Time Series Prediction: Comparative Study of ML Models in the Stock Market

Gaurav Goverdhan
Vellore Institute of Technology University
Sarthak Khare
Vellore Institute of Technology University
Manoov R (manoov.r@vit.ac.in)
Vellore Institute of Technology University

Research Article

Keywords: ARIMA, XGBoost, LSTM, Time Series Forecasting

Posted Date: October 24th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2183122/v1

Time Series Prediction: Comparative Study of ML Models in the Stock Market

Gaurav Goverdhan[1†], Sarthak Khare[1†] and Manoov R[1*]

[1]School of Computer Science and Engineering, Vellore Institute of Technology, Tiruvalam Rd, Vellore, 632014, Tamil Nadu, India.

*Corresponding author(s). E-mail(s): manoov.r@vit.ac.in;
Contributing authors: gaurav.goverdhan2019@vitstudent.ac.in;
sarthak.khare2019@vitstudent.ac.in;
†These authors contributed equally to this work.

Abstract

Analyzing the Stock Market is a perpetual process and hard to grasp, especially for newcomers looking to invest in the market. This paper will be useful for novice investors to learn to invest in the stock market based on various factors that dictate prices. The paper's target is to create a program that analyses previous stock data of companies. This also includes identifying factors affecting the share market. We generate the patterns from large data sets of data of the stock market and predict an approximate value of share price. The stock market can have a significant impact on individuals and the economy as a whole. Therefore, accurately predicting stock trends can minimize loss risk and maximize profit. The primary objective of this paper is to generate an approximate forecasting output and a general idea of future values by generating a pattern from historical data. The scope of this paper extends to model a suggestion tool based on ARIMA, XGBoost, and LSTM and give critical insights into their relative advantages and disadvantages, finally determining the best tool for forecasting the trend or even the future. The trend chart will adequately guide all prospective investors. We have analyzed data from the stock market collected from Yahoo Finance. The performance of the models was evaluated using metrics such as MSE, MAE, RMSE, and MAPE. During our trials, we found XGBoost to perform the best with an accuracy of 98.92%.

Keywords: ARIMA, XGBoost, LSTM, Time Series Forecasting

1 Introduction

The work done in time series forecasting in the past is analyzed to understand the progress already made, expand upon current knowledge, and then use some of them as a baseline to show how our models outperform theirs, in addition to comparing the three models themselves.

In [1], the authors propose a hybrid method using HAVOK analysis to predict chaotic time series. The model is evaluated on Lorenz, Mackey-Glass, and sunspot time series. The model performs the best with the Mackey-Glass time series giving an RMSE value of 9.92×10^{-6}. The model requires tuning with particular optimization methods which are not explicitly suggested, making it impractical to implement.

In [2], the authors predict the NIFTY50 stocks of NSE (National Stock Exchange) from 2014-2020 using the models - logistic regression, SVM, and perceptron model. Closing prices of stocks are predicted based on the previous 3 days. SVM with a linear kernel performs the best.

In [3], the authors propose a hybrid method for time series forecasting using LR/ARIMA (Autoregressive Integrated Moving Average) for the linear part of the time series and a deep belief network for the non-linear part of the time series. ARIMA proves to be better with non-stationary time series. The model is evaluated on the Mackey-Glass time series, sunspot time series, Individual Household Electric Power Consumption (IHEPC), and electricity load demand data sets from the Australian Energy Market Operator (AEMO). The model performed best with the IHEPC time series with an MSE (Mean Squared Error) of 7.47×10^{-8}.

In [4], the authors compare the model - ARIMA, FBProphet, and XGBoost in their ability to predict the cryptocurrency markets. The models were evaluated on historical data of Bitcoin with RMSE, MAE, and R2 as the performance metrics. ARIMA performed best with an RMSE of 322.4.

In [5], the authors detect jumps in financial time series using models such as SVM and KNN, combined with LSTM. The models were evaluated on Dow Jones Industrial Average (DJI), S&P 500 Index (SPX), and many more. The computational complexity of the proposed model is high. The jump detection model is suitable for a tail risk protection trading strategy, instead of a momentum strategy.

