Journal of Information and Organizational Sciences
Volume 48, Number 2 (2024)
Journal homepage: jios.foi.hr

JIOS

DOI: 10.31341/jios.48.2.2
Open Access

UDC 004.032.26:336.76(532)-047.72
Original Scientific Paper

Long Short-Term Memory and Discrete Wavelet Transform based Univariate Stock Market Prediction Model

Mutasem Jarrah[1*]

[1]Information Technology Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah – Saudi Arabia
[*]Correspondence: mmjarrah@kau.edu.sa

PAPER INFO

Paper history:
Received 20 August 2023
Accepted 26 January 2024
Published 18 December 2024

Citation:
Jarrah, M. (2024). Long Short-Term Memory and Discrete Wavelet Transform based Univariate Stock Market Prediction Model. In Journal of Information and Organizational Sciences, vol. 48, no. 2, pp. 263-277

ABSTRACT

Analyzing financial situations in the current scenario is difficult, as it requires understanding the quality and value of investments. This study predicted the movement of stock prices in the Saudi Arabian stock market (Tadawul) over a one-week period using a proposed integrated model of Long Short-Term Memory (LSTM), which combines LSTM, Discrete Wavelet Transform (DWT), and Autoregressive Integrated Moving Average (ARIMA). Historical closing prices of a group of four companies listed on Tadawul were used as input for the proposed LSTM model, which consists of memory units capable of storing long time periods. Once the LSTM model predicted the closing values of stocks in Tadawul, they were further analyzed using the ARIMA model. The prediction accuracy of the proposed LSTM model and the traditional ARIMA model were 97.54% and 96.29% respectively. Therefore, the proposed integrated model of LSTM is considered a useful tool for predicting stock market values. The results emphasize the significance of Deep Learning (DL) and leveraging multiple information sources in predicting stock prices.

Keywords: Long Short-Term Memory, deep learning, prediction, Univariate, Discrete Wavelet Transform, stock market, Time series

1. Introduction

Forecasting, the process of predicting the future based on past and present data, has long been of interest in stock price prediction. However, due to the complex and constantly changing nature of the financial environment, accurately making these predictions is challenging[1]. Predicting stock market trends and prices is crucial in the investment and financial sectors, as it helps traders' profit from their trades. Various techniques, such as statistical and technical analysis, can be employed to forecast market prices [2].

The movement of stock prices is difficult to predict due to various unpredictable factors and noises. Factors like changes in sentiment, national economy, product value, political factors, and weather can influence stock prices on a daily basis [3]. Numerous researchers have studied stock price movements to understand the key factors that significantly impact stock prices [4]. Hence, this study aims to develop a hybrid model called the discrete long short-term memory (LSTM) model by combining a conventional LSTM artificial neural network (ANN) and a conventional discrete wavelet transform (DWT) ANN. The goal is to use this model to predict the prices of specific stocks on the Saudi Stock Exchange (TADAWUL) for a next one-week period.

Consumer sentiment plays a crucial role in influencing stock market price movements. Future research should explore the impact of financial events and the attitudes of stock buyers towards goods and services[5].This study compares the performance of the proposed LSTM model with conventional methods like the autoregressive integrated moving average (ARIMA) model. Various modeling techniques were evaluated, and the best model was selected after considering different model configurations.

This introduction examines the application of two prominent time series forecasting techniques: Long Short-Term Memory (LSTM) networks and Autoregressive Integrated Moving Average (ARIMA) models in predicting stock market behavior. To enhance prediction accuracy, we employed the DWT process, which uses unsupervised techniques to eliminate data noise. The data underwent DWT processing before being inputted into LSTM and subsequently ARIMA. LSTM, a recurrent neural network, excels at capturing long-term dependencies in sequential data, making it suitable for modeling the dynamic and non-linear nature of stock prices. On the other hand, ARIMA, a classical statistical model, leverages autoregression, differencing, and moving average components to analyze historical stock price trends. The integration of LSTM and ARIMA provides a comprehensive approach to forecasting stock market behavior, recognizing the importance of adaptability in response to ever-changing market dynamics.

This research is organized according to the following sections: After the introduction comes the second section, which is the covers previous studies on stock price prediction, while the third section introduces and explains the proposed methodology and its characteristics. The fourth section presents and discusses the experiments of this study, and the fifth section is the discussion and results, and then the last section, which is the conclusions and future work.

2. Literature review

It is essential to review extant studies prior to conducting research. In this study, Dou Wei (2019) compared and analyzed different methods for predicting neural networks. Ultimately, the study selected the LSTM neural network, which was optimized by the MBGD algorithm, to predict stock prices. The focus of the study was to assess the feasibility and applicability of this method. The study concluded that historical information is crucial for investors when making investment decisions. While previous studies mainly relied on opening and closing prices, this study found that absolute highs and lows can provide additional insights into future price behavior. To further investigate, the study selected three representative stock indicators in the Chinese stock market. The main data collected included the opening price, closing price, low price, high price, date, and daily trading volume. The results indicate that the LSTM neural network model, with the inclusion of the attention layer, can predict stock prices despite some limitations, such as time delay in forecasting. The model utilizes time sequence analysis of historical information and the selective memory function of the LSTM network to uncover internal rules and ultimately predict stock price trends[6].

Diqi, & el. (2022) This study demonstrates the effectiveness of using a generative adversarial network (GAN) for accurately predicting stock prices. We discuss the process of collecting and preprocessing the dataset, extracting features, evaluating the model, and implementing the GAN method for stock price forecasting. The GAN model consists of a generator and discriminator trained using adversarial learning techniques. Our model includes date, opening price, highest price, lowest price, closing price, and trading volume as training features. The experimental results show high accuracy and low error rates, indicating significant potential for precise and dynamic stock price predictions. Our proposed model achieves satisfactory results with an R2 score of 0.811166 for real expectations and 0.674971 for artificial expectations. The MAE function generates real expectations of 0.020665 and artificial expectations of 0.042406. The MRLE values are 0.001087 for real expectations and 0.002479 for artificial expectations[7].

