Revolutionizing Stock Market Intelligence: A Deep Dive into Machine Learning for IT Sector Price Predictions

Parth Mishra*, and Gaurav Parashar§
Data Science & Deep Learning Lab
KIET group of Institutions
Delhi-NCR, Ghaziabad, India
Email: *pm74parth@gmail.com,
§https://orcid.org/0000-0003-4869-1819

Vaibhav Chaudhary† and Ishant Sehgal‡
KIET group of Institutions
Delhi-NCR, Ghaziabad, India
Email: †vaibhav.2024cse1073@kiet.edu,
‡ishant.2024cse1174@kiet.edu,

***Abstract*—This study challenges the efficient market hypothesis by presenting a diverse approach to stock price prediction, incorporating statistical, machine learning, and deep learning models with daily stock data. Demonstrating effectiveness in capturing volatile patterns, the framework integrates artificial intelligence with traditional time-series analysis methods, including autoregressive integrated moving average (ARIMA), long short-term memory (LSTM), and Facebook Prophet. A noteworthy advancement is the exceptional forecasting accuracy of the predictive toolkit, enabling users to create and share personalized stock portfolios based on advanced, unique forecasts. In the realm of business intelligence, the framework employs a novel machine-learning approach for predicting stock price movements in the information technology sector. Emphasizing the significance of accurate stock price prediction, the paper advocates for continuous exploration of diverse machine learning techniques. In conclusion, the review highlights the synergy between traditional time-series analysis and modern machine-learning techniques with substantial potential for navigating the complexities of financial markets and empowering users in making well-informed investment decisions.**

***Index Terms*—Keywords: Agglomerative framework, Time-series analysis, Stock Market, CryptoCurrency, Deep Learning, Machine Learning**

I. Introduction

The exponential growth of stocks in recent years has drawn significant attention from individual enthusiasts and institutional players alike. Microsoft stock's meteoric rise, from around 63.62 dollars at the close of 2016 to over 231.65 dollars in mid-2020, has underscored the need for accurate prediction models tailored to predict stock prices. This paper explores the adaptability of existing stock analysis techniques focusing primarily on Microsoft stocks. While traditional markets rely on well-established mathematical models, the cryptocurrency and stock landscape lacks a robust framework for predicting the volatile price movements of digital assets. Our research investigates whether machine learning and deep learning models, proven effective in traditional financial markets, can be seamlessly applied to stock data. The allure of stocks, with their transparency, anonymity, and resistance to fraud, has heightened investor interest, leading them to seek advanced tools, including Machine Learning and Deep Learning models, to inform their trading decisions. This paper addresses the question of whether Machine Learning and Deep Learning models can be applied effectively to stocks and cryptocurrency data, with objectives including the identification of pertinent features, assessment of ML algorithms' suitability, and determination of the optimal approach for prediction.

Our methodology involves the application of linear regression and recurrent neural networks (RNNs) with LSTM cells to historical bitcoin price data. Additionally, we extend our analysis to include time series models—ARIMA, and the Facebook Prophet model commonly used in financial forecasting. Results and analyses from our experiments on historical Microsoft Stock price data are presented, comparing linear regression, RNNs with LSTM cells, ARIMA, and Facebook Prophet. Evaluation metrics such as accuracy, precision, and recall provide a nuanced understanding of each model's strengths and weaknesses [1]. By delving into the symbiotic relationship between conventional financial frameworks and state-of-the-art machine learning and deep learning methodologies, while integrating time series models, our study unveils fresh perspectives for forecasting stock prices. The potential extension of these models to autonomous agents capable of trading based on real-time data, news sentiment analysis, and historical price information adds sophistication to the evolving landscape of stocks and cryptocurrency investments.

To contextualize our study, we provide a comprehensive review of existing literature, highlighting the transformative power of ML in financial markets and its potential application to the stocks and cryptocurrency domain. Acknowledging the limitations of current approaches, we conclude by outlining potential directions for future research. This exploration seeks to contribute valuable insights into the dynamic landscape of

979-8-3503-0775-7/24/$31.00 ©2024 IEEE

stock investment and the evolving role of ML and DL in predicting price movements.

