Scientific Publications

Journal of Artificial Intelligence and Big Data, 2022, 2, 1341
www.scipublications.com/journal/index.php/jaibd
DOI: 10.31586/jaibd.2022.1341

Article

Big Data-Driven Time Series Forecasting for Financial Market Prediction: Deep Learning Models

Jayakeshav Reddy Bhumireddy [1,*], Rajiv Chalasani [2], Mukund Sai Vikram Tyagadurgam [3], Venkataswamy Naidu Gangineni [4], Sriram Pabbineedi [5], Mitra Penmetsa [3]

[1] University of Houston, USA
[2] Sacred Heart University, USA
[3] University of Illinois at Springfield, USA
[4] University of Madras, Chennai, India
[5] University of Central Missouri, USA

*Correspondence: Jayakeshav Reddy Bhumireddy (jayakeshav10807@gmail.com)

Abstract: Financial markets have become more and more complex, so has been the number of data sources. Stock price prediction has hence become a tough but important task. The time dependencies in stock price movements tend to escape from traditional models. In this work, a hybrid ARIMA-LSTM model is suggested to enhance accuracy of stock price forecasts. Based on time series indicators like adjusted closing prices of S&P 500 stocks over a decade (2010–2019), the ARIMA-LSTM model combines influences of both autoregressive time series forecasting with the substantial sequence learning property of LSTM. Data preprocessing in all aspects including missing values interpolation, outlier's detection and data scaling – Min-Max guarantees data quality. The model is trained on 90/10 training/testing split and met with main performance metrics: MaE, MSE & RMSE. As indicated in the results, the proposed ARIMA-LSTM model gives a MAE value and MSE value of 0.248 and 0.101 respectively and RMSE of 0.319, a measure high accuracy on stock price prediction. Coupled comparative analysis with other Artificial Neural Networks (ANN) and BP Neural Networks (BPNN) are examples of machine learning reference models, further illustrates the suitability and superiority of ARIMA-LSTM approach as compared to the underlying models with the least MAE and strong predictive capability. This work demonstrates the efficiency of integrating the classical time series models with deep learning methods for financial forecasting.

Keywords: Stock Price Prediction, ARIMA-LSTM, Time Series Forecasting, Financial Market, Machine Learning

How to cite this paper:
Bhumireddy, J. R., Chalasani, R., Tyagadurgam, M. S. V., Gangineni, V. N., Pabbineedi, S., & Penmetsa, M. (2020). Big Data-Driven Time Series Forecasting for Financial Market Prediction: Deep Learning Models. *Journal of Artificial Intelligence and Big Data, 2*(1), 153–164. DOI: 10.31586/jaibd.2022.1341

Received: September 19, 2022
Revised: November 17, 2022
Accepted: December 3, 2022
Published: December 27, 2022

1. Introduction

In the realm of financial forecasting, here the demand of accurate decision making is the highest because they have a direct implication for investment decision-making and portfolio management, and market analysis as well [1]. Monitored by a variety of indicators (economic indicators, geopolitics, and investor sentiment), financial markets are very suitable for commodity trading, thanks to its high dynamics and the impossibility of prediction using traditional linear models [2]. Although approaches such as ARIMA have provided the grounds for time series finance, they are sometimes not useful in describing how complex the patterns are within market data. Financial time series data are nonstationary, showing volatility clustering and the sudden shifts elements that traditional models are not able to model [3]. This has resulted on the increasing research in ML and DL methods which are flexible and performing well in handling the intricacies of finance data.

In several technical and scientific domains, including telecommunications and finance, time series forecasting is an essential scientific discipline. Specifically, financial markets forecasting is an important issue for research since predictive estimates are needed to guide investments, optimize market strategies, and provide valuable insights for the risk management process [4]. For a long time now, traditional techniques used in time series forecasting, ARIMA, have been the backbone of financial forecasting. Still, these methods tend to find it difficult to handle the fundamental intricacies of financial data, which may be nonlinear, noisy, and volatile. Consequently, there has been increased interest in how to identify advanced methods that can overcome these limitations. In particular, multi-step forward forecasting has grown to be a challenging issue. Multi-step ahead activities, in contrast to one-step ahead forecasting, are plagued by a number of problems, including the accumulation of prediction inaccuracies, decreased precision, and an increase in uncertainty over time [3]. These challenges are particularly acute in the financial market where data is highly dynamic, and the models must be able to change to account for constant rates of change. That is why, to make effective forecasts of the financial market's trends, complex techniques that can process a high dimension are required for a non-linear connection between the variables.

In the past few years DL models have received great attention for the problem of time series prediction capable of identifying complex patterns in an enormous set of data [5]. DNN [6], CNN, as well as LSTM networks, has demonstrated spectacular performance when it comes to time series forecasting since with it they are able to learn and recognize complex relationships in data without performing complex feature engineering [7]. With the rise of big data including enormous historical information on markets, the use of DL techniques in financial forecasting has significantly increased [8]. By exploiting big dataset these models are able to take into considerations of many kinds of market conditions optimizing both their predictive capabilities and generalization.

