LSTM architecture had an advantage in recognizing patterns in data sequences in certain periods [27]. The distinguished factor for LSTM was its ability to retain information from previous steps in the sequence, even when that information might not have been crucial for models. However, this feature positioned LSTM as one of the popular architectures in the field of deep learning, specifically when addressing problems that require a deep understanding of contextual data sequences.

A significant strength of LSTM remains in its proficiency in identifying patterns in data sequences across time intervals [27]. This implied that LSTM could preserve information from steps in the sequence, which might have been disregarded or ignored by models. In addition, this inherent capability placed LSTM at the forefront of deep learning architectures, particularly when tackling challenges that demanded an understanding of data sequences.

3.3.2. Gated Recurrent Units (GRU)

Introduced by Cho et al. in 2014, GRU became a variant of conventional Recurrent Neural Networks (RNN). The beginning of GRU was predicated upon addressing certain intrinsic limitations of traditional RNN, specifically the famous vanishing gradient dilemma, which impeded the capacity of RNN to assimilate long-term dependencies in data sequences. While LSTM captured three distinct gates, GRU, in its bid for stinginess, combined merely two gates namely the update and the reset gate [28].

The functionality of the update gate in GRU architecture was twofold including ascertaining the proportion of old information that should be retained and showing the extent to which new information should be combined. Conversely, the reset primary role of gate was to determine the information quantum from the antecedent time step that ought to influence the current data. In a comparative analysis with LSTM, GRU boasted a more streamlined architecture. This inherent simplicity afforded GRU accelerated training speeds and a reduction in the requisite parameters [28], as Figure 3 showed a visual representation of GRU architecture.

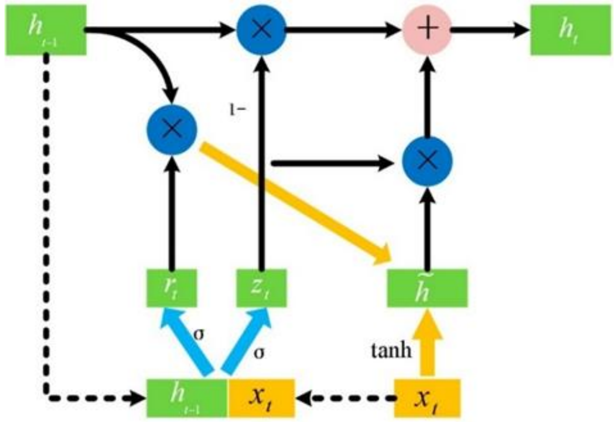


Fig. 3. GRU architecture [29]

From a comparative perspective, though the architecture of GRU was more fundamental, experimental evidence proposed that it delivered performance on par with LSTM across diverse tasks, particularly when exchanging with relatively shorter data sequences. Both frameworks excelled in retaining long-term information and adapting to the sequential nature of data. However, GRU became a computationally efficient alternative, distinguished by its reduced complexity and fewer parameters.

3.3.3. RNN

RNN possess the essential ability to preserve memory or information from earlier steps when analyzing data in subsequent time frames. RNN are composed of multiple neurons forming a network [30] which are designed specifically for managing sequential data. Additionally, this architecture provided RNN with a trace of memory about prior events in the sequence, distinguishing RNN from other neural network architectures such as feedforward networks, which treat each input in isolation.

Traditional RNN faced challenges in learning long-term dependencies in data sequences. When tasked with remembering information far apart in a sequence, RNN often encounter difficulties due to a problem known as the vanishing or exploding gradient [31]. However, this showed that information from early time steps frequently got lost when reaching more advanced time steps.

Conventional RNN encountered challenges in understanding long-term dependencies in sequential data. When these networks had to retain information across distant points in a sequence, RNN often struggled with the vanishing or exploding gradient problem [31]. Moreover, this problem led to the confusion or dilution of information from the early time steps as it progressed to later steps.

RNN served as a foundational architecture for the evolution of subsequent models, specifically LSTM and GRU. Both LSTM and GRU were conceived to address the inherent constraints of traditional RNN, specifically improving the capacity of RNN for long-term dependency retention. In numerous sequence-processing applications, RNN became a crucial component. Therefore, following the operational principles of RNN was crucial for comprehending the advancements in artificial intelligence modified for sequential data as the architecture of RNN was shown in Figure 4.

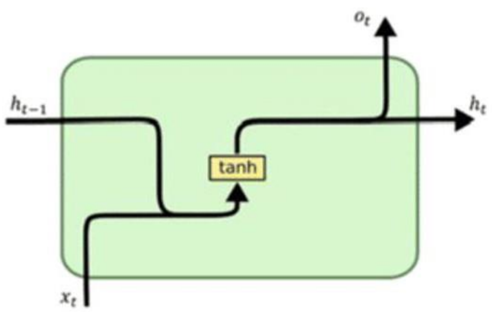


Fig. 4. RNN architecture [32]

3.4. Model Training

An approach was adopted to train and validate learning models for accurate stock predictions based on historical data. The approach included dividing the dataset, tuning hyperparameters, and optimizing the models.