II. Related Work

In the extensive landscape of research on stock price prediction, three main strands of literature have emerged, each employing distinct methodologies. The first strand utilizes straightforward cross-sectional regression analysis. Time series models such as ARIMA are used in the second. Natural language processing, deep learning, and machine learning are all used in the third strand. Noteworthy contributions include Mehtab and Sen's [2] Employing machine learning techniques alongside LSTM-based deep learning networks, researchers achieved precise forecasts of NIFTY 50 stock prices. Their integration of sentiment analysis from Twitter resulted in enhanced predictive capabilities. Additionally, they presented a series of regression models utilizing convolutional neural networks (CNNs), showcasing remarkable accuracy and robustness in forecasting multivariate financial time series data. A study by Nabipour [3] uses a combination of three deep learning models (ANN, RNN, and LSTM) and six tree-based models (Decision Tree, Bagging, Random Forest, AdaBoost, Gradient Boosting, and XGBoost) to predict stock prices on the Tehran Stock Exchange. Forecasts were made for one, two, five, ten, fifteen, twenty, and thirty days in advance. Evaluation of the models utilized MAPE, MAE, RRMSE, and MSE as assessment metrics.. [3]

LSTM emerged as the most accurate with the least error across all examined stock markets. Nonetheless, creating a resilient and highly precise predictive model for stock prices continues to pose a significant challenge, given the inherent randomness and volatility of financial time series. This research addresses this challenge by leveraging the feature extraction and learning capabilities inherent in deep learning models, harnessing their architectural diversity to achieve robustness and accuracy in predicting stock prices for detailed time series data. [4]. The literature employs a diverse range of techniques, including chaos theory-based neural networks, linear wavelet neural networks, and even ensemble models combining SVM and ANN, highlighting the breadth of approaches explored. By analyzing the strengths and weaknesses of various methods, such as ARIMA-based forecasting, deep learning frameworks, and multiple regression with systematic predictor selection, we can contribute to the evolving discourse on stock price prediction.

III. Financial Instruments

Financial instruments encompass agreements concerning tradable assets, including stocks, bonds, bills, currencies, swaps, futures, and options. These contracts confer the right to assert ownership over an entity's assets or to hold partial or complete ownership of the entity itself. [5]. They represent income claims generated by real assets, such as the sale of cocoa beans, property leasing, or service provision.

Equity assets, commonly known as shares, are issued by public companies to signify partial ownership. Individuals or groups, referred to as stockholders or shareholders, attain the status of company owners. In instances where a company seeks to expand its operations and requires additional capital, it may issue new shares, subject to approval by existing shareholders, as the issuance dilutes their ownership. The value of the stock tends to increase with the success of the company, making the performance of stock investments contingent on both the company's success and its real assets [5].

A stock market, also referred to as an equity market, serves as a public arena where traders engage in the buying and selling of a company's shares and related derivatives, either through electronic platforms or traditional exchanges. Generally, financial instruments are exchanged within the broader capital market, consisting of a primary market, where securities are initially issued, such as through initial public offerings (IPOs), and a secondary market, where trading occurs among investors. Notable examples of stock markets include the New York Stock Exchange, London Stock Exchange, Japan Exchange Group, Shanghai Stock Exchange, and NASDAQ. [5].

A stock index serves as a significant indicator reflecting the collective performance of a group of stock prices, calculated based on the prices of specific stocks. Fluctuations in the index reflect the overall performance of the listed stocks. Importantly, a stock index represents the weighted average market value of multiple companies relative to a designated base trading day. Examples of stock indices include the Financial Times Stock Exchange 100 Index (FTSE 100). [5]. Stock trading presents a substantial challenge for investors owing to the influence of a myriad of complex factors. Economic conditions, local and international politics, as well as social dynamics, all weigh heavily in investment decisions. Traders employ various strategies, such as day trading, position trading, swing trading, and scalping, to navigate the buying and selling of company shares. [5].

IV. Methodology

The notion of artificial neural networks was initially introduced by McCulloch and Pitts [6], gaining popularity among researchers for modeling nonlinear processes experimentally. Among the various types of neural networks, the perceptron stands out, available in both single- and multi-layer configurations. McClelland, Rumelhart, and Hinton further advanced this field by successfully creating a multilayer feedforward neural network (MLP) through the implementation of the back-propagation (BP) algorithm. In the financial and investment sectors, neural networks find extensive application, including tasks such as bankruptcy prediction, decision-making, and financial planning, refer to figure 1.