The change to data driven methods from linear statistical models is an essential first step in time series forecasting. Although linear assumptions of the traditional methods DL models are more resilient since ARIMA cannot manage nonlinearities in the financial data [9]. These models can then be trained end to end on raw data such that they can detect patterns that conventional models may fail to detect. The ability to automatically pick up nonlinear dependencies and interactions between a number of factors places deep learning as an effective tool for financial market prediction.

1.1. Motivation and Contribution of the Study

The driving force of this research is evoked by the rise of complexity and volatility of financial markets, requiring sophisticated procedures related to prediction to improve investment judgment. Complex patterns and underlying correlations in time series data are difficult for conventional stock price forecasting algorithms to extract. Therefore, there is an urgent demand for data-driven methods that can use ML and big data to increase forecast accuracy. Using DL models like ARIMA, this research aims to provide a strong basis for predicting stock price correlation coefficient cooperation, as well as LSTM in improved understanding of market dynamics and investment strategies. The following are the main contributions of this study:

- Presents an integrated ARIMA-LSTM framework; competent to not only capture to improve forecast accuracy, use both there are both linear and non-linear trends in the financial time series data.
- Implements robust techniques for handling missing values, outlier detection, and normalization, ensuring high-quality input data for model training.
- Applies feature selection methods to identify the most relevant predictors, optimizing the forecasting capabilities of the hybrid model.

- Evaluates the model using standard performance metrics (MAE, MSE, RMSE) and demonstrates its potential for real-time stock price forecasting to support better investment and portfolio management decisions.

1.2. Justification and Novelty of the Paper

The justification and novelty of this paper lie in the application of advanced AI-driven models for accurate stock price correlation coefficient prediction in financial markets. This study enhances forecasting precision through a robust data pre-processing framework that includes handling missing values, outlier detection, as well as normalization methods like Min-Max scaling. In a comparative study, the ARIMA-LSTM model shows that it is capable of identifying time series data's linear and non-linear correlations, outperforming traditional forecasting methods. Furthermore, this research explores the implications of real-time predictive analytics in investment strategies, enabling timely decision-making, reducing risks associated with market volatility, and ultimately enhancing portfolio management in dynamic financial environments.

1.3. Structure of the paper

The paper is structured as follows: Section II reviews related work on stock price prediction. Section III details the methodology, including data collection and model implementation. Results and comparative analysis result section appears in section IV while section V concludes with insights and future research direction.

2. Literature Review

This section examines and highlights the position of Time Series Forecasting in the financial markets. It covers existing methodologies in forecasting, their deficiencies and necessity of advance AI driven models. Among the reviewed works there are:

Ma et al. (2019) introduced ANN and SVM-based Forecasting a hybrid dynamic stock time series model, improves feature quality, forecasts stock fluctuation, where dynamic adjustment bias enhances prediction accuracy. The model's assessment was done using China stock exchange data between June 8 2015 and May 26 2016, having an accuracy of 79%. The model's effectiveness is demonstrated through data pre-processing, dynamic adjustment bias, and improved feature quality [10].

Liu and Liu (2018) suggested a two-step pre-processing strategy for the trend indicator and A technique for preparing statistics based on movement trends. The GRU was used to simulate the stock index. Five characteristics were used to extract trend indicators, which were then discretized according to the dynamic connection in the two-stage pre-processing. Additionally, financial time series forecasting was done using three recurrent neural networks. Our approach increased projection of the pattern of stock index movement from 33% to 68% in contrast to the random tri-prediction method [11].

Tsang, Deng and Xie (2018) introduced a novel deep time-series data modelling technique for LSTM-based stock market index forecasting. The method uses a dataset of six global market indices to forecast the day after's closing price. According to experimental data, it is feasible and produces notable outcomes in financial market prediction, with an average yearly profitability performance of up to 200% [12].

Raimundo and Okamoto (2018) introduced an adaptive hybrid prediction model for Combining SVR with wavelet models, the SVR-wavelet model predicts time series related to finance, specifically for the FOREX market. The data from FOREX time series is broken down by the model using the DWT, and the resulting SVR input variables are utilized to create new predictions [13].

Althelaya, El-Alfy and Mohammed (2018) examined how DL techniques may be used into stock market prediction. Several DRNN variations evaluated and contrasted with LSTM and GRU. For predicting in the short and long term, Multivariate inputs were utilized in both unidirectional and bidirectional stacked structures. Additionally, the DL architectures were compared to Simplified neural networks with historical S&P 500 index data [14].

Beyaz et al. (2018) explores the use of ML in forecasting stock prices by identifying and accounting for market mood states. It uses clustering methods to identify different market states and develops forecasting models for selected companies. Results show that accounting for market for a 126-day forecasting horizon, mood increases predicting accuracy 47% of the time [15].

Bakhach, Tsang and Jalalian (2017) concentrated on applying the DC framework for forecasting the course of trends in the foreign exchange market. It seeks to ascertain if a certain proportion of the present trend will continue before it terminates. Three currency pairings are used to test the method, which employs one independent variable: EUR/CHF, USD/JPY, and GBP/CHF. The experimental results are quite accurate, sometimes exceeding 80% [16].