The researcher Muhammad Ali (2022) proposed a mixed method that combines a new version of Empirical Mode Decomposition (EMD) signal analysis with Long Short-Term Memory (LSTM) network, which is a well-known technique for deep learning. We evaluate the accuracy of our proposed method using the KSE-100 index for the Pakistan Stock Exchange. Instead of the traditional Triple EMD technique, we use a new version called Akima-EMD to analyze the oscillatory data of stocks into multiple components known as Intrinsic Mode Functions (IMFs) and one residue. We then use the closely related sub-components to build the LSTM neural network. We compare the performance of our mixed model with the LSTM model, as well as other models such as Support Vector Machine (SVM), Random Forest, and Decision Tree. We use three statistical measures, namely Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE), to compare these techniques. Our experimental results show that the proposed Akima-EMD-LSTM mixed model outperforms all other models in this study, making it an effective model for predicting non-stationary and nonlinear financial time series data [8].

According to Patil et al. (2023), this research presents the Deep Recurrent Rider LSTM strategy as an effective method for accurately detecting stock market values. It combines two classifiers: Rider Deep LSTM and Deep RNN. Rider Deep LSTM combines the Rider concept with Deep LSTM, while Deep RNN is trained using the Shuffled Crow Search Optimization (SCSO) technique. SCSO is a combination of the Shuffled

Shepherd Optimization (SSO) algorithm and the Crow Search Algorithm (CSA). The expected output is determined based on error conditions. The proposed Deep Recurrent Rider LSTM achieves MSE and RMSE values of 0.018 and 0.132 respectively, indicating higher performance and accuracy. Overall, the proposed classification model enhances the accuracy of stock market prediction[9].

According to the study conducted by Bhupinder et al. (2023), this study aims to predict and estimate real-time stock assets in financial markets without relying on external brokers. We focus on broadcasting-based trading and utilize various performance factors and data metrics. Accurate sample data from the Y-finance sector is collected through API-based data chains. We employ reliable machine learning algorithms to classify and predict complexities. However, stock volatility and uncertainty can introduce noise and weaken decision-making. Previous studies used fewer performance metrics. In this study, we combine Dickey-Fuller test scenarios with volatility prediction and a short-term memory algorithm. This algorithm is implemented in a recurrent neural network to forecast closing prices of large companies in the stock market. To analyze forecast accuracy, we integrate LSTM methods with ARIMA and evaluate metrics such as root mean square error, mean square error, mean absolute percentage error, mean deviation, and mean absolute error. We design experimental scenarios to minimize hardware resource usage and conduct a test case simulation[10].

Using ARIMA and XGBoost sought to forecast Saudi Telecom Company stock price trends based on its historic prices by Almaafi et al. (2023), this study evaluates and compares the effectiveness of ARIMA and XGBoost models in predicting the weekly closing prices of Saudi Telecom Company stocks. Our results show that XGBoost outperforms ARIMA in all evaluation metrics, highlighting the effectiveness of machine learning techniques in stock price prediction. The study also sheds light on the limitations of traditional statistical models in forecasting stock price volatility and emphasizes the potential of machine learning techniques in discovering hidden patterns and trends in the data. Our research provides a comprehensive overview of the suitability of different forecasting models for predicting stock prices and emphasizes the need for further exploration of machine learning techniques in finance[11].

Meanwhile, another study used LSTM networks as a cutting-edge method of learning the arrangements. Although these networks were mostly utilized with financed-based time series data, they are inherently relevant. For predicting the directional trend of the unseen data of the Standard and Poor's 500 (S&P 500) index's equities, this present study used LSTM-based networks. Additionally, factors that affect profitability are addressed, revealing details about how complex artificial neural networks function. Stock trading shows high volatility and exhibits a short-term reversal trend [12].

Jarrah and Derbali (2023) introduced a novel approach to predicting stock market indices in the Kingdom of Saudi Arabia (KSA). The study utilized opening, lowest, highest, and closing prices as variables to assist investors in making informed decisions. To eliminate noise from the input data obtained from the Saudi Stock Exchange (Tadawul), exponential smoothing (ES) was employed. The time series forecasting task was addressed using a sliding-window method with five steps. The prediction of stock market prices was carried out using a multivariate long short-term memory (LSTM) deep-learning (DL) algorithm. The proposed multivariate LSTM-DL model demonstrated its effectiveness in forecasting stock market prices, achieving prediction rates of 97.49% and 92.19% for the univariate model. These results highlight the accuracy of DL and the utilization of multiple information sources in predicting stock market trends [13]. Table 1 below summarizes the Literature review. The main contributions of this article are as follows:

- Applies DWT to Noise Reduction, using it to remove noise from the dataset and extract relevant features for predicting stock prices. This improves the quality of input for LSTM and ARIMA models.
- The LSTM and ARIMA models demonstrate adaptability by learning from historical data and adjusting to changing market conditions. This makes them suitable for predicting stock prices influenced by a mix of short-term and long-term factors.
- The research utilizes the lag feature also known as the delayed feature, for data analysis and prediction, primarily in time series analysis.

Authors	Year	Title	Algorithm	Dataset	Accuracy
Dou Wei[6]	2019	Prediction of Stock Price Based on LSTM Neural Network.	LSTM, Mini-Batch Gradient Descent (MBGD)	Chinese stock market	Classification
Diqi, & el [7].	2022	StockGAN: robust stock price prediction using GAN algorithm	Generative Adversarial Network (GAN)	Yahoo finance	R2 = 0.811166. MAE = 0.020665 MRLE = 0.001087