The dataset for this study was obtained using the Yahoo Finance library, a robust tool designed for extracting financial data from Yahoo Finance. Yahoo Finance, an integral component of the Yahoo! network, is a significant resource for financial information and analysis which offers a wide range of financial information including stock quotes, cryptocurrencies, press releases, financial data, and educational articles. Data

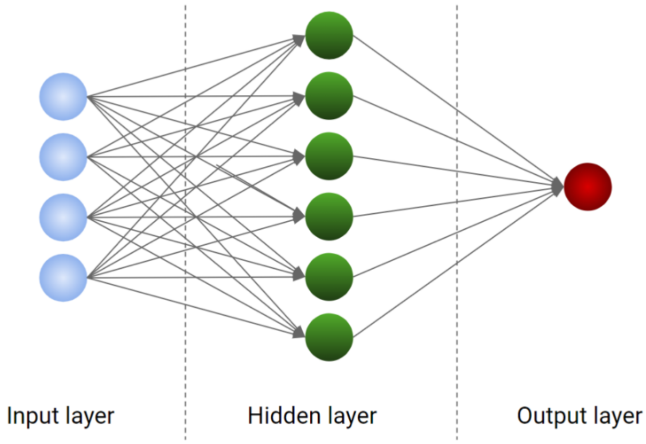

Fig. 1. Neural Network [7]

for this investigation was systematically gathered using Yahoo Finance, with data collection occurring at one-day intervals. This method facilitated the acquisition of daily information on metrics such as opening price, closing price, and adjusted closing price at regular intervals. refer to figure 2.

Date	Open	High	Low	Close	Adj Close	Volume
2014-12-01	47.880001	48.779999	47.709999	48.619999	42.061855	31191600
2014-12-02	48.840000	49.049999	48.200001	48.459999	41.923435	25773500
2014-12-03	48.439999	48.500000	47.810001	48.080002	41.594692	23534800
2014-12-04	48.389999	49.060001	48.200001	48.840000	42.252190	30320400
2014-12-05	48.820000	48.970001	48.380001	48.419998	41.888828	27313400

Fig. 2. Yahoo Finance Data

Combining this one-day gap data with the particular attributes that were retrieved. In terms of capturing long-term dependencies, the LSTM model outperforms other architectures, like the RNN. During the experimentation phase, the LSTM showed fewer negative effects from choosing a longer temporal window. Using autocorrelation lag as a guide, the window size was determined using a methodology akin to that of the RNN.

The preliminary model was configured with a window size of 60 days, incorporating two LSTM layers with fifty hidden nodes each, followed by a dense layer containing 25 nodes, and concluded with an output layer. Training of the model utilized the Adam optimizer and mean squared error loss function for a total of 10 epochs. Hyperparameter tuning was performed using Bayesian optimization with the hyperas library. The optimal dropout rate and optimizer (RMSprop) were identified. The LSTM model demonstrated convergence in 50–100 epochs with early stopping. Batch size had a notable impact on execution time but not on accuracy. A window size of 100 days was found to be optimal for capturing nonlinear relationships in the time series. Two LSTM layers with 20 hidden nodes each struck a balance between computational efficiency and model effectiveness. The model retained the default activation functions (tanh and sigmoid) for LSTM layers, deemed suitable for capturing temporal dependencies in sequential data. To summarize, meticulous attention to the window size, layer configuration, and hyperparameters resulted in an optimized LSTM model tailored specifically to the distinctive traits of the provided time series data. Given that the LSTM inherently employs a sequence of tanh and sigmoid activation functions for different gates within the cell, there was no alteration made to the activation functions in the LSTM model. [8]. For the LSTM models, convergence was usually reached in 50–100 epochs, with early stopping. It was observed, in line with the RNN observations, that batch size significantly affected execution time rather than accuracy. The somewhat small size of the could be the cause of this effect. The careful consideration of these factors contributes to the optimized performance of the LSTM model in handling the specific characteristics of the time series data.

