Journal of Infrastructure, Policy and Development (2023) Volume 7 Issue 3.
DOI: 10.24294/jipd.v7i3.2631

ORIGINAL ARTICLE

Development of stock price prediction system using Flask framework and LSTM algorithm

***Kefas Bagastio*[1], *Raymond Sunardi Oetama*[1], *Arief Ramadhan*[2,*]**

[1] *Department of Information Systems, Faculty of Engineering and Informatics, Universitas Multimedia Nusantara (UMN), Tangerang, 15811, Indonesia*

[2] *School of Computing, Telkom University, Bandung, 40257, Indonesia*

ABSTRACT

Stock investment in Indonesia has been steadily growing in the past five years, offering profit potential alongside the risk of loss. Stockholders must analyze the stocks they intend to purchase. Stockholders often analyze stocks by observing patterns that occurred in the previous days to predict future prices. Therefore, a method is needed to simplify the process of analyzing the stock pattern. Although there are already several websites that have the concept of predicting stock prices, these websites do not utilize deep learning algorithms. This research aims to develop a stock price prediction website using deep learning algorithms, specifically the Long Short-Term Memory (LSTM) algorithm to help users predict stock prices. This research focuses on five banks with the highest market capitalization in Indonesia, namely Bank Central Asia, Bank Rakyat Indonesia, Bank Mandiri, Bank Negara Indonesia, and Bank Syariah Indonesia. The website utilizes Flask framework and LSTM. Flask is used to apply LSTM model to the website, while the LSTM can capture long-term dependencies in high-complexity data. The result of this research is a stock price prediction website application, where the prediction results are displayed through the website. The LSTM model for each stock has a Mean Absolute Percentage Error (MAPE) of less than 10%, which indicates that the model is "Highly accurate" based on the MAPE accuracy scale judgment.

KEYWORDS

stocks; prediction; Flask framework; Long Short-Term Memory algorithm; Indonesia

ARTICLE INFO
Received: 22 August 2023
Accepted: 13 September 2023
Available online: 11 October 2023

*CORRESPONDING AUTHOR
Arief Ramadhan,
School of Computing, Telkom University, Bandung, 40257, Indonesia
arieframadhan@telkomuniversity.ac.id

CITATION
Bagastio K, Oetama RS, Ramadhan A (2023). Development of stock price prediction system using Flask framework and LSTM algorithm. *Journal of Infrastructure, Policy and Development* 7(3): 2631. doi: 10.24294/jipd.v7i3.2631

1. Introduction

Shares are a form of ownership document. The more stocks a stockholder owns, the more power they have over the company (Ilham et al., 2022). In the field of finance, shares are the divided property of a company that gives the possibility of the right to receive the company's income and vote at the stockholders' meeting (Guha et al., 2019). Stockholders can easily buy shares and benefit significantly from dividends provided in the company's bonus program for stockholders (Rahmawati and Garad, 2023). In Indonesia, stock investment is increasingly in demand by many people.

Kustodian Sentral Efek Indonesia (KSEI) as one of capital market authorities in Indonesia noted that the number of stockholders in the Indonesian capital market has exceeded 4 million. Based on KSEI data at the end of the first half of 2022, the number of Single Investor Identification (SID) has reached 4,002,289, with 99.79% being local individual stockholders. Looking at the growth, since 2021 the number of stockholders has increased by 15.96% starting from 3,451,513 at the end of 2021 to 4,002,289 at the end of June 2022. The upward trend has been visible since 2020 when stockholders still numbered 1,695,268 (Statistik Pasar Modal Indonesia, 2023). In Indonesia Financial sector stocks have a high popularity and are highly demanded by stockholders, as mentioned in the press release which states that the number of investors in the financial sector reached 209,053 and 481,197 for the Gen Z and Millennial generations, respectively (Ramyakim and Widyasari, 2022).

Based on data from KSEI which states that the number of stockholders has been increasing, stock investment in Indonesia has increased. However, stockholders need to remember that besides providing potential profits, stock investment also has the risk of loss. Therefore, stockholders must analyze the shares to be purchased (Untoro, 2021). Stockholders often rely on analyzing stocks by paying attention to patterns that occurred on the previous day. In general, keeping track of stock prices is difficult because it does not follow a specific pattern, or it frequently changes. The pattern analysis used now can become out of date in a short period of time (Prasad and Seetharaman, 2021). Therefore, there is a need for a method that can simplify the process of analyzing stock patterns. With a more sophisticated method, shareholders will have the opportunity to recognize patterns that may have been missed before, identify potential trends, and make more knowledgeable investment decisions.

Based on the mentioned phenomenon, this research is aimed to develop a stock price prediction method by utilizing a deep learning algorithm. This research focuses on five banks with the highest market cap in Indonesia. According to the data gathered from yahoo finance, the five banks with the highest market cap in Indonesia are Bank Central Asia (BBCA), Bank Rakyat Indonesia (BBRI), Bank Mandiri (BMRI), Bank Negara Indonesia (BBNI), and Bank Syariah Indonesia (BRIS). Stocks were selected based on the highest market cap because stocks with high market caps tend to represent large and well-known companies in a particular industry. Moreover, high market cap stocks also tend to have adequate data availability (Kumar and Kumara, 2020). Therefore, researching the stock price movements of companies with the highest market cap can provide easier access to relevant data and information for analyzing and predicting stock prices.

Over the past few decades, there have been algorithms related to machine learning and deep learning implemented in the field of finance (Hastomo et al., 2021), one of which is used to analyze historical stock price data to make predictions by utilizing stock movements recorded on websites that provide historical stock price data such as yahoo finance (Prasetya et al., 2019). There are several machine learning or deep learning algorithms that can be used to make predictions, including Autoregressive integrated moving average (ARIMA), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Support Vector Regression (SVR), Facebook Prophet, and many more

(Bathla, 2020; Siami-Namini et al., 2022; Meshram et al., 2022).

Based on this phenomenon, several studies have been made in the past that propose solutions for predicting stock prices. The first solution is found in a journal that compares the use of the LSTM algorithm with SVR performed by Bathla (2020). Through that research, both algorithms were tested for their performance, and the results showed that LSTM had superior performance (Bathla, 2020). Furthermore, there is another journal that compares LSTM with ARIMA performed by Siami-Namini et al. (2018). The study tested both algorithms and the results showed that LSTM has better performance than ARIMA (Siami-Namini et al., 2022). Another solution is designing a stock price prediction website using phyton conducted by Meshram et al. (2022).