The comparative analysis of background study based on their Methodology, Data, Key Findings, Limitations and Future work are provided in Table 1.

Table 1. Summary of the related work on Time Series Forecasting in the financial market

Ref	Methodology	Dataset	Performance	Limitations & Future Work
Ma et al. (2019)	Hybrid model combining ANN and SVR with dynamic adjustment bias	China Stock Exchange data (June 8, 2015 – May 26, 2016)	79% accuracy rate	Future work could involve more diverse market conditions and cross-market validation
Liu and Liu, (2018)	GRU with movement trend-based preprocessing (two-step: trend extraction + discretization)	Predicting the pattern of stock index movement	Accuracy improved from 33% to 68%	Needs validation across different indices and longer time frames
Tsang, Deng and Xie, (2018)	Deep LSTM-based time-series prediction	To predict the next day's closing price, six global market indicators	Annual profitability up to 200%	Risk-adjusted returns and robustness under volatile conditions not explored
Raimundo and Okamoto, (2018)	SVR-Wavelet hybrid model using DWT for input enhancement	FOREX market time series data	Improved financial series prediction accuracy	Scalability to other financial domains and high-frequency data not addressed
Althelaya, El-Alfy and Mohammed, (2018)	Deep RNNs (LSTM and GRU, uni- & bi-directional, stacked) with multivariate input	S&P500 historical index data	Superior to shallow networks in short/long-term forecasting	Needs improved interpretability and computational efficiency for real-time systems
Beyaz et al. 2018	Machine learning with clustering to account for market mood states	Forecasting selected company stock prices with 126-day horizon	Mood-based forecasting improved accuracy in 47% cases	Further study needed on integrating sentiment analysis with fundamental and technical data
Bakhach, Tsang and Jalalian, (2017)	Directional Change (DC) framework for trend prediction	Forex market (EUR/CHF, GBP/CHF, USD/JPY)	Accuracy often exceeded 80%	Limited to one independent variable; expansion to multivariate models is needed

3. Methodology

The methodology for Time Series Forecasting involves a structured approach similar to what is seen in Figure 1. The process's initial stage is to collect information from the adjusted closing prices of the stocks that make up the S&P 500. Data quality is improved by data preparation methods such as Min-Max normalization, outlier identification, and managing missing information. The most relevant predictors for the forecasting models are found using feature selection techniques. The ARIMA and LSTM models were then created using the processed datasets for training in order to identify the relationships between time series data that are both linear and non-linear. We employ performance measures like MAE, MSE, and RMSE to evaluate the models and make sure they are reliable. Last but not least, the architecture is set up for real-time analysis, which facilitates better investment choices by accurately predicting stock price movements and enhancing portfolio management strategies. Figure 1 depicts the workflow of the complete process from the dataset application to the evaluation of the final results.

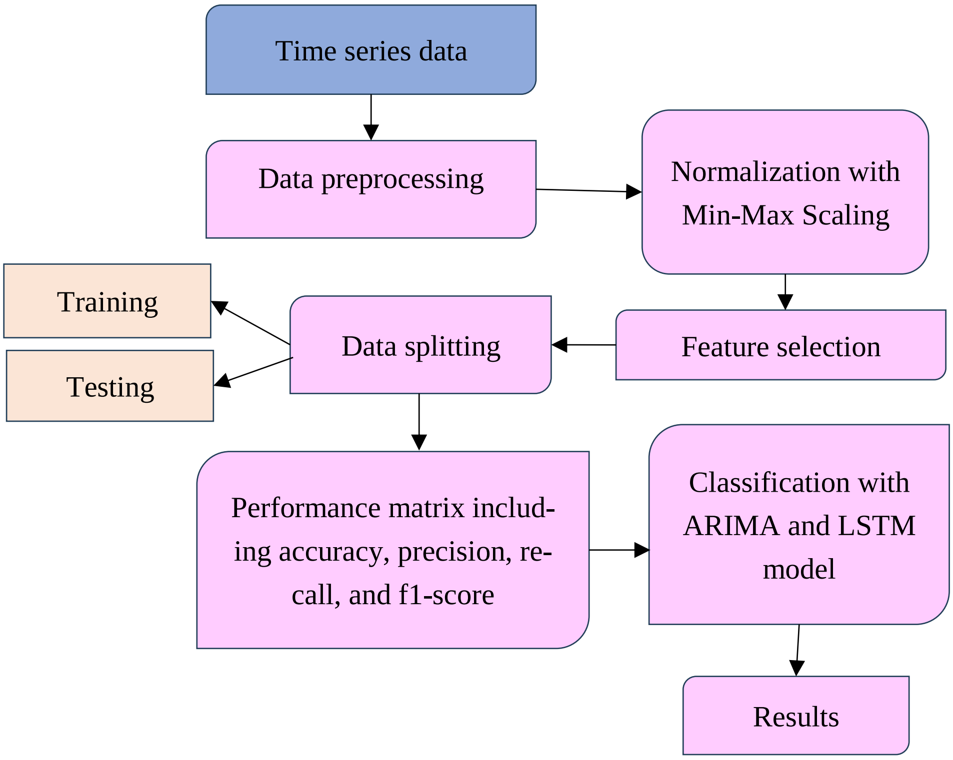

Figure 1. Data Flowchart diagram for Time Series Forecasting in the financial market