3.4.1. Dataset Segmentation

In this research, the dataset was divided into three parts based on time. The first part was the Training Set, covering data from January 1, 2007, to December 31, 2016. This set formed the foundation for training the models and constituted 70% of the dataset. Furthermore, the second part was the Validation Set, including data from January 1, 2017, to December 31, 2018. This set was crucial for validating the models, allowing adjustments as well as optimization of parameters without using the test data and this Validation Set represented around 15% of the dataset. Finally, the Testing Set was spanning from January 1, 2019, to December 31, 2020, which evaluated how well the model performed with data. Test datasets were essential for assessing the performance of the developed models.

3.4.2. Tuning Hyperparameters

For the finding, combinations of hyperparameters were experimented with, specifically examining the effects of epochs, learning rate, and timesteps. The major metrics used to assess the models were MSE, RMSE, and R^2-score.

3.4.2.1. Epochs

Throughout the findings, different numbers of epochs were used for the experiment to determine the optimal duration for training the models without encountering overfitting issues. The team conducted tests with epoch values ranging from 50 to 250.

3.4.2.2. Learning Rate

The learning rate played a crucial role in determining how quickly the model absorbed information. In the research, the team tested various learning rate values, such as 0.0001, 0.001, 0.01, and 0.1. The aim was to strike a balance between convergence and model stability.

3.4.2.3. Timesteps

Timesteps were significant in training a stock prediction model and throughout the research, the team explored timestep values corresponding to trading day intervals. The values included 1, 2, 5, 10, and even longer periods such as 120 days or more, depending on the dataset available at each time point considered for analysis. In addition, these variations aimed at evaluating how the length of historical data influenced the accuracy of predictions made by the model.

3.4.3. Optimization

In this research, Adam Optimizer method was used because it could adjust the learning rate during training, aiding the model in converging quickly. Additionally, the team decided to use the MSE as the loss function to minimize differences between the predictions of the model and the actual values.

3.5. Model Evaluation

Evaluation constituted a crucial phase in the model development lifecycle. This phase aimed to assess the performance of the model against unfamiliar data, ensuring that the trained model possessed commendable generalization capabilities and was not merely repeating the training data. In the research, the team used several evaluation metrics commonly leveraged in regression analysis.

3.5.1. Mean Square Error (MSE)

MSE represented the average of the squared differences between the original value and the predicted value. The mathematical equation was,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (2)$$

where y_i represented the original value and $\hat{y}_i$ represented the predicted value for the ith sequence in which the MSE value reduces, the resulting model increases.

3.5.2. Root Mean Squared Error (RMSE)

RMSE was the square root of MSE, providing a calculation of prediction error in the same dimensions as the target variable.

$$RMSE = \sqrt{MSE} \quad (3)$$

A low RMSE value showed that the error between the prediction and the original value tended to be small, while a large RMSE value indicates otherwise.

3.5.3. R^2-Score

The R^2-score was used to measure how well the variation in the target variable could be explained by the model, ranging from 0 to 1. An R^2 close to 1 showed that the created model could explain significant variations in the target variable,

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (4)$$

where $\bar{y}$ represented the average of the target variable.

By integrating three evaluation metrics, a comprehensive understanding of the performance of the model developed in this research when predicting previously unseen data was achieved. Furthermore, these metrics collectively ensured that the model showed stable and reliable performance.

3.6. Comparison and Analysis

At the stage of the research, the performance of three models namely LSTM, GRU, and RNN was compared to determine the most influential architecture for stock predictions based on historical data.

3.7. Model used

The model applied followed the steps described in the preceding section, as shown in Figure 5. Additionally, the architecture of the model was shown in Table 1.