Muhammad Ali & el.[8]	2022	Prediction of complex stock market data using an improved hybrid emd-lstm model	LSTM & Empirical Mode Decomposition (EMD)	Pakistan Stock Exchange	RMSE = 317.504 MAE = 246.675 MAPE = 0.711
Patil et al.[9]	2023	Wrapper-Based Feature Selection and Optimization-Enabled Hybrid Deep Learning Framework for Stock Market Prediction	RNN, LSTM, Shuffled Crow Search Optimization (SCSO) technique	Yahoo finance	MSE = 0.018 RMSE = 0.132
Bhupinder et al.[10]	2023	Auto-Regressive Integrated Moving Average Threshold Influence Techniques for Stock Data Analysis	LSTM, ARIMA	Yahoo finance	LSTM RMSE = 2.50113 ARIMA RMSE = 0.0238892
Almaafi et al. [11]	Almaafi et al. (2023	Stock price prediction using ARIMA versus XGBoost models: the case of the largest telecommunication company in the Middle East	ARIMA, XGBoost	Saudi Telecom Company	Classification
Fischer & Krauss [12]	2018	Deep learning with long short-term memory networks for financial market predictions	LSTM, random forest	S&P 500	Classification
Jarrah & Derbali [13]	2023	Predicting Saudi Stock Market Index by Using Multivariate Time Series Based on Deep Learning	LSTM, Exponential smoothing (ES)	stock market of the Kingdom of Saudi Arabia (Telecommunication)	RMSE multivariate = 0.404 RMSE univariate = 0.552

Table 1. Literature review

3. Proposed Methodology

This research introduces the LSTM, ARIMA and DWT because the field of time series analysis has witnessed significant progress in the field of deep learning techniques. Among these techniques, Long Short-Term Memory (LSTM) networks have emerged as a powerful tool for modeling and predicting sequential data, especially univariate sequential data. Time series, which represent sequentially ordered data points, are abundant in various fields, including finance, economics, medicine, and environmental sciences. Researchers and specialists understand time series contain noise, and to obtain accurate forecasts, it is necessary to remove the noise. Therefore, the Discrete Wavelet Transform (DWT) is integrated as a first step to denoise the data before entering the second phase, which is the LSTM. The attached Figure. 1 below illustrates the stages of the proposed model. The effectiveness of an LSTM model relies on the quality of the data, the architecture and hyperparameter choices, and domain expertise. To obtain accurate and meaningful results, it is essential to experiment, iterate, and possess a profound understanding of the underlying principles.

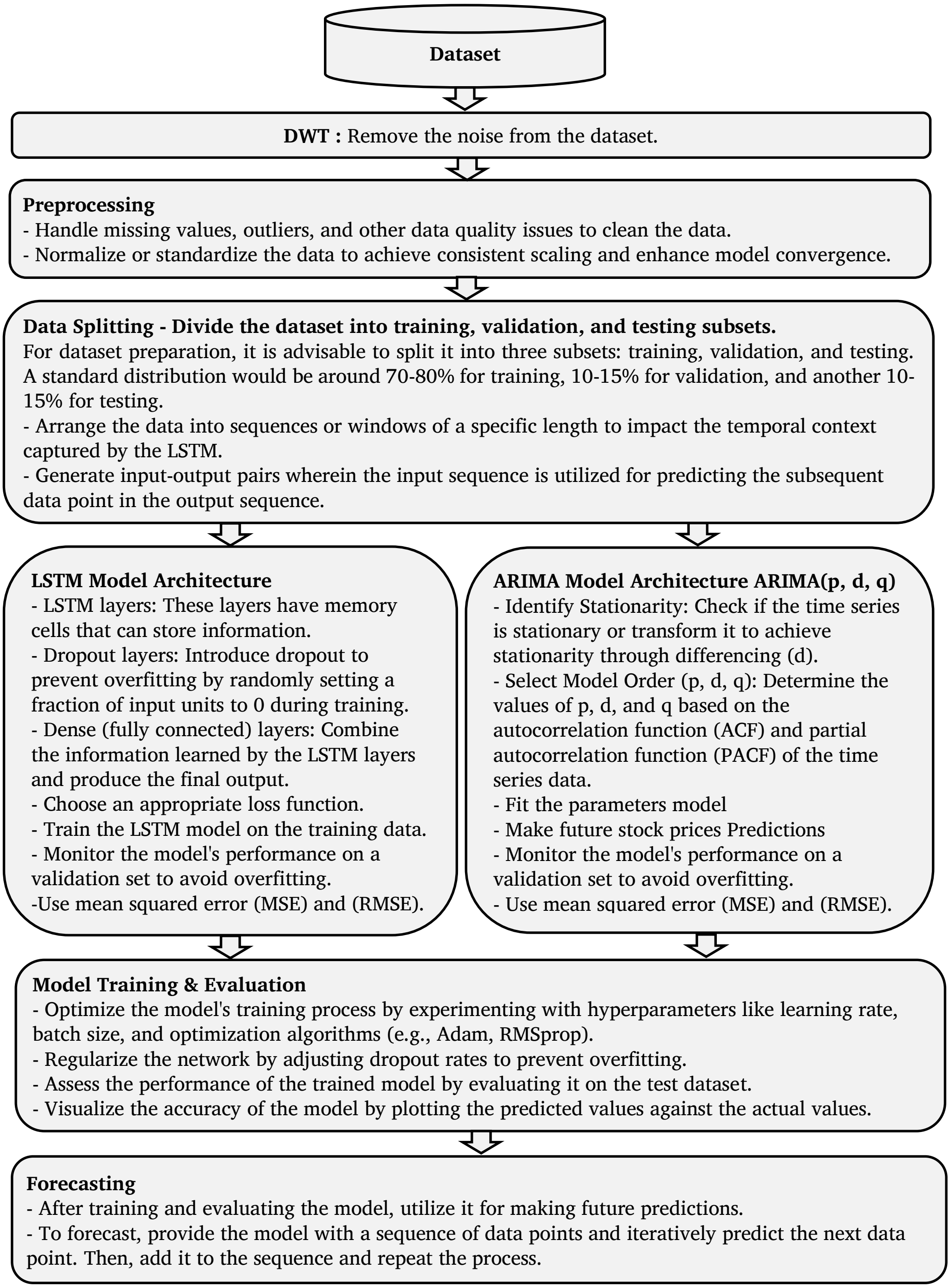

Figure 1. Proposed Methodology

3.1. Discrete Wavelet Transform (DWT)

The Discrete Wavelet Transform (DWT) is an analytical technique used for signal and multi-dimensional data analysis. DWT works by dividing the signal into wavelet components across a specified set of basic waves, allowing analysts to understand signal details at different levels of detail and frequencies. Regarding the use of DWT in eliminating input data noise, it can definitely play a role in improving signal quality and getting rid of noise. The basic idea here is to utilize DWT's unique ability to represent the signal at multiple levels of detail, enabling the separation of important and useful signal components from those affected by noise. The general steps for using DWT to eliminate input data noise include [14]:

- DWT Transformation: Applying DWT to the input signal to transform it into a set of wavelet components at different levels of details and frequencies.
- Noise Estimation: Estimating the type and level of noise in the wavelet components at different levels.
- Filtering Noise-Affected Components: By separating the noise-affected components and either ignoring or appropriately modifying them.
- Inverse DWT Transformation: After filtering the noise-affected components, an inverse transformation should be performed to return to the time domain.