In [6], the authors propose a multi-output least squares support vector regression (MLSSVR) model. The hyperparameters are fine-tuned using an accelerated particle swarm optimization (ASPO) algorithm though most of the parameters require self-tuning. The model is not favorable for long-term investments.

In [7], the authors of this paper compare ML models for time series prediction, such as kNN, SVM, and SARIMA. The evaluation is done with the ICMC-USP Time Series Prediction Repository. SARIMA shows the best MSE but is the most expensive and complex in terms of computational costs. The hyperparameters for SARIMA are estimated using the Box-Jenkins method.

In [8], the authors propose a hybrid model integrating an extreme learning machine (ELM) and discrete wavelet transform (DWT). The last transaction price of stocks of the following companies: INTC, HPQ, CSCO, VZ, MSFT, and IBM, from February 5th, 2010 between 9:30 and 16:00hrs are used to evaluate the model. Three models ensembled together (an extreme learning machine, a non-linear multi-scale neural-wavelet machine, and a linear predictor) receive the same input in parallel. The outputs of each of the three models are linearly combined into a single output neuron. The ELM, WE, and μ-XNW models achieve THEIL values of 0.84780, 0.80959, and 0.80636, which are very ideal time series prediction algorithms.

In [9], the authors review convolutional neural network (CNN), wavelet neural network (WNN), fuzzy neural network (FNN), Markov chain models, and long short-term memory (LSTM) to predict time series data of oil, wind energy - specifically chaotic time series. A detailed study of the various evaluation metrics used to judge the outcome of these models is presented which gives us some factors to consider while comparing models in our trials.

In [10], the authors forecast high-frequency financial time series by combining the traditional ARIMA model with the deep learning model. For machine learning and deep learning models that rely on big data, a lack of sample size will prevent the model from acquiring data characteristics, preventing it from capturing abnormal fluctuations and predicting future trends for timeseries. In the context of the stock market, this is not a concern.

In [11], the authors use the Prophet model to predict the future price of a stock, specifically the Santander Group. They find it to be useful to predict trends around months and days of the week It also handles seasonality well. Thus, it stands to be a model effective in forecasting time series data.

In [12], the theory of the modern portfolio is applied, which is the basic principle of diversification and the development of an effective frontier. Methods like linear regression, K means Clustering, and K nearest neighbor is evaluated. It is found that the application of several different models does not yield a linear relationship between different factors. The efficiency of results can be improved by limiting the number of models used.

In [13], the authors compare ARIMA and Prophet models using MAPE as an evaluation metric to compare error analysis results in three different periods. The model does not produce seasonal patterns which reduce the accuracy of the predicted value.

In [14], the primary objective of the authors is to determine the accuracy of a machine learning algorithm's predictions and the extent to which epochs improve the model. They build a model using RNN and LSTM models to predict future stock market values. The testing result confirms that their model is capable of tracing the evolution of opening prices for both assets, Google and Nike.

In [15], the authors compare SVR and LSTM based on the MAPE (Mean Absolute Percentage Error) values. The datasets in use are stock indexes such as NSE, BSE, NASDAQ, S&P 500, and Dow Jones Industrial Average. LSTM provides better prediction accuracy based on the authors' research of the MAPE values of the models.

In [16], the authors use LSTM and XGBoost to forecast future values and evaluate performance with keeping ARIMA as the baseline. The feature is selected using XGB and then pushed to LSTM. We further extrapolate and apply this information to all 3 models to give the best possible outcome.

In [17], the authors work on unsupervised change point detection using the Iterative Cumulative Sum of Squares (ICSS). This information is used to predict trends of financial time series yielding an accuracy of 92.47% for short-term predictions. The dataset used to achieve this result is generated using a traditional AutoRegressive Moving Average (ARMA) algorithm. The proposed model allows for predictions with relatively small amounts of data which enables it to be used in real-time and retrospect.