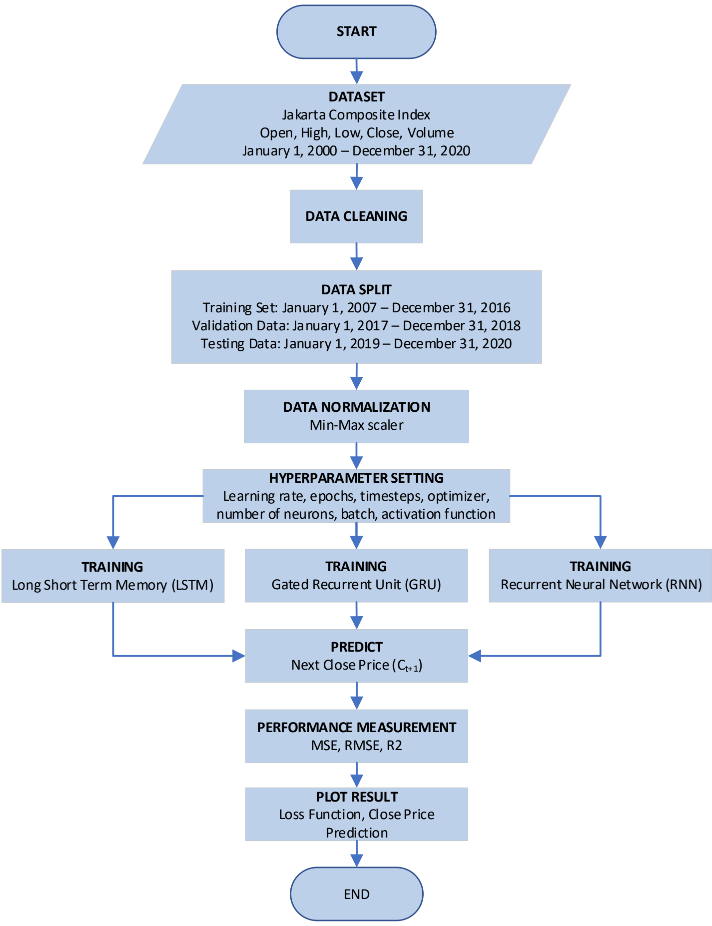


Fig. 5. Methodology used in research

Table 1. Model Architecture

Model	Parameter	Value
LSTM	Number of Hidden Layers	2
	Number of neurons per layer	45 and 45
	Learning rate	0.01, 0.001, 0.0001
	Epochs	50, 100, 150, 200, and 250
	Timesteps	1, 2, 5, 10, 20, 30, 40, 50, 60, 80, and 120
	Optimizer	Adam
	Batches	64
GRU	Number of hidden layers	2
	Number of neurons per layer	25 and 25
	Learning rate	0.1, 0.01, 0.001, 0.0001

	Epochs	50, 100, 150, 200, and 250
	Timesteps	1, 2, 5, 10, 20, 30, 40, 50, 60, 80, and 120
	Optimizer	Adam
	Batches	64
RNN	Number of hidden layers	2
	Number of neurons per layer	25 and 25
	Learning rate	0.1, 0.01, 0.001, 0.0001
	Epochs	50, 100, 150, 200, and 250
	Timesteps	1, 2, 5, 10, 20, 30, 40, 50, 60, 80, and 120
	Optimizer	Adam
	Batches	64

4. Results and Discussion

In the current era of advancements in computational science, predictions about stocks using deep learning models have received significant attention among specialists and practitioners. Three models which were GRU, LSTM, and RNN frequently distinct in academic discussions. This research aimed to compare these three models in the context of stock predictions, with particular emphasis on forecasts for IDX.

4.1. Model Training Results

After training, the three models showed convergence, although with varying degrees of loss. Based on the evaluation, RNN model outperformed the others, consistently yielding the highest R^2-scores across numerous scenarios. While LSTM and GRU also produced commendable outcomes in certain conditions, the models did not match the prowess of RNN. This section conducted tests with diverse variations of epochs, learning rates, and timesteps for each model.

4.1.1. Epoch Variations

After conducting tests using GRU model with a learning rate of 0.001, timesteps of 40, and a batch size of 64, it was observed that increasing the number of epochs from 50 to 100 significantly improved the R^2-score from 0.906 to 0.985. However, further increases to 150 and 250 epochs only led to slight improvements. In some cases, extending the number of epochs led to a decrease in the R^2-score. To address this issue, the examiners implemented early stopping for epoch values, showing potential overfitting. Table 2 showed a sum up of the test outcomes for variations in epochs.

Table 2. Test Results with Varying Epochs

Model	Epoch	Early Stopping	MSE	RMSE	R^2-score
GRU	50		0.002074	0.045541	0.906245
GRU	100		0.000325	0.018039	0.985289
GRU	150	102	0.000357	0.018907	0.983841
GRU	250	113	0.000342	0.018491	0.984544
LSTM	50		0.003447	0.058707	0.844197
LSTM	100		0.001576	0.039696	0.928765
LSTM	150		0.000319	0.017863	0.985575
LSTM	200		0.002949	0.054307	0.866675
RNN	50		0.000865	0.029405	0.960969
RNN	100	83	0.000591	0.02432	0.973303
RNN	150	96	0.000302	0.017381	0.986364
RNN	250	103	0.000531	0.02305	0.976017

In experiments using LSTM model with an identical learning rate, timesteps, and batch size, an increase in the number of epochs from 50 to 150 led to an improvement in the R^2-score from 0.844 to 0.985. However, when the number of epochs was adjusted to 200, there was a decline in performance.