Based on the solutions that have been proposed by several previous studies, this research combine the use of the LSTM model with website design using the Flask framework. Currently, no journal combines these two elements to predict the five Indonesian banks with the highest market cap, namely Bank Central Asia (BBCA), Bank Rakyat Indonesia (BBRI), Bank Mandiri (BMRI), Bank Negara Indonesia (BBNI), and Bank Syariah Indonesia (BRIS). The LSTM model is selected based on previous research that indicated superior performance when compared to other models (Bathla, 2020; Siami-Namini et al., 2022). Although in previous research the Streamlit framework was used, this research use Flask framework because of its flexibility and ability to build more complex websites. In this research, the LSTM model that has been built is stored as a pre-trained model and stored in the database, then the pre-trained model stored in the database is used to make predictions. Therefore, Flask is the most suitable choice as a backend framework to access data from the database and use the model to make predictions. This research aims to provide knowledge related to the integration of the LSTM machine learning model into website applications using the Flask framework and help to analyze stock prices through the predictions that are obtained from patterns that have happened in the past.

2. Methodology

This study used LSTM as one of Deep Learning algorithms. Deep learning is a concept within machine learning that relies on the use of artificial neural networks (Janiesch et al., 2021). It is important to acknowledge another distinguishing aspect, which is the presence of multiple layers between the input and the output in deep learning. These layers can consist of numerous neural units, sometimes reaching several hundred or even thousand (Dong et al., 2021). Deep learning is a powerful tool for processing unstructured data because it can learn and make intelligent decisions on its own. It is capable of processing large amounts of data and features, which makes it an ideal solution for dealing with unstructured data (Kyin et al., 2020).

LSTM is a variant of the recurrent neural network (RNN) that is more suitable for processing sequential data (Xia et al., 2020). The architecture of LSTM can be seen in **Figure 1**. LSTM consists of a set of recurrently connected sub-networks, known as memory blocks. The idea behind memory blocks is to maintain their state over time while regulating the flow of information through non-linear gating units. We can see in **Figure 1** the architecture of the LSTM algorithm, which involves gates, Input Signal x(t) and output y(t), activation functions, and peephole connections. The output of the block will be repeatedly connected back to the input block and to all gates (Van Houdt et al., 2020).

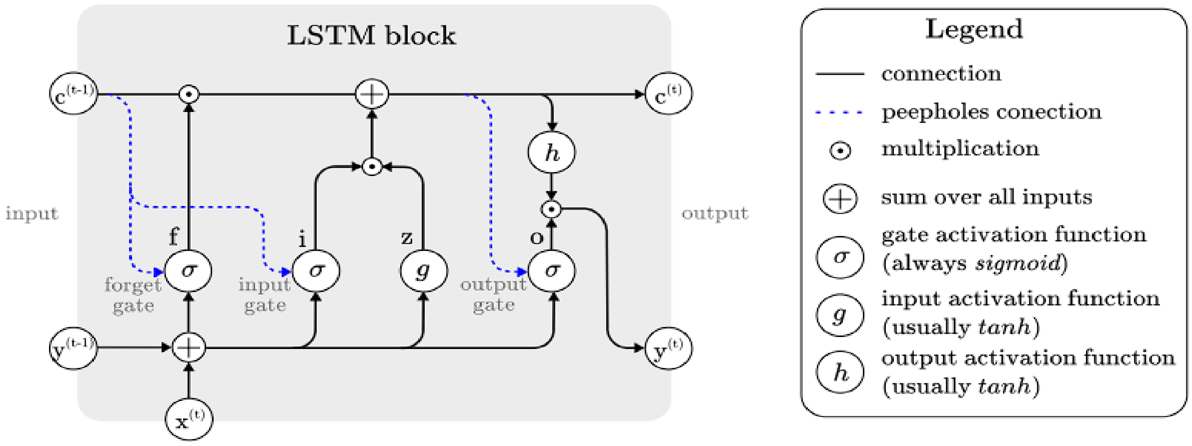

Figure 1. The architecture of LSTM (Van Houdt et al., 2020).

LSTM is capable of prediction, pattern classification, different types of recognition, and sequence generation (Smagulova and James, 2019). Sequence data refers to a collection of values or events that occur in a specific order, such as historical stock data. The structure of LSTM involves a series of interconnected sub-networks called memory blocks. These memory blocks are designed to retain their state over time and control the flow of information using non-linear gating units (Van Houdt et al., 2020).

This study also used Flask framework when building the system. The architecture of Flask Framework is shown in **Figure 2**. We modified this architecture from the architecture used in (Liang et al., 2021). By using the Flask framework, users will perform interactions, such as providing input and receiving output, through the "views" component. When a user makes input in the "views" component, the input will then be forwarded to the "service" component to be processed based on the input made by the user. Next, the "service" component will interact with the "models" component to access data in the "database" component. After the "service" component completes the process, the results will be sent back to the "views" component to be provided as output to the user. The "Template" component in **Figure 2** is used to build a user interface that includes elements such as the footer, navigation bar, and layout of the website.

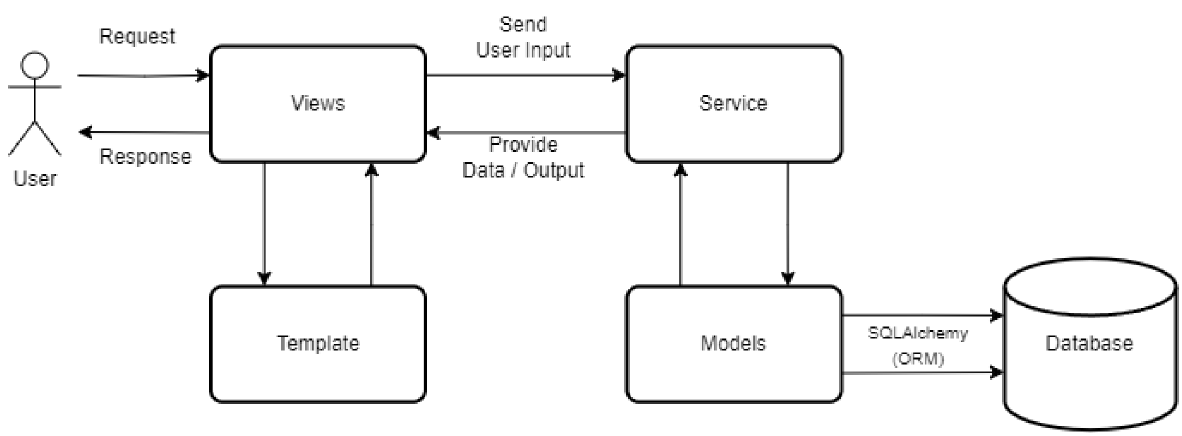

Figure 2. The architecture of Flask framework.