The ARIMA (p, d, q) model stands out in time series analysis for its effectiveness in generating short-term forecasts, consistently surpassing complex structural models. When dealing with non-stationary time series, adjustments are necessary to transform them into stationary ones. The determination of ARIMA model orders involves aligning observed patterns in sample autocorrelation functions (ACF) and sample partial autocorrelation functions (PACF) with the theoretical patterns of known models, thereby identifying the appropriate orders. [9]. Differences can be made to convert non-stationary time series with short-term trends into stationary time series. ARIMA (1, 0, 1) is the model's most basic equation. Because it is straightforward, effective, and efficient, ARIMA (1, 0, 1) is a good model for non-stationary time series with short-term trends. While the MA(1) term attenuates the effects of noise, the AR(1) term captures the trend in the data. Based on decomposable models, Prophet is an open-source library. It enables us to use straightforward parameters to make accurate time-series predictions. The fact that it supports taking into account the influence of custom, seasonality, and holidays is also significant. [10]. Prophet is an effective additive-based model for time series forecasting because it was designed to capture non-linear trends, a variety of seasonal patterns (annual, weekly, and daily), and the effect of holidays on data. Prophet, created by Facebook's core data science team, works especially well with time series datasets that have significant seasonal fluctuations and a large amount of historical data spanning several seasons.

The Prophet model's resilience to missing data and its ability to adapt well to outliers and changes in trends are two of its main advantages. Because of this, it is a flexible technique that may be used for predicting in real-world situations where anomalies and poor data quality are frequent problems. Crafted to capture non-linear trends, diverse seasonal patterns (annual, weekly, and daily), and the impact of holidays on data, Prophet emerges as a robust additive-based model for time series forecasting. Developed by Facebook's core data science team, Prophet excels particularly with time series datasets characterized by pronounced seasonal fluctuations and extensive historical data across multiple seasons. [11]. The data frame that contains the input data for the Prophet model has the columns "ds" and "y" designated as required fields. The date stamp can be found in the 'ds' column, and it must be prepared in compliance with the Pandas Library guidelines. This format allows for the inclusion of dates (such as YYYY-MM-DD) or timestamps (such as HH:MM: SS). The 'y' column, which must contain numeric values, represents the goal measurement or attribute that the model aims to forecast refer to figure 3

	ds	y
Date		
2014-12-01	2014-12-01	42.061859
2014-12-02	2014-12-02	41.923431
2014-12-03	2014-12-03	41.594707
2014-12-04	2014-12-04	42.252186
2014-12-05	2014-12-05	41.888840

Fig. 3. Prophet data-table

Prophet offers a user-friendly and potent solution for time series forecasting, specifically tailored for datasets exhibiting pronounced seasonal effects and encompassing multiple seasons of historical data.

As mentioned earlier, Prophet is an additive model composed of the following components:

y(t) = g(t) + s(t) + h(t) + ϵ g(t) models' trend, This component describes long-term trends, whether they are upward or downward, within the data. [12].

s(t) models' seasonality with the Fourier series, This component characterizes the impact of seasonal factors on the data, such as variations based on the time of the year.

h(t) models the This component accounts for the effects of significant occasions or large events that influence the time series of business data.

ϵ represents an irreducible error term

$y_t = \hat{y}_t + \hat{s}_t + \hat{h}_t + \epsilon_t$

(Decomposition of observed value)

$\hat{y}_t = g(t)$

(Fitted trend value)

$g(t) = \alpha + \beta t + \sum_{i=1}^{m} \gamma_i \sin(\frac{2\pi i}{s}) + \sum_{i=1}^{m} \delta_i \cos(\frac{2\pi i}{s})$

(Trend function with Fourier terms)

$\hat{s}_t = \sum_{i=1}^{n} \delta_i D_i(t)$

(Seasonal component with dummy variables)

$\hat{h}_t = \sum_{i=1}^{k} \gamma_i H_i(t)$

(Holiday component with specific weights)

$\epsilon_t \sim N(0, \sigma^2)$

(Error term follows normal distribution)

V. Evaluation

To perform data preprocessing for LSTM download historical stock data using Yahoo Finance for a specified period from the year 2014 select the target stock symbol 'MSFT' for Microsoft stock.