The use of DWT in eliminating data noise relies on a good understanding of the signal and the present noise, as well as identifying the appropriate levels of detail that contain vital information and those that carry noise. If these steps are executed correctly, DWT an contribute to improving data and signal quality by eliminating annoying noise[15]. The Figure 2 shows the benefits of using the DWT, where the noise in the input data is eliminated before entering it into the proposed model.

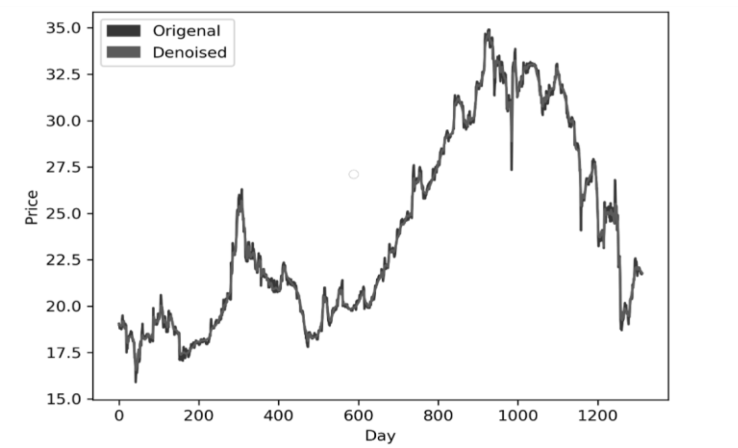

Figure 2. Output dataset after removed the noise by DWT

3.2. Long Short-Term Memory (LSTM)

An LSTM, known as Long Short-Term Memory, is a recurrent neural network (RNN) architecture that effectively captures long-range dependencies in sequences. It tackles common problems faced by traditional RNNs, such as the vanishing gradient problem. This is achieved by incorporating specialized gates that regulate the information flow within the network. The three main gates in LSTM networks are the forget gate, the input gate, and the output gate. Each gate has a distinct role in managing the flow of information. Figure 3 below explain that gates.

The forget gate helps the LSTM decide what information to discard from the cell state (the memory of the network) from the previous time step. It takes the previous hidden state (h_[t-]) and the current input (x_t) as inputs and produces a value between 0 and 1 for each element in the cell state. This value represents how much of the information from the previous cell state should be forgotten and how much should be retained. [Ref] Mathematically, it is calculated as:

$$f_t = \sigma\left(W_f[h_t - 1, X_t] + b_f\right) \quad (1)$$

Here, W_f and b_f are the weights and bias associated with the forget gate, and σ is the sigmoid activation function.

The input gate helps the LSTM decide what new information to update in the cell state from the current input. It also consists of two parts: the first part calculates a candidate cell state update (C̃_t), and the second part determines which portions of this candidate update to add to the cell state. The input gate takes the previous hidden state (h_[t-1]) and the current input (x_t) as inputs. Mathematically, the calculations are as follows:

$$i_t = \sigma(W_i[h_t - 1, X_t]) + b_i \quad (2)$$

$$C_t = \tanh(W_c\,[h_t - 1, X_t] + b_c) \quad (3)$$

The output of the input gate is i_t, represents how much of the candidate update to add to the cell state. where, *tanh* is the function that activates the state [13].

Cell State Update: The actual update to the cell state (C_t) is a combination of forgetting some previous information and adding new information. This is done by multiplying the previous cell state by the forget gate output and adding the candidate update scaled by the input gate output:

$$C\_t = f_t * C\_(t\text{-}1) + i_t * C_t \quad (4)$$

Output Gate: The output gate determines the new hidden state for the current time step. It takes the previous hidden state (h_(t-1)), the current input (x_t), and the updated cell state (C_t) as inputs. It produces a new hidden state for this time step (h_t). Mathematically, the calculations are as follows:

$$O_t = \sigma\,(W_o\,[h_t - 1, X_t] + b_o) \quad (5)$$

$$h_t = O_t * \tanh(C_t) \quad (6)$$

The output gate regulates the extent to which the cell state is revealed as the hidden state. In essence, LSTM gates govern information flow in the network by deciding what to forget, what new information to incorporate, and how to update the hidden state [16]. This architecture empowers LSTMs to effectively manage long-range dependencies and capture contextual information in sequential data.[17]. Figure 2 below illustrates the gates.

The term "sliding window size" in LSTM networks refers to how the LSTM processes sequential data. Sequential data includes time series and natural language sentences, where the order of elements matters. The sliding window size determines the number of time steps the LSTM considers at a time. It acts like a moving window that slides through the sequence. For example, if the sequence has timestamps from t=1 to t=10 and a sliding window size of 3, the LSTM will initially consider inputs from t=1, t=2, and t=3. Then it would slide the window to consider t=2, t=3, and t=4, and so on.

When processing data using a sliding window, the LSTM considers the input data within that window to produce an output, such as a prediction or classification. This approach allows the LSTM to capture patterns and dependencies spanning a specific number of time steps.

The choice of sliding window size has trade-offs. A larger window size can capture longer-term dependencies but increases model complexity and computational intensity. A smaller window size captures short-term dependencies well but may miss longer-term patterns. To ensure important transitions between adjacent windows are not missed, overlapping sliding windows are commonly used. For instance, with a window size of 3 and overlapping by 1, the LSTM would process t=1, t=2, t=3, then move the window to t=2, t=3, t=4, and so on. In practice, the choice of sliding window size depends on the data characteristics and the specific problem being solved. It is often a hyperparameter that needs to be tuned along with other LSTM architecture parameters to achieve optimal performance for the task at hand.