The following lists the general procedures of the time series forecasting flowchart for financial markets:

3.1. Data Collection

The data used for stock price correlation coefficient prediction includes the adjusted closing price time series of the companies that make up January 1, 2010, through December 31, 2019: the S&P 500. The data is sourced from the Quandy database, and industry information is gathered from Wikipedia. Python's Beautiful Soup library is utilized for web scraping. After pre-processing, 150 random stocks are selected, and the correlation coefficients for each asset pair are calculated over a 100-day time window. The data distribution is presented below.

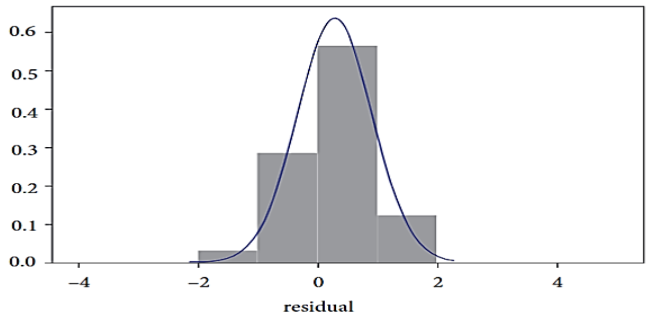

Figure 2. Residual Data Distribution

Figure 2 displays the distribution of residual data, with most values clustered around zero, indicating a relatively unbiased model. The overlaid blue curve suggests a roughly normal distribution, although the histogram shows some deviations, particularly a slightly higher peak and heavier tails than a perfect normal distribution.

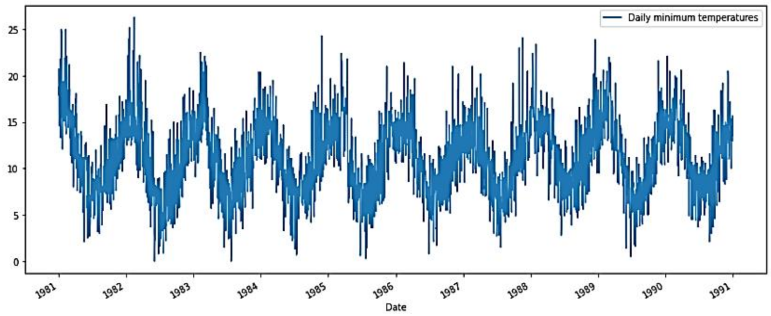

Figure 3. Daily Minimum Temperatures (1981-1991)

Figure 3 displays the daily minimum temperatures recorded between 1981 and 1991. A clear annual cyclical pattern is evident, with lower temperatures typically occurring at the beginning and end of each year and higher temperatures around the middle. There's also a noticeable variability within each year.

Figure 4 illustrates the daily minimum temperatures from 1981 to 1991. The yellow line clearly shows the recurring seasonal pattern, with temperature dips at the start and end of each year and peaks around the middle. The data also exhibits day-to-day fluctuations within these yearly cycles.

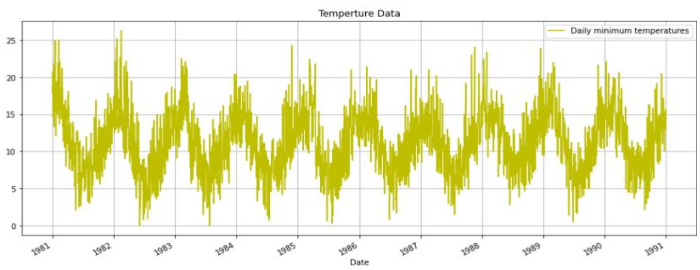

Figure 4. Daily Minimum Temperatures (1981-1991)

3.2. Data Preprocessing

The process of enhancing raw data is known as data preparation and making necessary adjustments to make it more appropriate for analysis. It includes actions like resolving missing values, identifying outliers, and normalizing data to guarantee accurate and dependable model training. The following lists the essential pre-processing steps:

- **Handle missing values:** This is the missing values in the stock price dataset are addressed through interpolation methods, ensuring continuity in time series data and preventing disruptions in model training.
- **Outlier Detection:** Outliers are detected using statistical techniques, to identify and remove extreme values that potentially distort the predictions made by the LSTM and ARIMA models.

3.3. Normalization with Min-Max Scaling

A scaling method called normalization converts a broad feature range into a standard range. It usually lies between zero and one. Min-Max Scaling is one of the most basic and commonly used scaling techniques. It converts characteristics to a predefined range, often [0, 1]. It calculates using Equation (1).

$$X' = \frac{(x - x_{min})}{(x_{max} - x_{min})} \tag{1}$$

where X′ is the scaled value normalized to the range [0, 1], x_{min} and x_{max} are X is the initial feature value, and X is the feature's lowest and maximum values.