Flask is a Python web framework that offers a set of tools and pre-written code to facilitate website development, without the need to do everything from scratch. We chose Flask because Flask is a micro-framework that does not contain many tools and libraries, so it is more portable and does not utilize a lot of resources (Anggoro and Aziz, 2021). With its streamlined features, Flask is lightweight and doesn't rely heavily on numerous external libraries (Mufid et al., 2021).

The development process of the system adopts prototyping method as explained in (Hamdani et al., 2021). The system development stage using the prototyping method can be seen from the diagram shown in **Figure 3**.

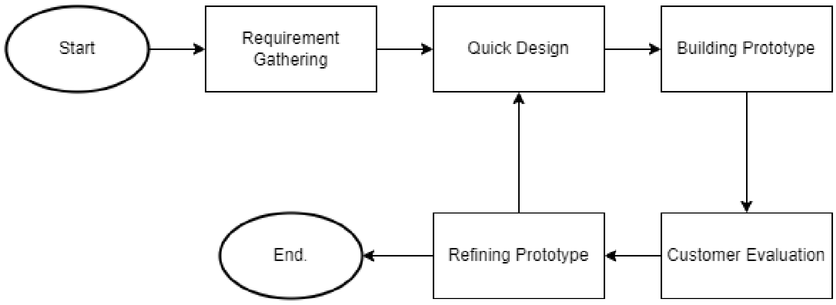

Figure 3. Prototyping method (Hamdani et al., 2021).

The data used in this study is obtained from Yahoo Finance, which then be predicted using deep learning algorithms. Yahoo Finance website provides historical stock price data (Yahoo finance-stock market live, quotes, business & finance news). The data is obtained using the yfinance library, which simplifies the process of retrieving data from Yahoo finance website. In this study, the data utilized consists of the closing stock prices of the top 5 Indonesian banks with the highest market capitalization. These banks include Bank Central Asia (BBCA), Bank Rakyat Indonesia (BBRI), Bank Mandiri (BMRI), Bank Negara Indonesia (BBNI), and Bank Syariah Indonesia (BRIS). The data period used to train the LSTM model is daily data from May 2018 to May 2023 (5 years).

In this study, there are independent variables and dependent variables. Independent variables are the cause variables or variables that can affect the changes of the other variables (Hamdani et al., 2021). The independent variable in this research is closing stock price data which are retrieved from the Yahoo finance website. Meanwhile, the dependent variable is a variable affected by the independent variable, therefore the dependent variable will respond to changes in the independent variable (Rachma and Muhlas, 2022). in this study, the Dependent Variable is the prediction result of closing stock prices produced by the LSTM model.

3. Results and discussion

3.1. Requirement gathering

In this research, there were no questionnaires and interviews, therefore the system requirements are determined through website reengineering by comparing several websites that already exist. The three websites include AiStockFinder.com, tipranks.com, and StockInvest.us. The comparison between the three websites can be seen in **Table 1**.

Table 1. Website comparison.

Component	AiStockFinder.com	tipranks.com	StockInvest.us	Score
Search Stocks	✓	✓	✓	3
Watchlist	✓	✓	✓	3
Signup	✓	✓	✓	3
Login	✓	✓	✓	3
Responsive	✓	✓	✓	3
prediction	✓	✓	✓	3
Stock Comparison	✓	✓	×	2
Stock Forum/comments	×	✓	×	1

Based on **Table 1**, it can be seen that all three websites have common features such as Search, Watchlists, Login, and Signup. In addition, the three websites also have stock prediction features. The three websites also have been designed as responsive websites to ensure optimal display on all devices. However, some features are only provided by certain websites, such as Stock Forum/Comments which is only provided by tipranks.com, and stock comparison which is only provided by AiStockFinder.com and tipranks.com. Through the comparison table, the website is going to have components of a Home Page, Search, Watchlists, Login, About, Chart, and Prediction because it can be concluded that these components and features are needed on all three websites. In addition, although not explicitly mentioned in **Table 1**, the website have an administrator role. The presence of an administrator is needed to provide more efficient content management, allowing activities such as Create, Read, Update, and Delete (CRUD) to be performed more easily. Therefore, based on the comparison table and the previous explanation, here are some functional requirements and nonfunctional requirements that be applied to this website:

1) Functional requirement

a. Search stocks (administrator, visitor)
Users or administrators can search for the desired stock by entering the stock symbol in the search bar and hitting the enter button. Through this feature, users can view information related to the stock such as company description, closing price chart visualization, and recent historical data. The search history is saved if the user has logged in.

b. View stock prediction (administrator, visitor)
Users or administrators can see the prediction results for the next 30 days for the stocks that have been provided by the website. The prediction results are displayed in the form of a chart visualization.

c. Watchlists (administrator, visitor)
Users or administrators can save the stocks or predictions they desire into the watchlist. All stock/prediction chart visualizations saved by the user are displayed on the watchlists page.

d. Signup (administrator, visitor)
Users can register if they do not have an existing website account. Users is asked to fill in their username, email, passwords, and confirm passwords on the form that has been provided.

e. Login (administrator, visitor)
Users or administrators are required to fill in identities such as username and password to access the website using the account that has been registered.

f. CRUD (administrator)
The administrator can perform activities such as Create, Read, Update, and Delete (CRUD) the pre-trained model data contained in the database.

2) Non-functional requirement

a. Responsive website.

b. The website can be accessed from any device.

3.2. Quick design

At this step, we create a model design for the system. In this research, Unified Modeling Language (UML) is used as a method of modeling the system. At first, we used Use Case Diagram as one part of UML which aims to display the main functions contained on the website and actors related to the functions. The use case diagram is shown in **Figure 4**. The use case diagram illustrates a system with two main actors: administrator and visitor. Both actors can engage in various activities, including login and signup for accessing the system, searching for stocks, viewing stock predictions, and managing watchlists. The 'Display login/signup error' activity extends the 'Login' and 'Signup' activities, occurring when invalid credentials are provided, or registration fails. The 'Signup' activity also includes the 'Verify Account' activity, while the 'View Stock Prediction' activity includes the 'Display Stock Prediction' activity. Additionally, the administrator has exclusive privileges to perform Create, Read, Update, and Delete (CRUD) operations on the pre-trained models. The diagram illustrates the various interactions and functionalities of the system.