The LSTM (Long Short-Term Memory) neural network algorithm has been used in stock price prediction for several reasons. Here are some reasons that make LSTM suitable for this task:

- Ability to handle time series:

Stock prices are time series, where today's price depends on the previous days' prices. LSTMs are designed to handle sequential data and interact with events over time, making them suitable for analyzing and predicting stock prices.

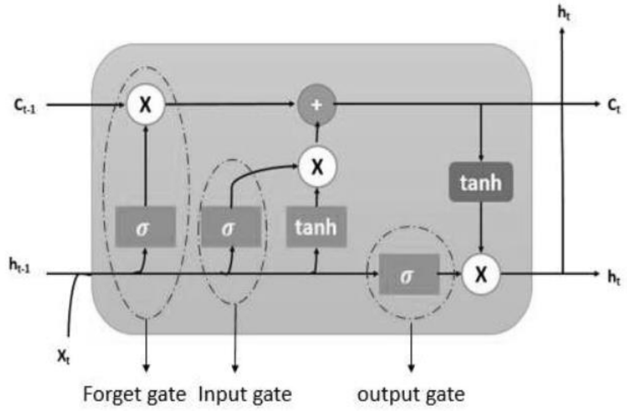

Figure 3. LSTM Gates [18]

- Providing temporal representation of information:

LSTMs retain internal memory that allows them to store and retrieve information over time. This helps in dealing with events that may be significant in the future and their impact on stock prices.

- Handling non-linearity and complex patterns:

The stock market is characterized by complex and volatile data. LSTMs are good at handling non-linear patterns and complexity in data, as they can understand and analyze the non-linear relationships between factors influencing stock prices.

- Ability to learn from the past:

LSTMs can learn complex relationships and patterns in data, thanks to their ability to use short-term memory to store and utilize important information for prediction.

- Dealing with the vanishing gradient problem:

LSTMs have a mechanism called "gates" that helps overcome the vanishing gradient problem, which is a problem faced by traditional neural networks when trying to train models on long time sequences.

Despite these benefits, the use of LSTM should be done carefully, often requiring consideration of other factors such as data quality, data volume, and indicators used to ensure achieving the best performance in stock price prediction task.[13]

3.3. Autoregressive Integrated Moving Average (ARIMA)

A linear model called ARIMA combines autoregressive (AR), moving average (MA), and differencing (I) components. It uses three parameters, p, d, and q, to model seasonality, trend, and noise in a time series.

The parameter p represents the autoregressive aspect of the model, incorporating past values. For example, if it rained heavily in the past few days, we could forecast that it is likely to rain tomorrow as well. The parameter d is associated with the integrated part of the model, determining the amount of differencing applied to the time series. For instance, if the daily amounts of rain have been similar over the past few days, we can forecast that tomorrow's rainfall will be similar to todays. The parameter q is related to the moving average part of the model.

If our model includes a seasonal component (explained later), we use a seasonal ARIMA model (SARIMA) with additional parameters: P, D, and Q. These parameters have the same associations as p, d, and q but correspond to the seasonal aspects of the model. Our methods only consider data from a univariate time series, focusing solely on the relationship between the y-axis value and the x-axis time points. We do not take into account external factors that may influence the time series. [6], [19]

4. Experiments Authors

The present study was conducted using the historical price data of four high volume stocks; namely, Saudi Arabian Mining Co, Yanbu Cement Co., Sabic, Saudi Indian and TSLA as a benchmark; over a period of 1132, days; specifically, from 1 January 2019 to 23 July 2023. They were obtained from the website https://www.investing.com/equities/ on 25/7/2023. Only the closing prices of the aforementioned stocks were used to univariately forecast their closing prices for the subsequent seven days. The company's data was used, and the general market index was not used because the aim of the research is to assist investors in identifying the appropriate company for investment. The general index provides an overall view of market behavior and cannot determine a specific company. The dataset was divided into two groups, the first for training with a total of 906 records, which is 80%, and the second for testing with a total of 226 records, which is 20%. Table 2. provides a summary of the statistics of the input data:

	Saudi Arabian Mining	Saudi Basic Industries	Saudi Riyal Indian	Yanbu Cement	TSLA
count	1132	1132	1132	1132	1132
mean	25.491422	103.337102	20.040194	36.194373	163.507069
std	13.164116	16.615524	1.077733	5.508642	110.823442
min	9.330000	62.000000	18.230000	22.200000	11.930000
25%	14.192500	89.875000	19.180000	32.737500	37.510000
50%	18.865000	100.200000	19.850000	36.800000	186.790000
75%	40.670000	120.650000	20.690000	40.050000	246.690000
max	56.930000	139.000000	22.100000	48.450000	409.970000

Table 2. The statistics of the input data

In this research, the lag feature is used in data analysis and prediction, particularly in time series analysis. Time series analysis involves analyzing data that changes over time, such as price series, monthly revenues, and industrial production. Lag features involve using previous values of a variable as inputs in a predictive model to forecast future values of the same variable. For example, in analyzing a time series of commodity prices, lag features can be created based on past prices. By incorporating these lag features, the model can better predict future values since current price changes can influence future values. Similarly, when analyzing sales data for a specific product, a lag feature can be created using the sales from the previous month. Including this lag feature as an input in the prediction model improves the accuracy of future forecasts. In summary, lag features introduce a temporal aspect to analysis and prediction models, allowing them to capture relationships and transitions in data over time. This incorporation enhances their ability to predict future events.

4.1. Analyzing Autocorrelation and Partial Autocorrelation

Autocorrelation (ACF) and partial autocorrelation (PACF) are vital tools for understanding the underlying patterns in our time series data. ACF measures the correlation between a time series and its lagged version, while PACF measures the exact correlation while accounting for the effect of other lags. By examining the ACF and PACF plots, we can identify the appropriate lags for our forecasting models.

4.2. Forecasting process

The first LSTM model parameter that requires tuning is the number of training cycles. The next parameter is the size of the batch, which determines the frequency of updating the weights of the network. The third factor that affects the system's learning capacity is the number of neurons. The Adam-optimized approach was employed as it is based on the stochastic gradient descent optimization algorithm, which is popularly used in DL. The ability of a system to learn an issue structure often correlates with the number of neurons present, even when training time is lengthened. However, a higher learning capability causes the issue of training data overfitting. Table 3 includes a list of the test parameters as well as average values for the prediction accuracy, including the mean squared error (MSE), mean absolute error (MAE), and root mean squared error (RMSE) of all the cases.