Similar patterns were observed in tests conducted with RNN model where the R^2-score saw an increase when the number of epochs was adjusted from 50 to 150. However, further increasing the epochs led to a level in performance and these findings signified the importance of determining the number of epochs. Insufficient epochs could lead to learning by the model, while an excessive count might lead to overfitting without implementing an early stopping mechanism.

4.1.2. Variations in Learning Rate

The learning rate was a critical factor in training artificial neural networks, controlling the size of steps taken during optimization to update weights. A well-selected learning rate sped up convergence and improved prediction accuracy. On the contrary, an inappropriate learning rate hindered convergence or even led to divergence. The outcomes of tests for the three models across different learning rates were shown in Table 3.

Table 3. Test results with variations in learning rate

Model	Learning Rate	Time-steps	MSE	RMSE	R^2-score
GRU	0.1	40	0.022217	0.149053	-0.00434
GRU	0.01	40	0.0005	0.022366	0.977386
GRU	0.001	40	0.00049	0.022147	0.977828
GRU	0.0001	40	0.000608	0.024658	0.972514
LSTM	0.01	40	0.020018	0.141484	0.095083
LSTM	0.001	40	0.001284	0.035826	0.941977
LSTM	0.0001	40	0.003767	0.061375	0.829713
RNN	0.1	40	0.207811	0.455863	-8.38041
RNN	0.01	40	0.000408	0.020204	0.981574
RNN	0.001	40	0.000408	0.020198	0.981585
RNN	0.0001	40	0.000318	0.017846	0.985625

In experiments with GRU model, a learning rate of 0.0001 showed performance, although the R^2-score did not reach its full potential. However, increasing the learning rate to 0.1 significantly harmed the effectiveness of the model, leading to a low R^2-score. The hindrance was crucial to avoid using an extreme learning rate, as it could hinder the model from finding its best solution and eventually degrade performance. Standardizing the learning rate to 0.01 produced outcomes from GRU model, but not at its optimal state. The preferable choice for this model seemed to be a learning rate of 0.001, leading to an R^2-score close to 0.98.

In the case of LSTM model tests, a learning rate of 0.0001 produced suboptimal results. Opting for a value such as 0.01 did not benefit this model, leading to a decreased R^2-score. However, using a learning rate of 0.001 for LSTM models achieved performance with the highest recorded R^2-score around 0.941977.

Moving on to RNN model, a learning rate of 0.001 yielded outcomes, but there was room for improvement in results. Conversely, setting the learning rate at 0.1 led to a divergence in RNN performance, as shown by its negative R^2-score. Therefore, adjusting and fine-tuning with a value such as 0.0001 as the learning rate for our RNN models yielded performance and significant results.

The learning rate played a role in determining how a model converged without a doubt. Considering this dataset and the three models that were assessed, it seemed that a learning rate of 0.001 would be the optimal option. Tests were essential to be conducted across learning rate values because certain models might react strongly to this parameter. However, the ideal learning rate could vary depending on the properties of the data and specific model architecture details.

4.1.3. Variation of Timesteps

Timesteps represented the number of preceding days used for forecasting stock movements in stock predictions. Selecting appropriate timesteps significantly affected model performance, determining the extent of historical data considered during predictions. Figure 6 showed the variation in R^2-score values for the three models evaluated, plotted against different timesteps.

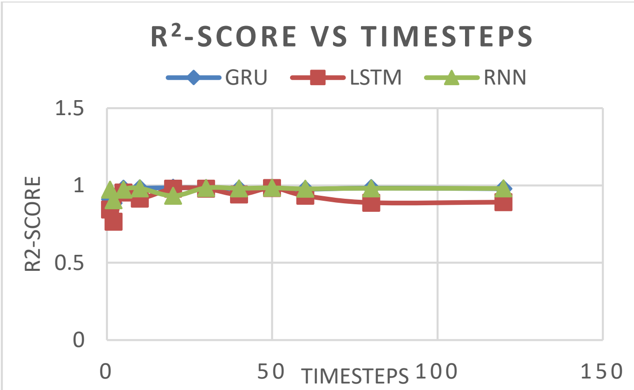


Fig. 6. Relationship between R^2-score and Timestep for GRU model. LSTM and RNN

Using GRU model, a timestep of 40 consistently led to a high R^2-score, and adjusting the timesteps to 10 or extending to 60 produced varied outcomes but still maintained commendable performance. This recommended that GRU was proficient in discerning patterns across different timeframes. However, opting for timesteps of 1 or 2 significantly reduced the effectiveness of the model, showing that revisions were needed for accurate predictions in such brief trading durations. Extending timesteps to 80 or 120 improved the outcomes, with the R^2-score proposing the proficiency of the model in assimilating information throughout extended durations.