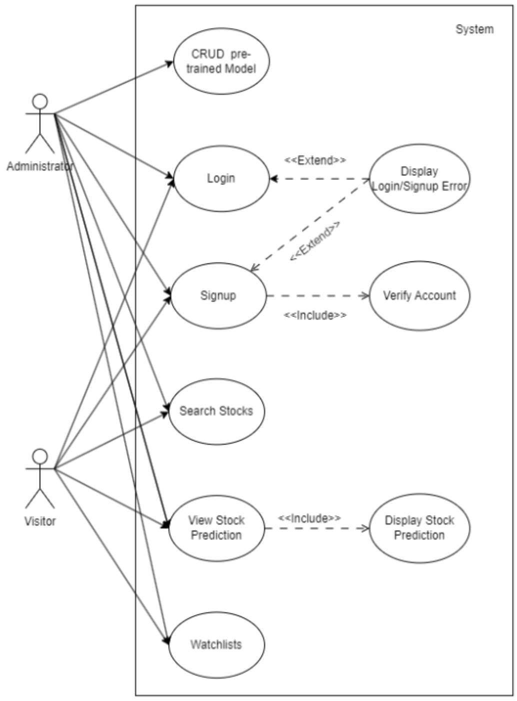

Figure 4. Use case diagram.

Figure 5 represents the activity diagram for stock price prediction system. The activity begins when the user visits the website. The website includes a navigation bar with a "Stock Predictions" menu option, which allows users to navigate and open the prediction page. Upon selecting this menu, the user is directed to a page where they can choose from various stock prediction options. Upon selecting a specific stock prediction, the system initiates a series of steps to provide the user with relevant information. First, the system downloads the required stock data, which serves as input for the pre-trained model. Subsequently, the pre-trained model performs the prediction based on the downloaded data. The system then displays the prediction results, presenting performance indicators and visualization charts to enhance the user's understanding. At this stage, the user can view the prediction results and assess the potential performance of the selected stock. Furthermore, the system offers an option for the user to save the prediction results into their watchlists, enabling them to track and monitor the stock's performance over time.

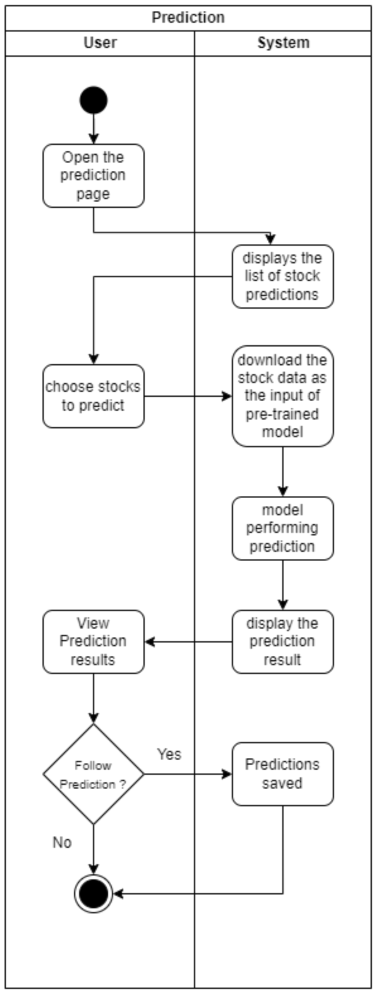

Figure 5. Activity diagram.

The database design in **Figure 6** consists of four tables to support the system's functionality. The first table, "Prediction", stores information related to pre-trained models. It includes fields such as PredictionId (primary key), PredictionName, PredictionModel (stored as binary data), Symbol, and performance indicators like RMSE, MSE, MAE, and MAPE. The second table, "Watchlists", manages user-specific watchlists and is associated with the UserTable and Prediction tables through foreign keys (UserId and PredictionId). The UserTable table contains user-related information, including UserId (primary key), UserName, Password, Email, and a field called Is_Admin to determine administrator privileges. Finally, the "History" table tracks user search history, storing details such as HistoryId (primary key), SymbolKeyword, UserId (foreign key), and a flag called Is_Saved, which indicates whether a stock has been saved to the watchlists or not. This database design facilitates the storage and retrieval of prediction models, user watchlists, user information, and search history, supporting the core functionalities of the system.

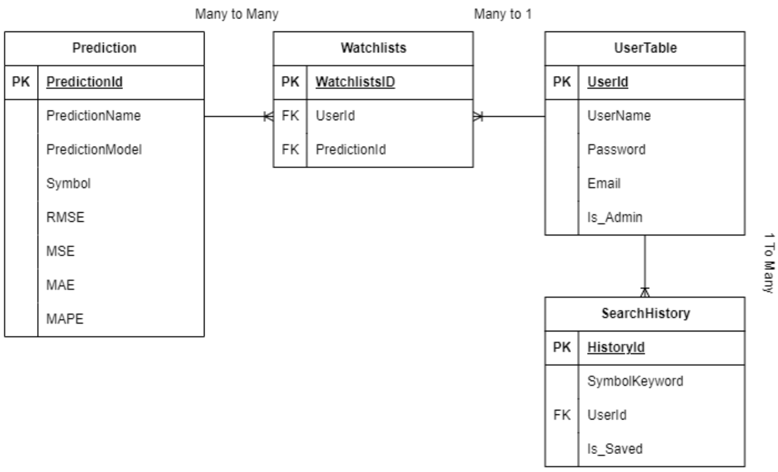

Figure 6. Database design.

3.3. Building prototype

Building Prototype is a phase of application development based on requirement gathering and quick design that have been discussed previously. In this research, the website is developed using Flask as the backend framework, Bootstrap as the frontend framework, while the LSTM deep learning algorithm is used to predict stock prices. In addition, the database used on this website is SQLite. The website development process is done by using Visual Studio Code software.