Extract the pertinent features from the downloaded data, specifically emphasizing the daily closing prices. Typically, normalization is required, especially for neural networks employing squashing functions like tanh and sigmoid, which struggle with inputs significantly beyond the (-1,1) range. Even if a network doesn't have this constraint, normalization is crucial when learning from multiple series with varying amplitudes. Normalize the closing prices using Min-Max scaling to ensure all input features are on a comparable scale, typically ranging between 0 and 1. This step is crucial for neural networks, including LSTMs, to converge efficiently during training [13].

Generate input samples for the LSTM model by creating sequences of closing prices. This entails specifying a sequence length (e.g., 50) and generating sequences along with their corresponding target values. For example, if the sequence length is set to 10, each training sample's input consists of a sequence comprising the preceding 10 closing prices, while the target value is the subsequent closing price. Reshaping for LSTM Input: Reshape the input sequences to a 3D array with dimensions (samples, sequence length, features). This is necessary to match the input requirements of the LSTM model. In the context of input data for the LSTM model, the "samples" dimension denotes the quantity of training samples, while "sequence-length" indicates the length of each input sequence. The dimension "features" signifies the number of features for each time step. In this case, since only the closing prices are utilized, the "features" dimension is set to 1.

Data Preprocessing for ARIMA

Retrieve historical stock data from the Yahoo Finance API for the 'MSFT' symbol within a specified period starting from the year 2014. When constructing dependable models, it is crucial to ensure that the model assumptions remain intact and that the input data is accurately preprocessed. Standard preprocessing procedures such as data cleaning to eliminate noise and inconsistencies, handling missing values, data reduction by removing redundant features or employing methods like clustering, and normalization and transformation of data into a suitable range are typically employed in a standard data mining process to ensure data quality. However, each statistical or machine learning-based model usually comes with specific assumptions. It is essential to consider these assumptions to determine a limited set of preprocessing tasks applicable to the input data. [14]. Isolate the daily closing prices from the acquired dataset as they form the primary focus for analysis. Utilize an automated procedure like auto-arima to identify the optimal order parameters for the ARIMA model. This entails determining the p, d, and q values, which correspond to the autoregressive, differencing, and moving average components, respectively.

Fit the model using the previously identified order parameters to the time series data of closing prices. This step involves configuring the model to capture the seasonality and trends within the data. Leverage the trained model to generate forecasts for forthcoming time steps. The model utilizes the learned patterns and relationships to provide predictions for future values in the time series.

Data Preprocessing for Facebook Prophet

Utilize the Yahoo Finance API, a widely used financial data source, to retrieve historical 'MSFT' (Microsoft stock) data. Specify the desired period from the year 2014 for analysis. After downloading the data, organize it by resetting the index and selecting only the 'Date' and 'Close' columns, streamlining the dataset for price prediction. dapt the column names to align with the input requirements of the Prophet model, specifying 'ds' for the date column and 'y' for the closing price column. This adjustment ensures compatibility with the model's expectations and facilitates smooth integration into the forecasting framework. Proceed to set up the Prophet model, a forecasting tool developed by Facebook. Configure the model parameters, including seasonality settings, holidays, and other relevant options. Proper initialization of the model is paramount for generating accurate and meaningful predictions.

Proceed to train the initialized Prophet model with the formatted historical stock data. During this stage, the model assimilates insights from past patterns and trends within the data, adjusting internal parameters to effectively capture the underlying structure of the time series data. Once trained, apply the Prophet model to produce forecasts for future time steps. This predictive capability allows the model to project how the stock price might evolve over a specified future period, leveraging insights gained during the model fitting phase. The dataset underwent division into two distinct parts: a training set inclusive of data spanning from 2014 to 2021, and a testing set encapsulating the entirety of the year 2022. This segregation enabled the training of models on historical information, thereby facilitating the evaluation of their predictive capabilities on unseen future data.