3.4. Feature selection

Feature selection is a method of data preparation that lowers the quantity of features in datasets following pre-processing. Feature selection methods go throughout the whole feature space to find the best feature set free of duplicate data and irrelevant features, ultimately enhancing model performance and interpretability while minimizing overfitting and computational complexity in machine learning tasks.

3.5. Data Splitting

The data is separated into two sets: one for training and one for testing. There are two distinct halves of the dataset, with a 90% training to 10% testing ratio.

3.6. Implementation with the ARIMA and LSTM Model

As can be seen below, this study suggests a hybrid model that combines ARIMA and LSTM:

3.6.1. Autoregressive Integrated Moving Average (ARIMA)

A well-liked forecasting technique for time series forecasting is ARIMA. This method uses a linear mix of historical data and random mistakes to make the prediction. Equation (2) is The ARIMA (p, d, q) model's general formula. MA and AR functions are the two main parts of ARIMA. Equation (2)'s p and q parameters stand for the AR and MA functions' respective orders [17]. The third parameter indicates how many differencing operations were applied to the series to make it stationary.

$$(1 - \emptyset_1 B - \emptyset_2 B^2 - \cdots - \emptyset_p B^p)(1 - B)^d \gamma_t = \left(1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q\right) e_t \tag{2}$$

where $\gamma_t$and etet show the error is the variation between, both the fitted value and the actual value. The ARIMA's AR and MA are represented by the parameter sets $\emptyset_i$ (i = 1, 2, …, p) and $\emptyset_j$ (j = 0, 1, 2, ., q). The term (1 − B)d acts as an operator for differences.

3.6.2. Long Short-Term Memory

The LSTM network architecture is based on the RNN. You can see the memory cell's structure in Figure 5. Typically, LSTM is made up of memory cells and gates. Using a variety of LSTM gate units might alleviate the problem of gradient vanishing or explosion during RNN training, which is caused by memory loss for long-term sequences [18]. To make an LSTM model fit a supervised classification task, we may find the activation functions of the network units and use them to create a mapping function that takes an input observation x = (x_1, x_2,...,x_N) and outputs a label y from the set [0,1] for attack detection. A conventional LSTM memory's setup parameters are an input vector (x), a vector that is hidden, and (h^()) from the prior timestep as well as a vector cell for output $h^{<t>}$. The following equations may be used to iteratively establish the memory cell's implementation are as follows in Equations (3-9):

$$i^{<t>} = \sigma(W_i x^{<t>} + W_i h^{<t-1>} + b_i) \tag{3}$$

$$f^{<t>} = \sigma(W_f x^{<t>} + W_f h^{<t-1>} + b_f) \tag{4}$$

$$o^{<t>} = \sigma(W_o x^{<t>} + W_o h^{<t-1>} + b_o) \tag{5}$$

$$u^{<t>} = tanh(W_u x^{<t>} + W_u h^{<t-1>} + b_u) \tag{6}$$

$$c^{<t>} = i^t \odot u^{<t>} + f^{<t>} \odot c^{<t-1>} \tag{7}$$

$$h^{<t>} = o^{<t>} \odot tanh(c^{<t>}) \tag{8}$$

$$y^{<t>} = \phi(u^{<t>} + f^{<t>} \odot c^{<t-1>} \tag{9}$$

where x is the logistic sigmoid function, σ is the current input, and ⊙ denotes element-wise multiplication. Weight matrices are indicated by the W terms, whereas bias vectors are shown by the b terms. The cell input activation is denoted by u. Input gate, forget gate, output gate, and memory cell are stands for "I," "F," "O," and "C," in that order. The method of updating the current memory is determined by the cooperation of these gates. cell $c^{<t>}$ and the current hidden state $h^{<t>}$.

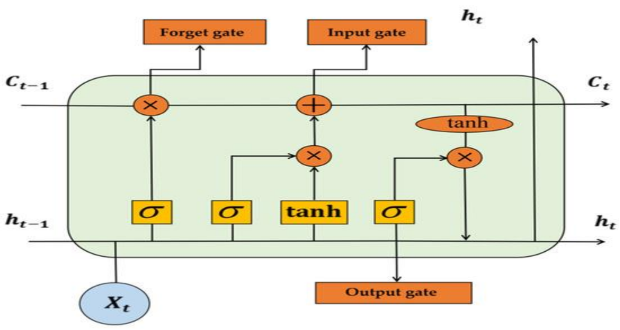

Figure 5. Long Short-Term Cell Structure.

The LSTM's cell input is shown by the bottom blue circle. The rectangular orange section of the forget gates, both the input and the output phases are.

The strengths of both ARIMA, which allows us to see linear patterns, and LSTM, which helps us profile complex nonlinear and the hybrid ARIMA-LSTM model incorporates long-term dependencies. ARIMA takes care of the structured stationary components of time series where LSTM learns complicated patterns through the processing of residuals. This combination improves forecast accuracy, especially for volatile and nonstationary data of the financial sector.