Experiment	Epochs	Batch Size	Neurons	MAE	MSE	RMSE
1	120	8	4	0.58	0.34	0.346
2	150	4	4	0.43	0.49	0.518
3	170	4	5	1.53	4.48	2.621
4	190	4	5	0.37	0.28	0.56
5	210	4	6	1.46	4.28	2.586

Table 3. Experiment Specifics Experiments

The findings presented in the table showed that Test 4 yields the most optimal results. Table 3 contains the information related to the sample that was selected and consists of four companies listed on the TADAWUL Saudi stock market.

The LSTM and ARIMA models were used to process all the data. The dataset of the TSLA was further divided into testing and training datasets. The training dataset comprised the initial 1132 entries and was used to create the models. The other entries made up the testing dataset and were used to verify the model by forecasting the results.

The time step of every dataset increased by one. The model was used to forecast the time steps. The anticipated values of the testing set were entered in the model to estimate the subsequent step. This scenario is analogous to a real-world scenario in which the most recent stock market information is retrievable on a daily basis and used to forecast the following day's prices. The testing dataset-based predictions were compiled, and their error rates were computed to determine the forecasting capability of the model, which was then tuned using the MAE, MSE, and RMSE metrics, where the higher errors were adjusted to enable more optimized results that were consistent with the actual data.

4.3. Model Evaluation

When evaluating a time series machine learning model, it is crucial to conduct error analysis. This involves calculating the discrepancy between the predicted and actual values for each data point, allowing you to assess the model's accuracy. There are various methods for error analysis, including the choice of evaluation metrics such as mean squared error (MSE), mean absolute error (MAE), or root mean squared error (RMSE). The use of mean squared error is suitable for models that can adapt to sudden changes in the data distribution. On the other hand, mean absolute error is more appropriate for models that do not swiftly react to changes in the data distribution.

5. Discussion and results

To validate the model, the researchers employed Python 3.8 which could be run using the Windows 10 operating system (OS). Many Python libraries were employed in this present study, including the fundamental object in Numpy which is a homogeneous multidimensional array. It could be seen as the element table, which normally contains numbers of the same value. Python's built-in Pandas module offers versatile, quick, and expressive process-es that are designed to produce "labelled" and "relational" data that can be interpreted easily. A library called Sklearnit offers simple and effective data assessment.

Also, Theano, TensorFlow, or the MS Cognitive Toolkit can all be used for executing the Keras library. This package focuses on modularity, extensibility, and user-friendliness and helps in rapid testing using DNNs. Finally, the NumPy mathematical extension is used in addition to the Matplotlibit module, a Python plotting tool. The data presented in Figure 4, Figure 5, Figure 6, and Figure 7. Table 4 offers information on the actual and forecasted stock prices as well as data on the accuracy of the forecasts and indicators of errors, such as MSE, MAE, and RMSE appears in Table 5 and Figure 8. Furthermore, presents the brief of the accuracy generated by applying the recommended LSTM framework. The obtained results show a very small error rate, indicating the quality of the results and the ability of the LSTM algorithm to predict stock prices.

Company		Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
Saudi Arabian Mining Co.	Actual	43.40	43.80	44.25	44.40	44.50	44.30	44.30
	LSTM	43.47	43.81	44.18	44.31	44.40	44.23	44.23
	ARIMA	43.21	43.53	43.92	44.30	44.29	44.01	44.21
Yanbu Cement Co.	Actual	40.00	40.50	40.95	40.60	40.60	40.45	39.45
	LSTM	40.19	40.68	41.09	40.77	40.77	40.623	39.65
	ARIMA	39.9,	40.2	40.0,	39.8,	39.9,	40.0,	40.2
Sabic	Actual	84.4	85.5	86.1	87.0	88.2	88.4	88.5
	LSTM	84.12	85.30	85.94	86.91	88.19	88.40	88.51
	ARIMA	85.0	85.2	85.8	86.2	87.5	87.2	88.9
Saudi Indian	Actual	21.84	21.87	21.86	21.86	21.87	21.87	21.85
	LSTM	21.82	21.84	21.83	21.83	21.84	21.85	21.83
	ARIMA	21.90	21.81	21.94	21.99	21.92	21.94	21.91

Table 4. Forecasts for The Subsequent Seven Days

Company		MAE	MSE	RMSE
Saudi Arabian Mining Co.	LSTM	0.0693	0.0057	0.0757
	ARIMA	0.154	0.0914	0.101
Yanbu Cement Co.	LSTM	0.1724	0.0301	0.1732
	ARIMA	0.413	0.0984	0.289
Sabic	LSTM	0.1109	0.0229	0.1514
	ARIMA	0.214	0.131	0.214
Saudi Indian	LSTM	0.0310	0.0009	0.0310
	ARIMA	0.195	0.0451	0.214

Table 5. Forecasts for The Accuracy

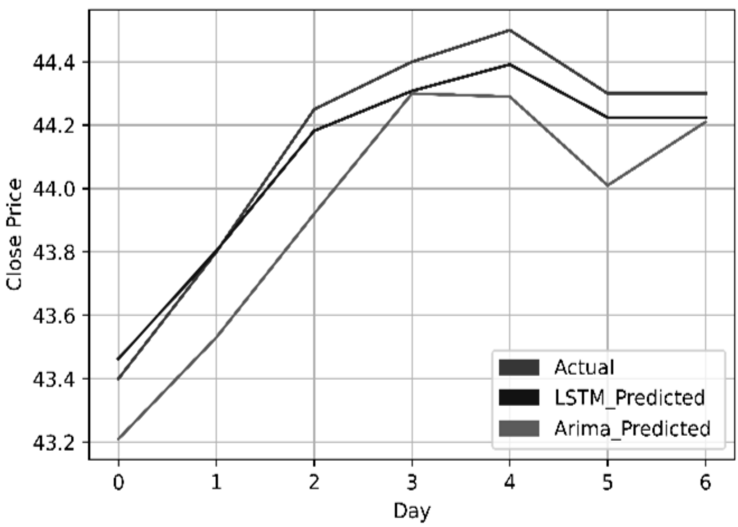

Figure 4. Forecasts for the Saudi Mining Co.