The first step before making predictions for stock closing prices using the LSTM algorithm is data preparation. In this stage, data is prepared by downloading stock data from Yahoo Finance using the yfinance library and creating descriptive statistics for the stocks to be predicted. These descriptive statistics provide information such as the minimum, maximum, mean, standard deviation, and range of stock prices. These statistics can be utilized to evaluate the LSTM model. The results of the descriptive statistics that have been obtained can be observed in **Table 2**.

Table 2. Descriptive statistics.

Symbol	Min	Max	Mean	Range	Std. deviation
BBCA.JK	4120	9300	6533.156200	5180	1230.820867
BBRI.JK	2170	5225	4008.735864	3055	609.563749
BMRI.JK	1860	10225	3627.451691	8365	727.664529
BBNI.JK	3160	9900	7268.698630	6740	1631.263765
BRIS.JK	135	3770	1161.637390	3635	761.921915

After understanding descriptive statistics, the next step is to build an LSTM model. The required libraries for building an LSTM model are imported to accomplish this. Following that, the downloaded stock data is converted into a dataframe. This dataframe consists of a unique identifier, datestamp, and closing price values. The dataframe is subsequently transformed into a PyTorch TimeSeriesDataset to fit the data into the model. Afterward, the data is split, with the last 30 data points allocated for testing purposes, while the remaining data is utilized for training the model.

The LSTM model is created using the AutoLSTM library from neuralforecast. This library simplifies the model creation process by performing hyperparameter tuning to select the best hyperparameters. It can help to avoid some problems, for example the local minima. The neuralforecast AutoLSTM library utilized Grid Search and BasicVariantGenerator to perform hyperparameter tuning. Once the model has been developed, the next step involves predicting the closing prices of stocks for the next 30 days. This prediction generates 30 data points, with each point corresponding to the closing price for a specific day within that future period.

Based on **Table 3**, which compares the number of days predicted, several metrics are displayed for different prediction periods: 30 days, 60 days, and 120 days. When considering the performance indicators, it is observed that the 30-day prediction period exhibits relatively favorable results compared to the other durations. The Mean Absolute Percentage Error (MAPE) for the 30-day prediction stands at 7.14%, indicating a relatively low level of prediction error. Additionally, the Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) values for the 30-day prediction are comparatively lower than those of the other periods. These metrics suggest that the model's accuracy in predicting stock performance within a 30-day timeframe is relatively higher. The validation loss value of 0.145 further supports the model's effectiveness in this prediction duration. Therefore, based on the table's findings, choosing a 30-day prediction period may provide more accurate and reliable predictions for future stock performance.

Table 3. Comparison of the number of days predicted.

Days	MAPE	MSE	RMSE	MAE	Val_loss
30	7.14%	435,056.20	659.58	636.76	0.145
60	9.34%	730,263.84	854.55	822.60	0.442
120	15.41%	188,9054.30	1374.42	1342.24	1.060

Furthermore, the performance of the prediction model can be evaluated by calculating RMSE (Root Mean Square Error), MSE (Mean Squared Error), MAE (Mean Absolute Error), and MAPE (Mean Absolute Percentage Error). The results of the calculated performance metrics can be seen in **Table 4**.

Table 4. Performance metrics.

No	Symbol	MAPE	MSE	RMSE	MAE
1.	**BBCA.JK**	7.14%	435,056.20	659.58	636.76
2.	**BBRI.JK**	8.06%	198,093.26	445.07	404.67
3.	**BMRI.JK**	4.10%	871,530.16	933.55	302.55
4.	**BBNI.JK**	4.75%	230,566.97	480.17	445.64
5.	**BRIS.JK**	8.38%	24,861.89	157.67	145.47

Based on the analysis of descriptive statistics in **Table 2** and MAPE accuracy scale judgment in **Table 5**, it is found that the LSTM model's performance metrics in **Table 4** have good values, where the RMSE, MSE, and MAE of each prediction are low if compared to the values in the descriptive statistics table, while the MAPE value is less than 10%, which indicates a "Highly accurate" forecast accuracy level. This indicates that the LSTM model can generate predictions with a low level of error. To seamlessly integrate the prediction model into the website, it is essential to save the model as a pre-trained model and subsequently store it in the database. This enables convenient access and utilization of the model within the infrastructure of the website.

Table 5. MAPE accuracy scale judgment (Maruddani and Trimono, 2018).

MAPE	Forecast accuracy
< 10%	Highly accurate
11% – 20%	Good forecast
21% – 50%	Reasonable forecast
> 51%	Inaccurate forecast

This prototype is built in the form of a website. The process of real time prediction in the website can be seen in **Figure 7**. It includes four modules: graphical user interface (GUI), database, real-time stock price data collection, and predicting. In this system, users select the stocks to predict from a list provided by the website through the GUI. The pre-trained model is then loaded from the database based on the user's stock selection. Subsequently, the system downloads the related stock data for the selected stocks. The predicting process begins by inputting the downloaded data into the pre-trained model, initiating the prediction phase. The pre-trained model generates prediction outputs, which are then visualized in the GUI using charts, enabling users to easily interpret and understand the predicted stock prices.

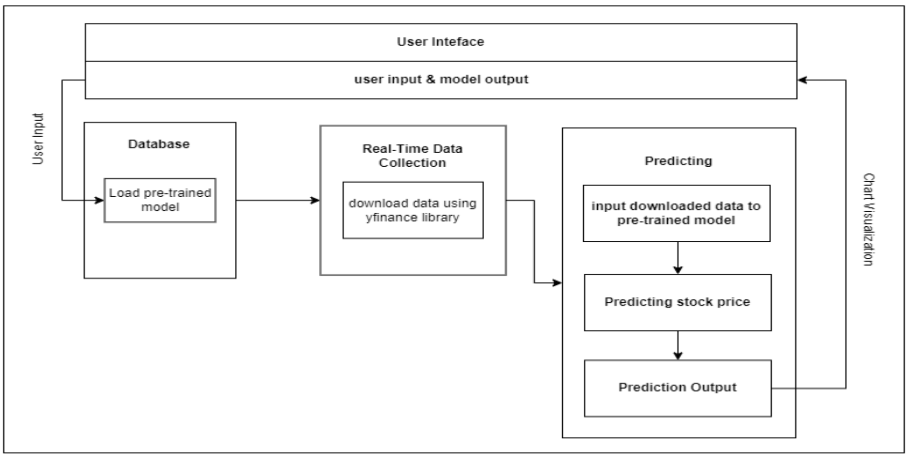

Figure 7. Real time prediction method.