Tests with LSTM model followed a similar trend where a timestep of 40 became prudent, delivering an R^2-score close to 0.986. Modifying timesteps to 10 and 60 maintained strong performances and extending to 80 and 120 reduced the R^2-score, signifying limited adaptability of LSTM to different historical timeframes.

Experiments with RNN model supported that timestep variations influenced predictive outcomes. Using RNN model with a timestep of 40 showed the resilience of the model to timestep modifications. However, such changes in RNN model had minimal impact on the fluctuation of the R^2-score.

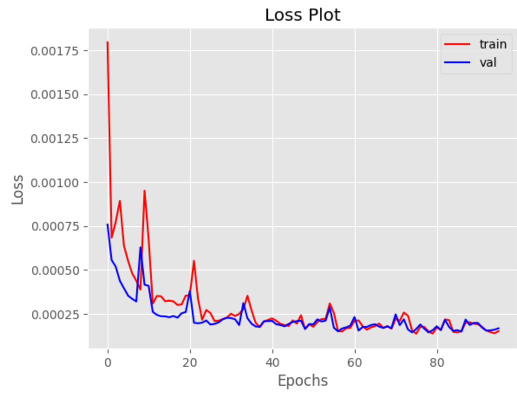
In total, timestep variations influenced the performance of all three models, and excessively brief timesteps were needed to provide more information for the model. Conversely, excessively extended timesteps might have needed to support more with the model and decelerate training. For the Jakarta Composite Index dataset, a timestep of around 40 struck a harmonious balance between sufficient historical data and manageable model complexity.

4.2. Model Evaluation

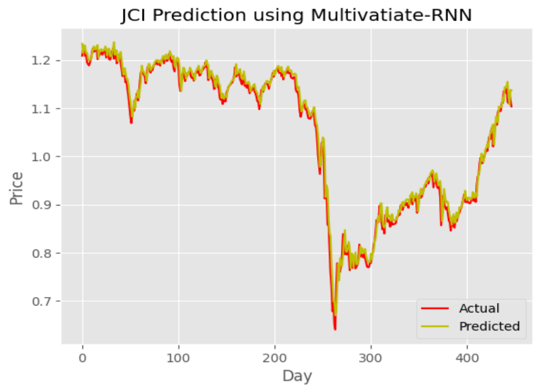
Table 4. Review of the 10 best models tried

Model	Learning rate	Time-steps	E-pochs	MSE	RMSE	R^2-score
RNN	0.001	40	150	0.0003	0.01738	0.986364
RNN	0.0001	40	200	0.00032	0.01785	0.985625
LSTM	0.001	40	150	0.00032	0.01786	0.985575
GRU	0.001	40	100	0.00033	0.01804	0.985289
GRU	0.001	20	200	0.00033	0.01818	0.985286
RNN	0.001	50	200	0.00034	0.01836	0.984694
RNN	0.001	30	200	0.00034	0.01855	0.984578
GRU	0.001	40	250	0.00034	0.01849	0.984544
GRU	0.001	30	200	0.00036	0.0189	0.983986
GRU	0.001	40	150	0.00036	0.01891	0.983841

Based on the conducted tests, it was found that RNN model performed the best with an RMSE value of 0.01738. LSTM model closely followed with an RMSE of 0.01786, and GRU model performed slightly worse, with an RMSE of 0.01804. When considering the R^2-score, both RNN and LSTM models achieved a score of 0.986, while GRU model achieved a lower score of 0.985. Although RNN-based model excelled in terms of its R^2-score, it was worth noting that GRU method showed stability across timestep variations. All three models accurately forecasted the Jakarta Composite Index in this research. Table 4 showed details on the ten models explored during our finding, while Figures 7 to 9 visually showed how each evaluated RNN, LSTM, and GRU model performed in terms of the loss function and optimal JCI prediction outcomes.