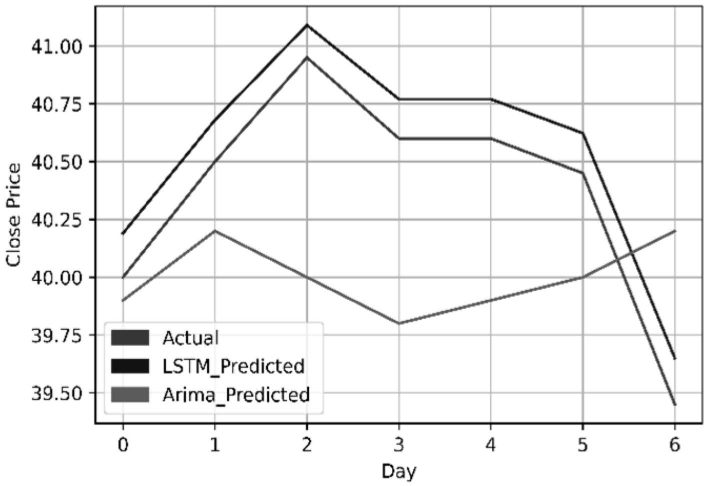

Figure 5. Forecasts for the Yanbu Cement Co.

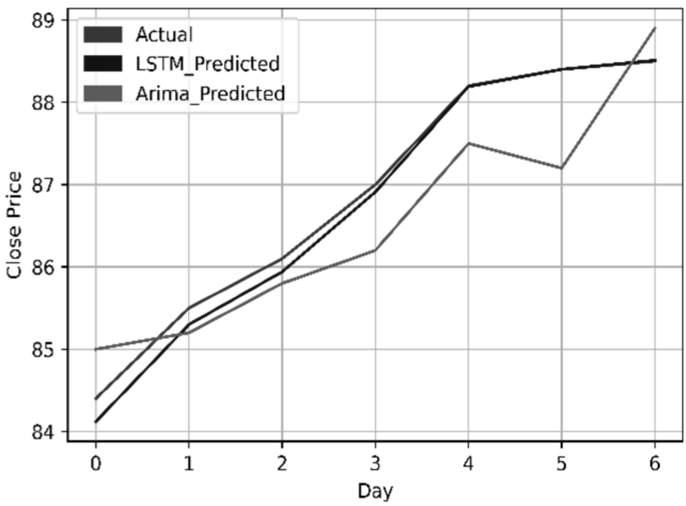

Figure 6. Forecasts for Sabic

The second step involved testing the TSLA Index using the ARIMA and LSTM models. Data related to the model results over the one-week period is presented in Figure 7 and Table 6.

Index		**Day 1**	**Day 2**	**Day 3**	**Day 4**	**Day 5**	**Day 6**	**Day 7**
TSLA	Actual	23.1870	23.0596	22.6296	22.4180	22.3859	21.2397	20.0817
	LSTM	23.000	22.5700	22.3600	22.3300	21.1800	20.0200	20.6700
	ARIMA	26.063	25.074	25.424	25.095	24.477	23.126	23.780

Table 6. Forecast One Week (TSLA Index)

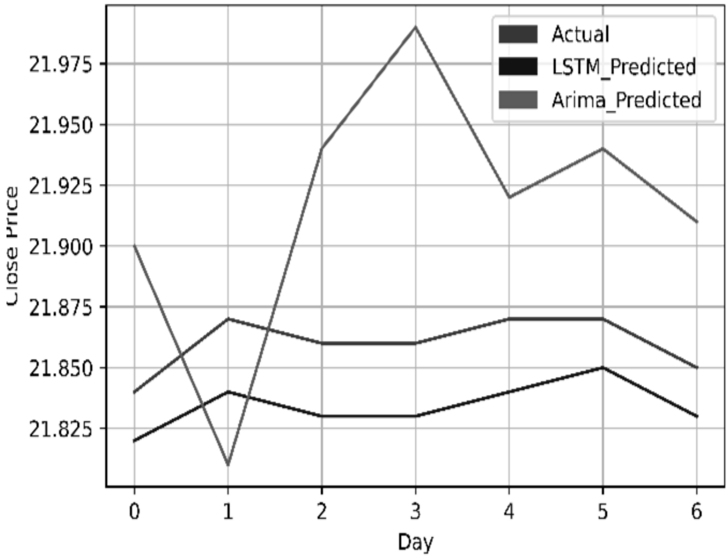

Figure 7. Forecasts for Saudi Indian

Comparison of the proposed LSTM and ARIMA models' forecasts vs. actual performance from the TSLA index over a one-week period. The accuracy of the forecasts and indicators of errors, such as MSE, MAE, and RMSE appears in Table 7 And Figure 8.

Index		**MAE**	**MSE**	**RMSE**
TSLA	LSTM	0.578	0.520	0.721
.	ARIMA	0.989	0.951	1.041

Table 7. Forecast for The Accuracy (TSLA Index)

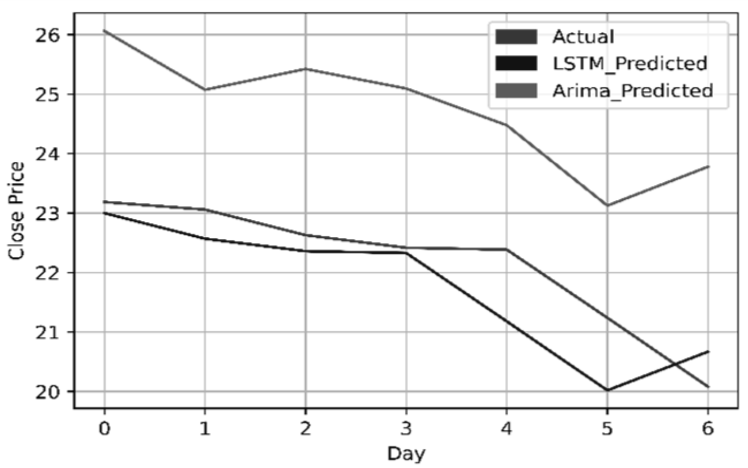

Figure 8. TSLA index

5. Conclusions and future work

The current study investigated strategies for predicting stock prices in the near future. The study applied the ARIMA algorithm and the LSTM algorithm after using DWT to eliminate noise from the input data. These predictions aid investors in making informed decisions regarding stock market investments. Below, we summarize our findings for each algorithm:

ARIMA is a powerful model that yielded the best results for stock data. However, it requires precise hyperparameter tuning and a good understanding of the data.