The visual presentation of the developed website can be observed in **Figure 8**. It represents a page designed to showcase the predicted outcomes that are presented to the user upon selecting the prediction feature. It can be observed that the prediction page contains several pieces of information to be displayed to the user. The displayed information includes a list of stocks for which their predictions can be viewed, the name of the stock, the company's description, performance metrics, and a graph illustrating the predictions for the next 30 days. The predicted outcomes shown on the website are generated using a pre-trained model. The model is trained using the LSTM algorithm and saved as a .pth file using the PyTorch library. The model, now in the form of a .pth file, is uploaded to the database through the admin page. The model stored in the database can then be accessed by the website. The prediction of stock prices takes place when a user accesses the "predict" page and selects which stock to predict. When the user chooses the stock to be predicted, the current stock price data of that particular stock is downloaded using the yfinance library. The data retrieved consists of the closing prices of the stock, which are used as inputs for the pre-trained model accessed through the database. Through this process, the resulting output is a prediction of the closing stock price for the next 30 days. The outcome of this prediction is then displayed on the website using the chart.js library. The prediction page provides a save button located in the top right corner (under the username). When this button is pressed, the respective stock prediction is saved to the watchlist.

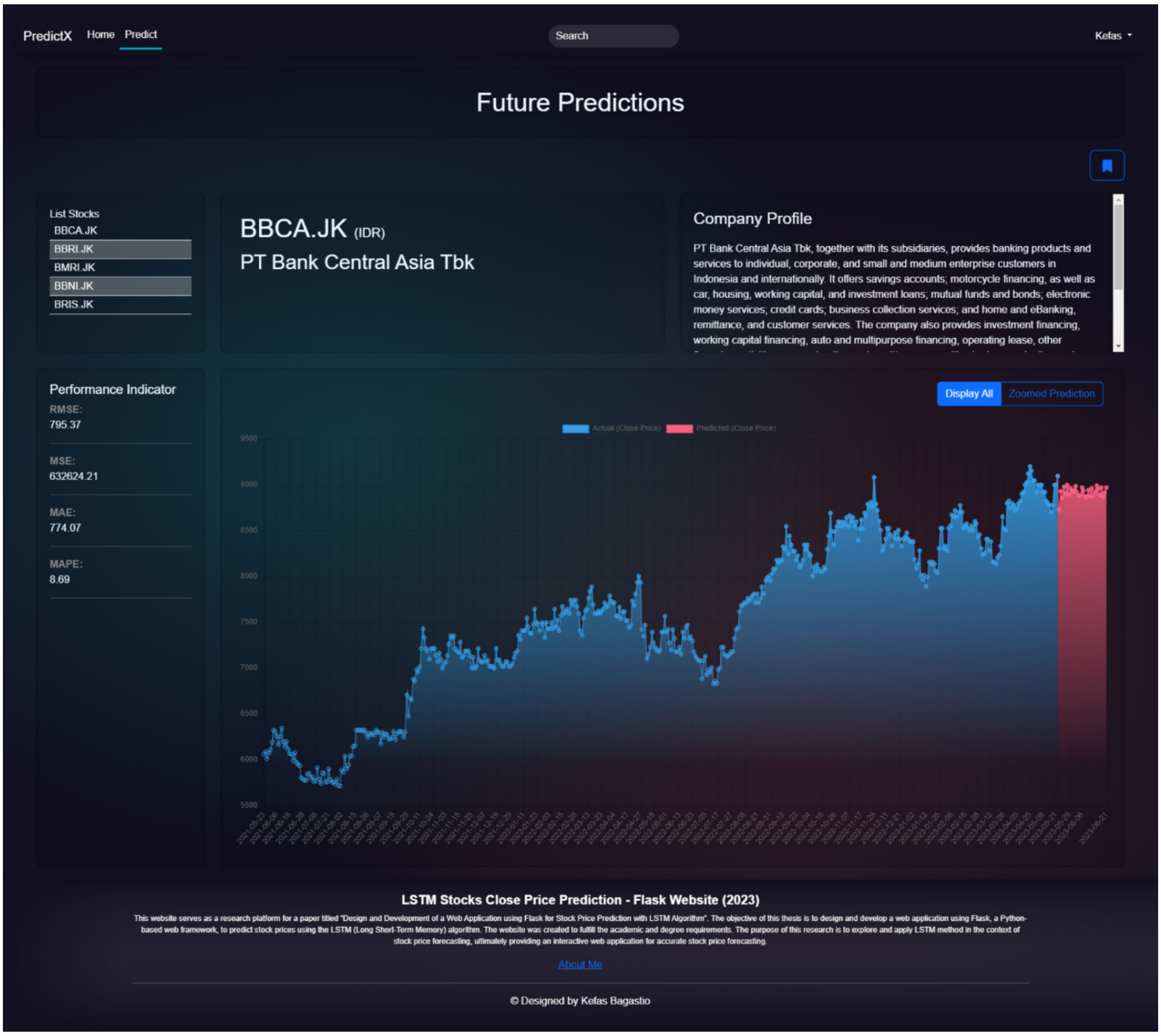

Figure 8. Prediction webpage.

3.4. Customer evaluation

This study employed the User Acceptance Testing (UAT) method to measure the suitability of the constructed website and ensure its proper functionality. Testing of various scenarios is carried out by users to evaluate its performance. The UAT test results can be seen in **Table 6**. It can be seen that all of the features are working properly, and accepted by the user.

Table 6. User acceptance testing.

Feature	User 1	User 2	Percentage
SignUp	*Pass*	*Pass*	100%
Log In	*pass*	*pass*	100%
Search stocks	*Pass*	*Pass*	100%
Predict	*Pass*	*Pass*	100%
Watchlist (Prediction)	*Pass*	*Pass*	100%
Watchlists (stocks)	*Pass*	*Pass*	100%

Other than conducting the UAT, the Customer satisfaction score (CSAT) is also carried out. This method involved gathering feedback from respondents who tried the website and shared their opinions through a questionnaire. The Customer satisfaction score provides insights into the satisfaction levels of customers regarding various features of the website. **Table 7** presents the satisfaction ratings of 25 respondents for various features of the website. The features include login, signup, search stock, predict, watchlist (predict), and watchlist (stocks). The ratings were categorized into five levels: Very Dissatisfied, Somewhat Dissatisfied, Neutral, Somewhat Satisfied, and Very Satisfied. The table also includes the percentage of respondents for each satisfaction level.