3.7. Performance Matrix

The key metrics, MAE measure of which is the efficiency of the ARIMA and LSTM models is assessed using the mean absolute differences between the forecasted and actual stock prices. To highlight more significant differences, MSE-counting uses the squared error average, whereas RMSE, MSE square root, a prediction error metric expressed in stock price units.

3.7.1. Mean Absolute Error (MAE)

The mean absolute difference between what was predicted and what was actually is computed using the popular MAE measure. For each time series observation, it is computed as the average of the absolute differences between what was predicted and what was actually found. When predicting the amount of predicted errors without accounting for direction, MAE is especially helpful. Equation provides a numerical representation of it Equation (10).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i| \tag{10}$$

3.7.2. Mean Squared Error (MSE)

The anticipated and actual values' MSE is calculated. Larger mistakes are highlighted and penalized more severely than smaller ones when the errors are squared. As shown in equation, MSE offers a measurement of the total variance of predicted errors Equation (11).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2 \tag{11}$$

3.7.3. Root Mean Squared Error (RMSE)

The square root of the MSE, or RMSE, is frequently used to describe errors in the same units as the original time series data. It provides an easier-to-understand indicator of the average forecast error. It is stated mathematically in an Equation (12).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2} \tag{12}$$

The efficacy of the model for time series forecasting and comparative analysis is evaluated using these model performance matrices.

4. Result Analysis and Discussion

This study evaluates how well the LSTM and ARIMA models predict stock prices. The experiments were performed using Python 3 with TensorFlow and Scikit-Learn on a 64-bit Windows 10 PC with an Intel i7 CPU with four cores running at 3.60 GHz and 16 GB of RAM. With a MAE of 0.248, the ARIMA-LSTM model achieved excellent prediction accuracy based on the performance parameters displayed in Table 2. The MSE of 0.101 shows the model's success of reducing bigger errors while the RMSE of 0.319 confirms its effectiveness in future stock prices prediction. These metrics combined explain the model's dependability and consistency in applications involving financial forecasting.

Table 2. Results of proposed ARIMA-LSTM Model

Matrix	ARIMA-LSTM Model
MSE	0.101
RMSE	0.319
MAE	0.248

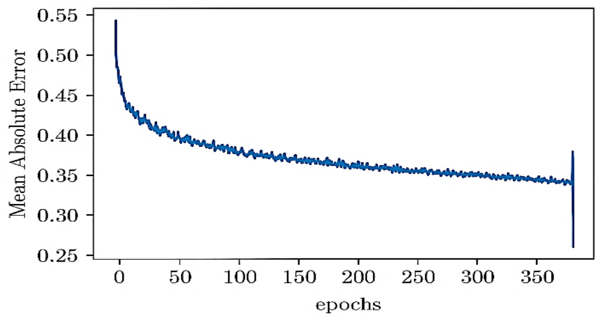

Figure 6. Testing Progress of ARIMA-LSTM Model Based on MAE

Figure 6 shows the ARIMA-LSTM model's training results is illustrated over 370 epochs it can be seen that MAE is also steadily decreasing. First of all, higher than 0.50, the MAE continues decreasing, approaching 0.30. The trend shows that the model learns well and enhances predictive accuracy as training progresses.

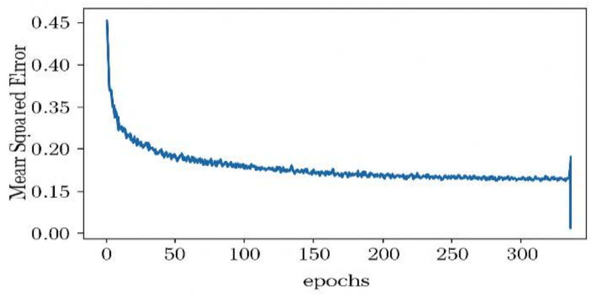

Figure 7. Testing Progress of ARIMA-LSTM Model Based on MSE

The MSE (see Figure 7) during training the ARIMA-LSTM model. From approximately 0.45, the MSE decreases very fast and stabilizes later at around 0.15 after approximately 100 epochs. The trend is indicative of the process of effective learning, as well as the prediction error, where the reduced prediction error corresponds to the model performance's improved performance after successive training epochs.

4.1. Comparative Analysis

This section compares many machine learning algorithms for stock price prediction. A comparison using MAE is presented in Table 3. The BPNN [19] model performed moderately since its MAE was 0.2556. The ANN [20] In the case of model, significantly higher MAE is recorded (20.06) indicating poor accuracy. of these, the ARIMA-LSTM Model produced MAE of 0.248, indicating that it is better at learning complex temporal dependencies and providing the most accurate stock price predictions of all the models in this analysis, and is therefore the best performing model in this analysis.