Each model was trained on the historical dataset, learning patterns, trends, and seasonality present in the data. The training process involved adjusting model parameters to minimize the difference between predicted values and actual observations in the training set. Following training, each model was employed on the testing set (2022 data) to produce forecasts for the future period. To assess the performance of the models, the predicted closing prices were subjected to Root Mean Square Error (RMSE) analysis, enabling the determination of the final minimized errors in the predicted prices. [15]

VI. Result and Conclusion

n our experimental setup, we utilized the LSTM, ARIMA, and Prophet models. Through our analysis, we observed that the LSTM model outperforms both the ARIMA and Prophet models in terms of predictive accuracy and performance, refer to figure 4 and figure 5. The code can be found at GitHub repository [1]

Based on its better performance measures, especially its minimal RMSE (Root Mean Squared) values refer to figure 7, the LSTM model is the best model for predicting MSFT (Microsoft stock) prices refer to figure 6. Remarkably, the LSTM model attains accuracy rates higher than 85 percent, demonstrating its resilient capacity to steer through the erratic swings in stock values, particularly during the difficult pandemic phase. Incorporating key financial variables into the model has the potential to enhance prediction accuracy in the future. These indicators, reflecting market and economic trends, could augment the model's predictive capabilities by capturing nuanced factors influencing fluctuations in stock and cryptocurrency prices.

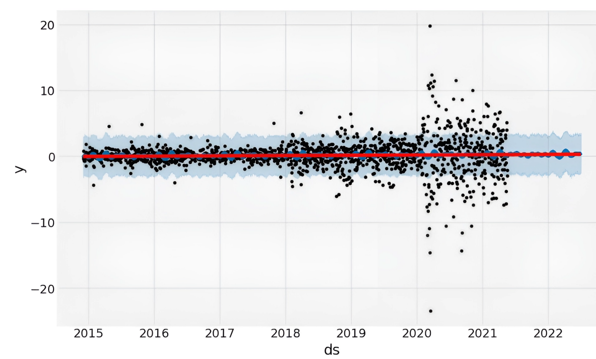

Fig. 4. Prophet

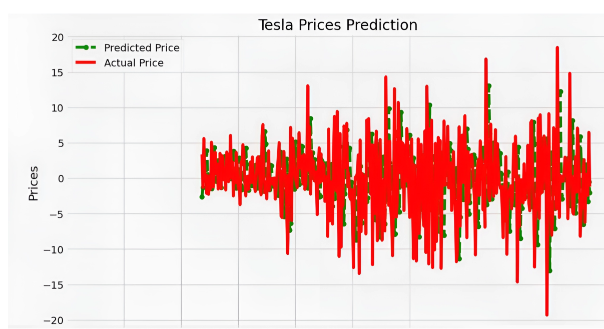

Fig. 5. ARIMA

The recommended strategies to further improve the accuracy and responsiveness of this predictive model to changing market conditions are the integration of sentiment analysis and financial indicators as we map out its future trajectory. In this study, predictions were solely based on price data. However, incorporating more influential features such as daily

[1]Github Repositoryhttps://github.com/praxton74/FinalYearProject

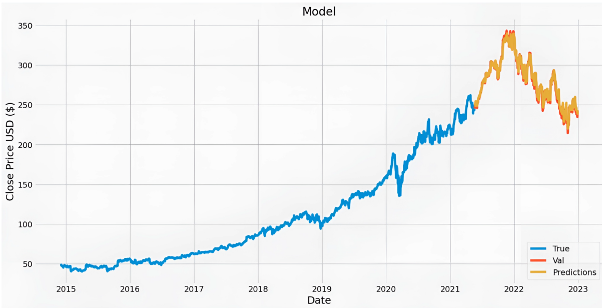

Fig. 6. LSTM

S.No	Evaluated Result	
	Model Name	**RMSE**
1	LSTM	1.204
2	ARIMA	6.615
3	Prophet	4.527

Fig. 7. Final Table

volume, volatility, fundamental ratios, etc., could enhance model performance. Future research could extend the model's applicability to predict less volatile market indices, such as Nasdaq. Additionally, exploring sentiments from news articles and integrating them into the LSTM model alongside other features could present an intriguing research avenue. Lastly, employing more sophisticated optimization techniques could prove beneficial when incorporating additional attributes into the model. [16]. Employing these four models was optimal as it provides a comprehensive examination of time series forecasting tools. Utilizing these models across three distinct asset classes allows us to observe their performance in varied environments. Moving forward, individuals can tailor their models based on the insights garnered from this report, selecting parameters that align best with their requirements. [17].

VII. Acknowledgement

We want to express our sincere gratitude to the Data Science & Deep Learning (DSDL) Lab, Computer Science & Engineering department, KIET Group of Institutions, Ghaziabad for their unwavering support and collaboration during this research. We also appreciate the access to the PARAM Shavak Supercomputing facility at DSDL Lab[2].