Table 7. Customer satisfaction ratings for website features.

Feature	Very dissatisfied	Somewhat dissatisfied	Neutral	Somewhat satisfied	Very satisfied	Percentage	Classification
	(1)	(2)	(3)	(4)	(5)		
Login	0	0	2	3	20	94.4%	**Very satisfied**
Signup	0	0	2	3	20	94.4%	**Very satisfied**
Search stock	0	0	2	1	22	96.0%	**Very satisfied**
Predict	0	0	0	1	24	99.2%	**Very satisfied**
Watchlist (predict)	0	0	1	3	21	96.0%	**Very satisfied**
Watchlist (stocks)	0	0	1	3	21	96.0%	**Very satisfied**

Table 7 reveals a high level of user satisfaction, with all users rating the features as "Very Satisfied", ranging from 94% to 99% satisfaction. The login and signup processes receive overwhelmingly positive feedback, with 80% of customers reporting being "Very Satisfied". This indicates that users find the system's registration and authentication procedures efficient and user-friendly. Similarly, the search stock feature has a high satisfaction rating of 96%, indicating that users find it effective and useful. The predicted feature receives a satisfaction rating of 99.2%, suggesting that users are extremely pleased with its performance. The watchlist feature, both for predictions and stocks, garners a 96% satisfaction rating, indicating that users are generally satisfied with its functionality. Overall, the table demonstrates a high level of user satisfaction across all features of the system, with the predicted feature standing out as the most favored.

3.5. Refining prototype

Users have suggested adding additional information such as the minimum, maximum, average, and standard deviation of the predicted data to provide a more comprehensive understanding of the predicted values. The change can be observed in **Figure 9**, where the added information is displayed alongside the predictions, offering a more comprehensive insight into the predicted data.

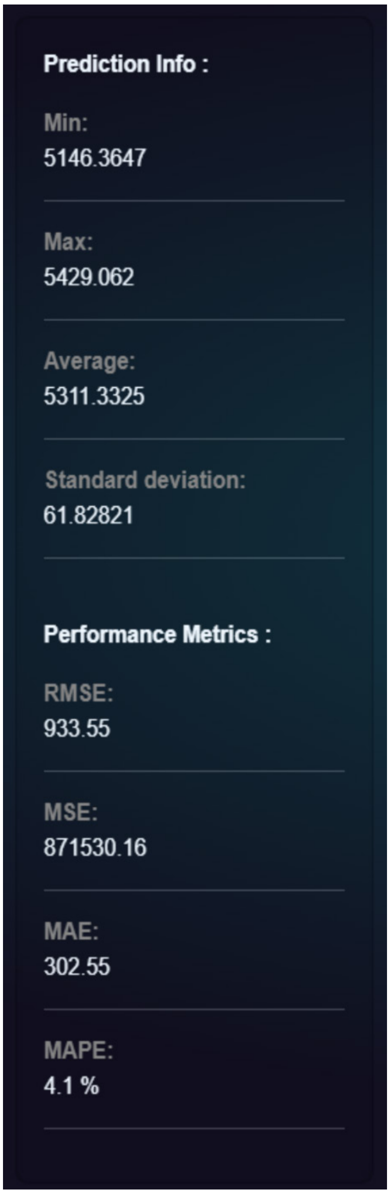

Figure 9. Refined information about the prediction result.

The prediction results displayed on the website use a pre-trained model. The model that has been trained using the LSTM algorithm, then be saved into a file with the .pth format using the pytorch library. The model that has become .pth file then be uploaded into the database through the admin page. The model stored in the database then can be accessed by the website. Stock price prediction is performed when the user accesses the “predict” page and selects which stock to predict. When the user selects the stock to be predicted, the current stock price data of the stock is downloaded using the yfinance library, the data taken is the closing price of the stock to be used as input for the pre-trained model that has been accessed through the database. Through this stage, the output produced is a prediction of the closing price of the stock for the next 30 days. The results of the prediction is displayed on the website using the chart.js library.

4. Conclusion

This research project successfully developed a stock price prediction website using the Flask framework, which relies on the Long Short-Term Memory (LSTM) deep learning algorithm to predict closing stock prices. The website incorporated various technologies such as neuralforecast AutoLSTM for LSTM model development, Flask as the backend framework, Bootstrap for the frontend framework, and an SQLite database for data storage. The website provided an interactive and responsive user experience, generating real-time forecasts based on pre-trained LSTM models stored in the database. The predictions, displayed using the chart.js library, covered a prediction for the next 30 days.

Based on the User Acceptance Testing (UAT) results, the website performed exceptionally well and met the expected standards. The customers’ satisfaction levels were measured through the Customer Satisfaction (CSAT) table, which indicated high ratings for all features. The login, signup, search stock, predict, and watchlist (both for predictions and stocks) features received overwhelmingly positive feedback, with satisfaction percentages ranging from 94.4% to 99.2%. Furthermore, the UAT results showed that all tested features, including signup, login, search stocks, predict, and watchlists (both for predictions and stocks), passed without any issues, achieving a 100% pass rate. These results highlight the website’s ability to deliver a user-friendly experience, meeting user expectations and ensuring a high level of satisfaction.

The developed model showed good performance in predicting stock prices, with low errors based on metrics such as MSE, RMSE, and MAE across all stock symbols. Additionally, the mean absolute percentage error (MAPE) ranged from 4.10% to 8.38%, indicating the model’s ability to achieve “highly accurate” predictions. The research provided valuable insights into the implementation of the LSTM algorithm within a stock prediction website using Flask as a backend framework. In future research, alternative algorithms such as the Gated Recurrent Unit (GRU) could be explored for stock price prediction, as well as the performance of multivariate predictions to provide a more comprehensive analysis. These further developments would enhance the capabilities of the site and provide a wider range of insights for users.

Author contributions

Conceptualization, KB and RSO; methodology, KB, RSO and AR; software, KB; validation, KB, RSO and AR; formal analysis, KB, RSO and AR; investigation, KB, RSO and AR; resources, KB; data curation, KB; writing—original draft preparation, KB and RSO; writing—review and editing, AR; visualization, KB and AR; supervision, RSO and AR; project administration, KB and RSO; funding acquisition, RSO. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare no conflict of interest.