Table 3. Comparative Analysis of proposed and baseline Models for time series in forecasting

Matrix	BPNN [19]	ANN [20]	ARIMA-LSTM
MAE	0.2556	20.06	0.248

Comparative analysis of forecasting models, using MAE shows the superiority of proposed ARIMA-LSTM model illustrate in Table III. Given that it shows up with a MAE of 0.248, the ARIMA-LSTM out-performs the BPNN and ANN with MAE value of 0.2556 and 20.06 respectively. Although the predictive accuracy of BPNN is relatively moderate, The ANN's much larger inaccuracy suggests that it does not adequately grasp the complexities of the time-limited financial data series. Because The advantages of linear and non-linear models are successfully combined in the hybrid ARIMA-LSTM technique, it may be more suited to identifying complex market trends and producing predictions that are more accurate and reliable.

The suggested ARIMA-LSTM model has various benefits, namely, the capacity of financial time series data to capture both linear and nonlinear tendencies, better forecast accuracy with a considerably lower MAE compared to more conventional models like ANN and BPNN. Its hybrid structure makes it very resistant to market fluctuations, volatility, and sudden jolts, common for financial data. Furthermore, the model does well in multi-step ahead forecasting, while exhibiting accuracy and stability in long horizons. Requiring little manual feature engineering, the model utilizes the power of LSTM to obtain complex patterns directly from the data, thereby being potent and efficient in predicting financial markets.

5. Conclusion and Future Scope

The study's findings showed that the proposed ARIMA-LSTM hybrid model was suitable. for the current inquiry, to forecast stock prices combining linear and nonlinear time series modelling approaches. The present model had achieved better MAE of 0.248 as compared to traditional models like BPNN and ANN. Its architecture effectively recognizes intricate temporal dependencies characteristic of financial data, whereas the pre-processing procedures (normalization, noise reduction, outlier elimination) improved both quality and predictive potential of data significantly. Comparative analysis confirmed demonstrates, when compared to the models under investigation, the ARIMA-LSTM model produces the most accurate and consistent findings and, therefore, is a promising candidate for stock price forecasting. However, it can be compromised by extreme market fluctuations or use with other sectors of the market, without training.

Future work will be aimed at improving the model's forecasting ability through the introduction of attention mechanisms, ensemble learning strategies, and more financial indicators, to better reflect market dynamics. Real time data streams will become an important step forward in enhancing responsiveness and validity in real trading environments the adaptation of the model for cross-market and multi-sector use is also likely to contribute to the creation of a more generalizable model that will be more practical and trustworthy when it comes to practical financial analytics and investment strategies.

References

[1] M. Mallikarjuna and R. P. Rao, "Evaluation of forecasting methods from selected stock market returns," *Financ. Innov.*, 2019, doi: 10.1186/s40854-019-0157-x.

[2] S. Ben Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa, "A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 7067–7083, 2012, doi: https://doi.org/10.1016/j.eswa.2012.01.039.

[3] M. C. Angadi and A. P. Kulkarni, "Time series data analysis for stock market prediction using data mining techniques with R," *Int. J. Adv. Res. Comput. Sci.*, vol. 6, no. 6, pp. 104–108, 2015, doi: 10.13140/RG.2.1.1347.3360.

[4] G. Bontempi, S. Ben Taieb, and Y.-A. Le Borgne, "Machine Learning Strategies for Time Series Forecasting," in *Business Intelligence: Second European Summer School, eBISS 2012, Brussels, Belgium, July 15-21, 2012, Tutorial Lectures*, M.-A. Aufaure and E. Zimányi, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 62–77. doi: 10.1007/978-3-642-36318-4_3.

[5] V. Kolluri, "A Comprehensive Analysis On Explainable And Ethical Machine: Demystifying Advances In Artificial Intelligence," *TIJER - Int. Res. Journals*, vol. 2, no. 7, pp. 2349–9249, 2015.

[6] S. Singamsetty, "Fuzzy-Optimized Lightweight Cyber-Attack Detection For Secure Edge-Based Iot Network.," *J. Crit. Rev.*, vol. 6, no. 07, pp. 1028–1033, 2019, doi: 10.53555/jcr.v6:i7.13156.

[7] J.-F. Chen, W.-L. Chen, C.-P. Huang, S.-H. Huang, and A.-P. Chen, "Financial Time-Series Data Analysis Using Deep Convolutional Neural Networks," in *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, 2016, pp. 87–92. doi: 10.1109/CCBD.2016.027.