References

[1] I.-C. Lolea, I.-R. Petrariu, and A. Giurgiu, "Arima vs. machine learning in terms of equity market forecasting," *The Annals of the University of Oradea. Economic Sciences*, vol. 30, no. 2nd, 2021.

[2] S. Mehtab and J. Sen, "Stock price prediction using cnn and lstm-based deep learning models," in *2020 International Conference on Decision Aid Sciences and Application (DASA)*, pp. 447–453, IEEE, 2020.

[3] M. Nabipour, P. Nayyeri, H. Jabani, S. Shahab, and A. Mosavi, "Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis," *IEEE Access*, vol. 8, pp. 150199–150212, 2020.

[4] J. Sen and S. Mehtab, "Accurate stock price forecasting using robust and optimized deep learning models," in *2021 International Conference on Intelligent Technologies (CONIT)*, pp. 1–9, IEEE, 2021.

[5] M. Obthong, N. Tantisantiwong, W. Jeamwatthanachai, and G. Wills, "A survey on machine learning for stock price prediction: algorithms and techniques," in *2nd International Conference on Finance, Economics, Management and IT Business (05/05/20 - 06/05/20)*, pp. 63–71, May 2020. Publisher Copyright: Copyright © 2020 by SCITEPRESS - Science and Technology Publications, Lda. All rights reserved.

[6] M. Nikou, G. Mansourfar, and J. Bagherzadeh, "Stock price prediction using deep learning algorithm and its comparison with machine learning algorithms," *Intelligent Systems in Accounting, Finance and Management*, vol. 26, no. 4, pp. 164–174, 2019.

[7] B. G. MALATO, *How many neurons for a neural network*. YOUR DATA TEACHER Online courses and lessons about data science, machine learning and artificial intelligence, 2014.

[8] S. McNally, J. Roche, and S. Caton, "Predicting the price of bitcoin using machine learning," in *2018 26th euromicro international conference on parallel, distributed and network-based processing (PDP)*, pp. 339–343, IEEE, 2018.

[9] S. Siami-Namini and A. S. Namin, "Forecasting economics and financial time series: Arima vs. lstm," *arXiv preprint arXiv:1803.06386*, 2018.

[10] M. Navratil and A. Kolkova, "Decomposition and forecasting time series in the business economy using prophet forecasting model," *Central European Business Review*, 2019.

[11] Z. Z. Oo and P. Sabai, "Time series prediction based on facebook prophet: A case study, temperature forecasting in myintkyina," *International Journal of Applied Mathematics Electronics and Computers*, 2020.

[12] A. Hasnain, Y. Sheng, M. Z. Hashmi, U. A. Bhatti, A. Hussain, M. Hameed, S. Marjan, S. U. Bazai, M. A. Hossain, M. Sahabuddin, *et al.*, "Time series analysis and forecasting of air pollutants based on prophet forecasting model in jiangsu province, china," *Frontiers in Environmental Science*, vol. 10, p. 945628, 2022.

[13] S. Smyl and K. Kuber, "Data preprocessing and augmentation for multiple short time series forecasting with recurrent neural networks," *arXiv preprint*, 2016.

[14] A. H. Adineh, Z. Narimani, and S. C. Satapathy, "Importance of data preprocessing in time series prediction using sarima: A case study," *International Journal of Knowledge-Based and Intelligent Engineering Systems*, vol. 24, no. 4, pp. 331–342, 2020.

[15] M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, "Stock closing price prediction using machine learning techniques," *Procedia computer science*, 2020.

[16] D. Shah, W. Campbell, and F. H. Zulkernine, "A comparative study of lstm and dnn for stock market forecasting," in *2018 IEEE international conference on big data (big data)*, pp. 4148–4155, IEEE, 2018.

[17] R. Verma, J. Sharma, and S. Jindal, "Time series forecasting using machine learning," in *Advances in Computing and Data Sciences: 4th International Conference, ICACDS 2020, Valletta, Malta, April 24–25, 2020, Revised Selected Papers 4*, pp. 372–381, Springer, 2020.

[2] https://www.kiet.edu/home/department_wise_club_detail/NQ==/NA==/student_club