Acknowledgements

This study is conducted effectively due to the assistance provided by Universitas Multimedia Nusantara, colleagues, relatives, and the supervising professor who provided valuable feedback and guidance throughout the research. The support and assistance received during the writing of this article is gratefully acknowledged.

References

Anggoro DA, Aziz NC (2021). Implementation of K-Nearest Neighbors algorithm for predicting heart disease using Python Flask. *Iraqi Journal of Science* 62(9): 3196–3219. doi: 10.24996/IJS.2021.62.9.33

Bathla G (2020). Stock price prediction using LSTM and SVR. In: Proceeding of the 2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC); IEEE; pp. 211–214.

Dong S, Wang P, Abbas K (2021). A survey on deep learning and its applications. *Computer Science Review* 40: 100379. doi: 10.1016/J.COSREV.2021.100379

Guha SK, Samanta N, Majumdar A, et al. (2019). Evolution of corporate governance in India and its impact on the growth of the financial market: An empirical analysis (1995–2014). *Corporate Governance (Bingley)* 19(5): 945–984. doi: 10.1108/CG-07-2018-0255

Hamdani NA, Yulianto E, Maulani GAF (2021). Designing loss event database using evolutionary prototyping model to perform bank operational risk management identification process. *IOP Conference Series: Materials Science and Engineering* 1098(4): 042008. doi: 10.1088/1757-899x/1098/4/042008

Hastomo W, Karno ASB, Kalbuana N, et al. (2021). Deep learning pptimization for stock predictions during the Covid-19 pandemic (Indonesian). *JEPIN* (*Jurnal Edukasi dan Penelitian Informatika)* 7(2): 133–140. doi: 10.26418/JP.V7I2.47411

Ilham RN, Irawati H, Nurhasanah N, et al. (2022). Relationship of working capital management and leverage on firm value: An evidence from the Indonesia stock exchange. *Journal of Madani Society* 1(2): 64–71. doi: 10.56225/JMSC.V1I2.129

Janiesch C, Zschech P, Heinrich K (2021). Machine learning and deep learning. *Electronic Markets* 31(3): 685–695. doi: 10.1007/s12525-021-00475-2

Kumar MP, Kumara NM (2020). Market capitalization: Pre and post COVID-19 analysis. *Materials Today: Proceedings* 37(2): 2553–2557. doi: 10.1016/j.matpr.2020.08.493

Kyin MS, Oo ZL, Cho KM (2020). An overview studying of deep learning. *International Journal of Scientific Research in Science Engineering and Technology* 7(2): 394–398. doi: 10.32628/IJSRSET207279

Liang Z, Liang Z, Zheng Y, et al. (2021). Data analysis and visualization platform design for batteries using flask-based Python Web Service. *World Electric Vehicle Journal* 12(4): 187. doi: 10.3390/wevj12040187

Maruddani DAI, Trimono (2018). Modeling stock prices in a portfolio using multidimensional geometric brownian motion. *Journal of Physics: Conference Series*. doi: 10.1088/1742-6596/1025/1/012122

Meshram S, Narsale S, Sangamnere S, et al. (2022). Stock prediction webapp using Python. *International*

Journal of Advanced Research in Science, Communication and Technology (IJARSCT)* 2(7). doi: 10.48175/IJARSCT-4359

Mufid MR, Basofi A, Al Rasyid MUH, et al. (2019). Design an mvc model using python for Flask framework development. In: Proceedings of the 2019 International Electronics Symposium (IES); IEEE. doi: 10.1109/ELECSYM.2019.8901656

Prasad A, Seetharaman A (2021). Importance of machine learning in making investment decision in stock market. *Vikalpa* 46(4): 209–222. doi: 10.1177/02560909211059992

Prasetya BD, Pamungkas FS, Kharisudin I (2019). Modeling and forecasting stock data with time series analysis using Python (Indonesian). *PRISMA, Prosiding Seminar Nasional Matematika* 3: 714–718.

Rachma N, Muhlas I (2022). Comparison of waterfall and prototyping models in research and development (R&D) methods for android-based learning application design. *Jurnal Inovatif: Inovasi Teknologi Informasi Dan Informatika* 5(1): 36–39. doi: 10.32832/inova-tif.v5i1.7927

Rahmawati A, Garad A (2023). Managerial ownership, leverage, dividend policy, free cash flow, firm value: Evidence in Indonesia stock exchange. *European Journal of Studies in Management and Business* 25: 32–44. doi: 10.32038/mbrq.2023.25.03

Ramyakim RM, Widyasari A (2022). Age demographics of shareholders investors per sector (up to March 2022) (Indonesian). Available online: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwiFsPnQ-7r_AhUTwTgGHbCPC8MQFnoECCAQAQ&url=https%3A%2F%2Fwww.ksei.co.id%2Ffiles%2Fuploads%2Fpress_releases%2Fpress_file%2Fid-id%2F205_berita_pers_saham_industri_keuangan_menjadi_incaran_investor_gen_z_20220420142705.pdf&usg=AOvVaw1wUq-2cdRxSSCmCIZwp6U8 (accessed on 11 June 2023).

Siami-Namini S, Tavakoli N, Namin AS (2018) A comparison of ARIMA and LSTM in Forecasting Time Series. In: Proceeding of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA); IEEE; pp. 1394–1401 2018.

Smagulova K, James AP (2019). A survey on LSTM memristive neural network architectures and applications. *The European Physical Journal Special Topics* 228(10): 2313–2324. doi: 10.1140/EPJST/E2019-900046-X.

Statistik Pasar Modal Indonesia (2023). Available onlnine: www.ksei.co.id (accessed on 23 August 2023).

Untoro AB (2021). Stock price prediction using artificial neural networks (Indonesian). *Jurnal Teknologi Informatika dan Komputer* 6(2): 103–111. doi: 10.37012/jtik.v6i2.212.

Van Houdt G, Mosquera C, Nápoles G (2020). A review on the Long Short-Term memory model. *Artificial Intelligence Review* 53(8): 5929–5955. doi: 10.1007/s10462-020-09838-1

Xia K, Huang J, Wang H (2020). LSTM-CNN architecture for human activity recognition. *IEEE Access* 8: 56855–56866. doi: 10.1109/ACCESS.2020.2982225

Yahoo finance-stock market live, quotes, business & finance news. Available online: https://finance.yahoo.com/ (accessed on 23 August 2023).