Recurrent Neural Networks (RNNs) based on LSTM are particularly effective for learning from sequential data, including time series. These models shine when working with large datasets and can uncover complex patterns. Unlike ARIMA, they do not rely on specific assumptions about the data, such as time series stationarity or the presence of a timestamp. However, interpreting the behavior of LSTM-based RNNs can be challenging. Additionally, achieving good results necessitates precise hyperparameter tuning.

Future studies on this subject should consider various factors to enhance prediction accuracy. For instance, the impact of events and annual seasons like Umrah, Hajj, and Ramadan on stock price movement accuracy should be examined using trading data.

Acknowledgements

This research work was funded by Institutional Fund Projects under grant no. (IFPIP:1107-611-1443). The author gratefully acknowledges technical and financial support provided by the Ministry of Education and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

References

[1] D. Gaur, and S. Kumar Dubey, “Development of Activity Recognition Model using LSTM-RNN Deep Learning Algorithm,” *Journal of Information and Organizational Sciences,* vol. 46, no. 2, pp. 277-291, 2022.

[2] D. P. Gandhmal, and K. Kumar, “Systematic analysis and review of stock market prediction techniques,” *Computer Science Review,* vol. 34, pp. 100190, 2019.

[3] J. A. Nasir, O. S. Khan, and I. Varlamis, “Fake news detection: A hybrid CNN-RNN based deep learning approach,” *International Journal of Information Management Data Insights,* vol. 1, no. 1, pp. 100007, 2021.

[4] M. Sadiq, C.-Y. Lin, K.-T. Wang, L. M. Trung, K. D. Duong, and T. Q. Ngo, “Commodity dynamism in the COVID-19 crisis: are gold, oil, and stock commodity prices, symmetrical?,” *Resources policy,* vol. 79, pp. 103033, 2022.

[5] M. Loxton, R. Truskett, B. Scarf, L. Sindone, G. Baldry, and Y. Zhao, “Consumer behaviour during crises: Preliminary research on how coronavirus has manifested consumer panic buying, herd mentality, changing discretionary spending and the role of the media in influencing behaviour,” *Journal of risk and financial management,* vol. 13, no. 8, pp. 166, 2020.

[6] D. Wei, "Prediction of stock price based on LSTM neural network." pp. 544-547.

[7] M. Diqi, M. E. Hiswati, and A. S. Nur, “StockGAN: robust stock price prediction using GAN algorithm,” *International Journal of Information Technology,* vol. 14, no. 5, pp. 2309-2315, 2022.

[8] M. Ali, D. M. Khan, H. M. Alshanbari, and A. A.-A. H. El-Bagoury, “Prediction of complex stock market data using an improved hybrid emd-lstm model,” *Applied Sciences,* vol. 13, no. 3, pp. 1429, 2023.

[9] P. R. Patil, D. Parasar, and S. Charhate, “Wrapper-Based Feature Selection and Optimization-Enabled Hybrid Deep Learning Framework for Stock Market Prediction,” *International Journal of Information Technology & Decision Making,* pp. 1-26, 2023.

[10] B. Singh, S. K. Henge, S. K. Mandal, M. K. Yadav, P. T. Yadav, A. Upadhyay, S. Iyer, and R. A. Gupta, “Auto-Regressive Integrated Moving Average Threshold Influence Techniques for Stock Data Analysis,” *International Journal of Advanced Computer Science and Applications,* vol. 14, no. 6, 2023.

[11] A. Almaafi, S. Bajaba, and F. Alnori, “Stock price prediction using ARIMA versus XGBoost models: the case of the largest telecommunication company in the Middle East,” *International Journal of Information Technology,* vol. 15, no. 4, pp. 1813-1818, 2023.

[12] 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.

[13] M. Jarrah, and M. Derbali, “Predicting Saudi Stock Market Index by Using Multivariate Time Series Based on Deep Learning,” *Applied Sciences,* vol. 13, no. 14, pp. 8356, 2023.

[14] H. U. Amin, R. Ullah, M. F. Reza, and A. S. Malik, “Single-trial extraction of event-related potentials (ERPs) and classification of visual stimuli by ensemble use of discrete wavelet transform with Huffman coding and machine learning techniques,” *Journal of neuroengineering and rehabilitation,* vol. 20, no. 1, pp. 70, 2023.

[15] S. P. Kumar, A. Daripelly, S. M. Rampelli, S. K. R. Nagireddy, A. Badishe, and A. Attanthi, "Noise Reduction Algorithm for Speech Enhancement." pp. 1-5.

[16] M. O. Alassafi, M. Jarrah, and R. Alotaibi, “Time series predicting of COVID-19 based on deep learning,” *Neurocomputing,* vol. 468, pp. 335-344, 2022.

[17] P. B. Weerakody, K. W. Wong, and G. Wang, “Policy gradient empowered LSTM with dynamic skips for irregular time series data,” *Applied Soft Computing,* vol. 142, pp. 110314, 2023.

[18] R. Zakovskis, A. Draguns, E. Gaile, E. Ozolins, and K. Freivalds, "Gates are not what you need in RNNs." pp. 304-324.

[19] W. Wei, G. Wang, X. Tao, Q. Luo, L. Chen, X. Bao, Y. Liu, J. Jiang, H. Liang, and L. Ye, “Time series prediction for the epidemic trends of monkeypox using the ARIMA, exponential smoothing, GM (1, 1) and LSTM deep learning methods,” *Journal of General Virology,* vol. 104, no. 4, pp. 001839, 2023.