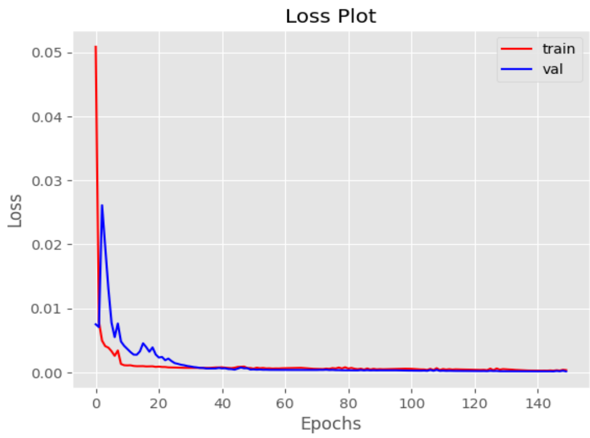


(a)

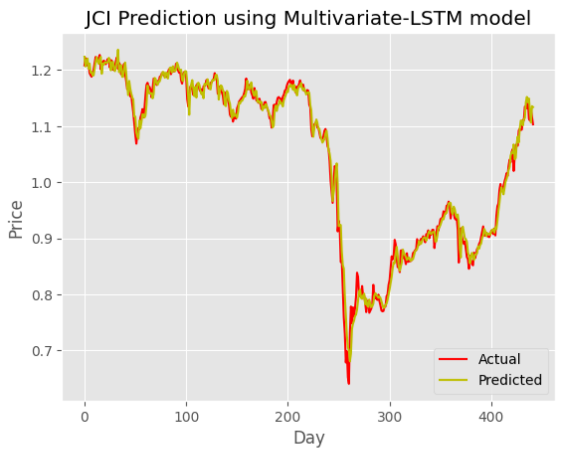


(b)

Fig 7. (a) Loss Function and (b) JCI Prediction Using RNN

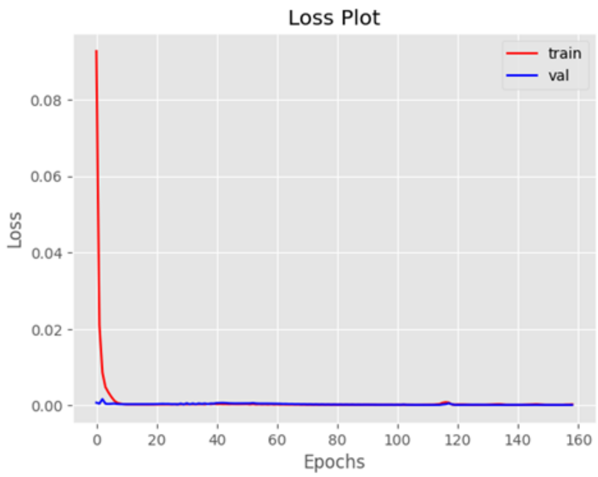


(a)

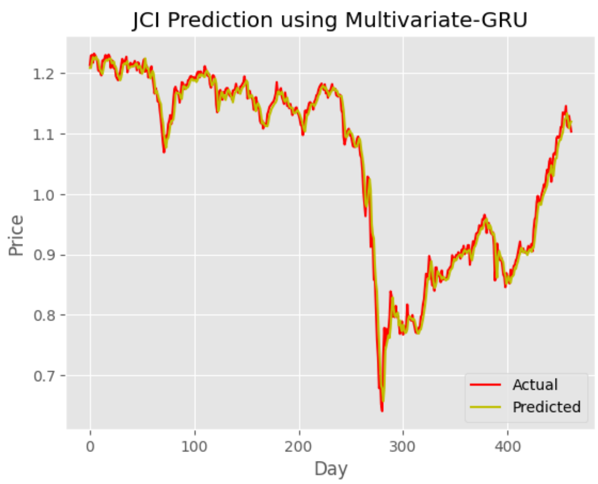


(b)

Fig. 8. (a) Loss Function and (b) JCI Prediction Using LSTM



(a)



(b)

Fig. 9. (a) Loss Function and (b) JCI Prediction Using GRU

4.3. Discussion

The research included creating a model using three deep learning techniques that used historical stock data, including low, close, and volume prices. This data proved valuable for predicting stock prices, and providing information for forecasts.

The primary focus of this finding was to analyze and compare the performance of three variations of RNN namely, GRU, LSTM, and RNN. By evaluating the R^2-score, the metric for assessing performance, distinct trends were observed in each performance of the model. In total, RNN model performed well, achieving an impressive R^2-score of 0.98 or higher in specific configurations, showing its adaptability to the dataset. LSTM model also showed R^2-scores in various setups but indicated more variability in its performance. While the model approached an R^2-score near 0.985 in some scenarios, LSTM model lacked performance in others. On the other hand, GRU model consistently delivered results with an R^2-score range between 0.88 and 0.98, across different conditions.

Each deep learning architecture explored in this research had its strengths and limitations. While RNN became the leading model based on the achieved R^2-score, it was essential not to rely solely on a metric when determining the best model. Factors such as model complexity, training time, and interpretability of results should also have been considered. Based on this discussion, it was crucial to acknowledge the limitations of this research, which primarily focused on data. Various factors influenced stock prices, such as news sentiment analysis, fundamental analysis, and macroeconomic indicators. The area of deep learning architectures was still largely unexplored and combining models through different approaches might have produced better results.