[8] J. F. Chen, W. L. Chen, C. P. Huang, S. H. Huang, and A. P. Chen, “Financial time-series data analysis using deep convolutional neural networks,” in *Proceedings - 2016 7th International Conference on Cloud Computing and Big Data, CCBD 2016*, 2017. doi: 10.1109/CCBD.2016.027.
[9] R. J. Hyndman and Y. Khandakar, “Automatic Time Series Forecasting: The forecast Package for R,” *J. Stat. Softw.*, vol. 27, no. 3, p. 1, 2008.
[10] C. Ma, Y. Ning, H. Jin, and J. Wu, “The hybrid dynamic stock forecasting model based on ANN and SVR,” in *Proceedings - 2019 International Conference on Intelligent Computing, Automation and Systems, ICICAS 2019*, 2019. doi: 10.1109/ICICAS48597.2019.00155.
[11] Y. Liu and X. Liu, “A Trend-based Stock Index Forecasting Model with Gated Recurrent Neural Network,” in *Proceedings of the 2018 IEEE International Conference on Progress in Informatics and Computing, PIC 2018*, 2018. doi: 10.1109/PIC.2018.8706267.
[12] G. Tsang, J. Deng, and X. Xie, “Recurrent Neural Networks for Financial Time-Series Modelling,” in *Proceedings - International Conference on Pattern Recognition*, 2018. doi: 10.1109/ICPR.2018.8545666.
[13] M. S. Raimundo and J. Okamoto, “SVR-wavelet adaptive model for forecasting financial time series,” in *2018 International Conference on Information and Computer Technologies, ICICT 2018*, 2018. doi: 10.1109/INFOCT.2018.8356851.
[14] K. A. Althelaya, E. S. M. El-Alfy, and S. Mohammed, “Stock Market Forecast Using Multivariate Analysis with Bidirectional and Stacked (LSTM, GRU),” in *21st Saudi Computer Society National Computer Conference, NCC 2018*, 2018. doi: 10.1109/NCG.2018.8593076.
[15] E. Beyaz, F. Tekiner, X. Zeng, and J. Keane, “Stock Price Forecasting Incorporating Market State,” in *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, IEEE, Jun. 2018, pp. 1614–1619. doi: 10.1109/HPCC/SmartCity/DSS.2018.00263.
[16] A. Bakhach, E. P. K. Tsang, and H. Jalalian, “Forecasting directional changes in the FX markets,” in *2016 IEEE Symposium Series on Computational Intelligence, SSCI 2016*, 2017. doi: 10.1109/SSCI.2016.7850020.
[17] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, “Stock Price Prediction Using the ARIMA Model,” in *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, 2014, pp. 106–112. doi: 10.1109/UKSim.2014.67.
[18] J. Zhang, Y. Zhu, X. Zhang, M. Ye, and J. Yang, “Developing a Long Short-Term Memory (LSTM) based model for predicting water table depth in agricultural areas,” *J. Hydrol.*, vol. 561, pp. 918–929, Jun. 2018, doi: 10.1016/j.jhydrol.2018.04.065.
[19] L. J. Cao and F. E. H. Tay, “Feature selection for support vector machines in financial time series forecasting,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1983, pp. 268–273, 2000, doi: 10.1007/3-540-44491-2_38.
[20] M. Khashei and M. Bijari, “An artificial neural network (p, d, q) model for timeseries forecasting,” *Expert Syst. Appl.*, vol. 37, no. 1, pp. 479–489, 2010, doi: 10.1016/j.eswa.2009.05.044.
[21] Chinta, P. C. R., & Karaka, L. M. (2020). Agentic AI and Reinforcement Learning: Towards More Autonomous and Adaptive AI Systems.
[22] Kalla, D., Kuraku, D. S., & Samaah, F. (2021). Enhancing cyber security by predicting malwares using supervised machine learning models. *International Journal of Computing and Artificial Intelligence*, *2*(2), 55-62.
[23] Katari, A., & Kalla, D. (2021). Cost Optimization in Cloud-Based Financial Data Lakes: Techniques and Case Studies. *ESP Journal of Engineering & Technology Advancements (ESP-JETA)*, *1*(1), 150-157.
[24] Kalla, D., Smith, N., Samaah, F., & Polimetla, K. (2021). Facial Emotion and Sentiment Detection Using Convolutional Neural Network. *Indian Journal of Artificial Intelligence Research (INDJAIR)*, *1*(1), 1-13.
[25] Routhu, K., Bodepudi, V., Jha, K. M., & Chinta, P. C. R. (2020). A Deep Learning Architectures for Enhancing Cyber Security Protocols in Big Data Integrated ERP Systems. *Available at SSRN 5102662*.
[26] Kuraku, S., & Kalla, D. (2020). Emotet malware—a banking credentials stealer. *Iosr J. Comput. Eng*, *22*, 31-41.
[27] Kalla, D., & Samiuddin, V. (2020). Chatbot for medical treatment using NLTK Lib. *IOSR J. Comput. Eng*, *22*, 12.
[28] Routhu, K., & Jha, K. M. (2021). Leveraging Big Data Analytics and Machine Learning Techniques for Sentiment Analysis of Amazon Product Reviews in Business Insights. *Available at SSRN 5106490*.
[29] Chinta, P. C. R., & Katnapally, N. (2021). Neural Network-Based Risk Assessment for Cybersecurity in Big Data-Oriented ERP Infrastructures. *Neural Network-Based Risk Assessment for Cybersecurity in Big Data-Oriented ERP Infrastructures*.
[30] Karaka, L. M. (2021). Optimising Product Enhancements Strategic Approaches to Managing Complexity. *Available at SSRN 5147875*.
[31] Boppana, S. B., Moore, C. S., Bodepudi, V., Jha, K. M., Maka, S. R., & Sadaram, G. AI And ML Applications In Big Data Analytics: Transforming ERP Security Models For Modern Enterprises.