Based on the metrics gathered in this research, it was possible to develop a prediction system for stock indices. However, investors should have exercised caution because built models could occasionally fail to predict sudden market shifts or unforeseen events. While this method provided a tool, cautious decision-making remained crucial in stock investments.

5. Conclusion and Future Work

In conclusion, the three models used in this research show the ability to predict stock indices effectively. Each model provided unique forecasts for the Jakarta Composite Index with varying levels of accuracy. Among the models, RNN was outstanding as the most adept for JCI predictions, closely followed by GRU. Moreover, LSTM seemed less optimal for long-term predictions.

Further than merely comparing different deep learning architectures, there was interesting potential in improving the dataset with additional features. These could include technical indicators, refined stock data, and sentiment analysis of market news. Additionally, the introduction of regularization techniques, such as dropout and weight decay, could act as effective measures against overfitting. Implementing these regularization strategies could further improve the ability of the model to generalize.

The improvements recorded could provide a more thorough understanding of the outcomes as well as fundamental processes. However, it tended to offer a distinctive context regarding the performance of each model, particularly in the field of stock predictions using IDX stock dataset.

Acknowledgment

I would like to express my sincere gratitude to Universitas Atma Jaya Yogyakarta for funding my research and studies in the Doctoral Program Department of Electrical Engineering and Information Technology, Universitas Gadjah Mada Yogyakarta.

Author contributions

All authors contributed equally as the main contributor of this paper. All authors read and approved the final paper.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] A. H. Manurung, W. Budiharto, and H. Prabowo, "Algorithm and modeling of stock prices forecasting based on long short-term memory (LSTM)," *ICIC Express Lett.*, vol. 12, no. 12, pp. 1277–1283, 2018, doi: 10.24507/icicel.12.12.1277.

[2] M. Sharaf, E. E. D. Hemdan, A. El-Sayed, and N. A. El-Bahnasawy, "StockPred: a framework for stock Price prediction," *Multimed. Tools Appl.*, vol. 80, no. 12, pp. 17923–17954, 2021, doi: 10.1007/s11042-021-10579-8.

[3] X. Zhang, N. Gu, J. Chang, and H. Ye, "Predicting stock price movement using a DBN-RNN," *Appl. Artif. Intell.*, vol. 35, no. 12, pp. 876–892, 2021, doi: 10.1080/08839514.2021.1942520.

[4] W. Khan, M. A. Ghazanfar, M. A. Azam, A. Karami, K. H. Alyoubi, and A. S. Alfakeeh, "Stock market prediction using machine learning classifiers and social media, news," *J. Ambient Intell. Humaniz. Comput.*, vol. 13, no. 7, pp. 3433–3456, 2022, doi: 10.1007/s12652-020-01839-w.

[5] S. K. Chandar, "Convolutional neural network for stock trading using technical indicators," *Autom. Softw. Eng.*, vol. 29, no. 1, pp. 1–14, 2022, doi: 10.1007/s10515-021-00303-z.

[6] H. Liu and Z. Long, "An improved deep learning model for predicting stock market price time series," *Digit. Signal Process. A Rev. J.*, vol. 102, p. 102741, 2020, doi: 10.1016/j.dsp.2020.102741.

[7] Z. Jin, Y. Yang, and Y. Liu, "Stock closing price prediction based on sentiment analysis and LSTM," *Neural Comput. Appl.*, vol. 32, no. 13, pp. 9713–9729, 2020, doi: 10.1007/s00521-019-04504-2.

[8] S. Das and S. Mishra, "Advanced deep learning framework for stock value prediction," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 10, pp. 2358–2367, 2019, doi: 10.35940/ijitee.B2453.0881019.

[9] Y. Li and Y. Pan, "A novel ensemble deep learning model for stock prediction based on stock prices and news," *Int. J. Data Sci. Anal.*, vol. 13, no. 2, pp. 139–149, 2022, doi: 10.1007/s41060-021-00279-9.

[10] T. T. Nguyen and S. Yoon, "A novel approach to short-term stock price movement prediction using transfer learning," *Appl. Sci.*, vol. 9, no. 22, 2019, doi: 10.3390/app9224745.

[11] M. Smorada, A. Lukáčková, Z. Hajduová, Ľ. Šrenkel, and J. Havier, "Predictive Financial Analysis as a Tool for Business Management," *TEM J.*, vol. 12, no. 1, pp. 147–155, 2023, doi: 10.18421/TEM121-20.

[12] X. Ji, J. Wang, and Z. Yan, "A stock price prediction method based on deep learning technology," *Int. J. Crowd Sci.*, vol. 5, no. 1, pp. 55–72, 2021, doi: 10.1108/IJCS-05-2020-0012.

[13] M. Jarrah and N. Salim, "A recurrent neural network and a discrete wavelet transform to predict the Saudi stock price trends," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 4. pp. 155–162, 2019. doi: 10.14569/ijacsa.2019.0100418.

[14] Y. F. Lin, T. M. Huang, W. H. Chung, and Y. L. Ueng, "Forecasting Fluctuations in the Financial Index Using a Recurrent Neural Network Based on Price Features," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 5, no. 5, pp. 780–791, 2021, doi: 10.1109/TETCI.2020.2971218.

[15] U. Gupta, V. Bhattacharjee, and P. S. Bishnu, "StockNet—GRU based stock index prediction," *Expert Syst. Appl.*, vol. 207, no. April, p. 117986, 2022, doi: 10.1016/j.eswa.2022.117986.

[16] D. Lien Minh, A. Sadeghi-Niaraki, H. D. Huy, K. Min, and H. Moon, "Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network," *IEEE Access*, vol. 6. pp. 55392–55404, 2018. doi: 10.1109/ACCESS.2018.2868970.

[17] K. Kumar and D. P. Gandhmal, "An intelligent indian stock market forecasting system using LSTM deep learning," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 21, no. 2, pp. 1082–1089, 2020, doi: 10.11591/ijeecs.v21.i2.pp1082-1089.

[18] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2. pp. 654–669, 2018. doi: 10.1016/j.ejor.2017.11.054.

[19] Y. Baek and H. Y. Kim, "ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a

prediction LSTM module," *Expert Systems with Applications*, vol. 113. pp. 457–480, 2018. doi: 10.1016/j.eswa.2018.07.019.

[20] S. Borovkova and I. Tsiamas, "An ensemble of LSTM neural networks for high-frequency stock market classification," *J. Forecast.*, vol. 38, no. 6, pp. 600–619, 2019, doi: 10.1002/for.2585.

[21] J. Luo, G. Zhu, and H. Xiang, "Artificial Intelligent based day-ahead stock market profit forecasting," *Comput. Electr. Eng.*, vol. 99, no. February, p. 107837, 2022, doi: 10.1016/j.compeleceng.2022.107837.

[22] H. Y. Kim and C. H. Won, "Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models," *Expert Systems with Applications*, vol. 103. pp. 25–37, 2018. doi: 10.1016/j.eswa.2018.03.002.

[23] X. Pang, Y. Zhou, P. Wang, W. Lin, and V. Chang, "An innovative neural network approach for stock market prediction," *J. Supercomput.*, vol. 76, no. 3, pp. 2098–2118, 2020, doi: 10.1007/s11227-017-2228-y.

[24] A. Q. Md *et al.*, "Novel optimization approach for stock price forecasting using multi-layered sequential LSTM," *Appl. Soft Comput.*, vol. 134, p. 109830, Feb. 2023, doi: 10.1016/j.asoc.2022.109830.

[25] M. Nabipour, P. Nayyeri, H. Jabani, S. Shahab, and A. Mosavi, "Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; A Comparative Analysis," *IEEE Access*, vol. 8. pp. 150199–150212, 2020. doi: 10.1109/ACCESS.2020.3015966.

[26] T. Swathi, N. Kasiviswanath, and A. A. Rao, "An optimal deep learning-based LSTM for stock price prediction using twitter sentiment analysis," *Appl. Intell.*, pp. 13675–13688, 2022, doi: 10.1007/s10489-022-03175-2.

[27] W. Budiharto, "Data science approach to stock prices forecasting in Indonesia during Covid-19 using Long Short-Term Memory (LSTM)," *J. Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00430-0.

[28] A. S. Saud and S. Shakya, "3-Way Gated Recurrent Unit Network Architecture for Stock Price Prediction," *Indian J. Comput. Sci. Eng.*, vol. 12, no. 2, pp. 421–427, 2021, doi: 10.21817/indjcse/2021/v12i2/211202011.

[29] C. Chen, L. Xue, and W. Xing, "Research on Improved GRU-Based Stock Price Prediction Method," *Appl. Sci.*, vol. 13, no. 15, 2023, doi: 10.3390/app13158813.

[30] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, and S. Shahab, "Deep learning for stock market prediction," *Entropy*, vol. 22, no. 8, 2020, doi: 10.3390/E22080840.

[31] M. Li, Y. Zhu, Y. Shen, and M. Angelova, "Clustering-enhanced stock price prediction using deep learning," *World Wide Web*, 2022, doi: 10.1007/s11280-021-01003-0.

[32] D. Satria, "Predicting Banking Stock Prices Using Rnn, Lstm, and Gru Approach," *Appl. Comput. Sci.*, vol. 19, no. 1, pp. 82–94, 2023, doi: 10.35784/acs-2023-06.