In [18], the authors work on the evaluation of imputation by feature importance (IBFI) on soil radon gas concentration (SRGC) time-series data. A minimum RMSE of 1141.1 is achieved on MCAR (missing completely at random) 10 to 30

In [19], the authors propose a new RNN model, NCGU. The datasets used for evaluation are air quality data from nowapi, Hang Seng Index, and gold price data from the Wind database with which MAEs of 6.1494, 254.4748, and 8.7322 are obtained respectively. NCGU can resolve issues caused by long-term data dependence of RNNs such as gradient disappearance and explosion.

In [20], the authors propose a model which decomposes a time series into intrinsic mode functions (IMFs) using complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN). Based on the Augmented Dickey-Fuller (ADF) test, ARMA is used on stationary time series, and LSTM on unstable time series. The datasets evaluated are DAX, HSI, S&P500, and SSE. An MAE, RMSE, and MAPE of 72.33, 101.83, and 0.60 are achieved respectively on the DAX dataset.

In [21], the authors propose a hybrid CNN-LSTM model which predicts large gaps of missing values in the SHS home appliances electricity consumption time-series datasets. The model achieves an average RMSE value of 5678.

In [22], the authors propose a model which uses a multimodal variational autoencoder to extract high-level features of multi-variate time series followed by LSTM to forecast the time series. Furthermore, a certified noise injection mechanism is proposed to improve the accuracy and robustness of the framework. The dataset used is the settle prices of soybean from USDA (United States Department of Agriculture, CBOT Exchange, and Quandl platform for the period of Jan 1956 to Jan 2020. The proposed model, DP-MAELS, achieves an average MAPE value of 0.015.

In [23], the authors propose a model, similar to that proposed in [20], where the time series is divided into IMFs using complementary ensemble empirical mode decomposition (CEEMD) from which the abstract features are fed into an LSTM to predict the closing prices on a day-to-day basis. The datasets used are the Shanghai Composite Index, SZSE Component Index, GEM Index, HSI, DJI, and the S&P500 Index for the period Jan 2010 to Apr 2018. The CEEMD-PCA-LSTM model yields an average NMSE value of 0.8471.

In [24], the authors propose a fuzzy LSTM model with reduced computational complexity to forecast time series data. They employ RMSE and SMAPE as performance metrics for TAIEX stock information from 2005 to 2010. Nemenyi hypothesis test is employed to determine the statistical superiority of the proposed method, having a lowest mean rank of 25.5 compared to competitors.

In [25], the authors propose a hybrid model consisting of Prophet providing seasonality information followed by a bidirectional LSTM for a deseasonalized version of the data. This data consists of monthly energy consumption values of 7 countries for ten and a half years. It is found to perform better than baseline models like SVR, and ARIMA on its own.

In [26], the authors propose a LightGBM optimized LSTM for short-term stock price forecasting. Performance is compared with deep network models like RNN, and GRU. The stock data used is of Shanghai and Shenzhen from CSI 300 index. RMSE and MAE are used as evaluation metrics and it is found that the proposed model outperforms others.

In [27], a novel fuzzy transfer learning is employed by the authors. The current time point value is found to be highly dependent on the previous time points. The proposed FuzzyTL performs better than the Wang-Mendel method which is used to predict the Mackey-Glass time series.

In [28], the authors propose a method to detect events using text data from information sources, apply NLP to them, and quantify the impact of the events on the stock market. This text data is extracted from Twitter after detecting temporal points. The extracted semantic data is passed on to a GAN for forecasting time series data. This method is susceptible to unforeseen events and is inherently biased towards anonymous Twitter data.

In [29], a novel forecasting framework, NuNet, is proposed. It consists of a market information feature extractor and a target index feature extractor. The authors also propose a novel regularisation method "column-wise random shuffling". This is a data augmentation method for CNNs. The trials were conducted on three indexes: S&P500, KOSPI200, and FTSE100. It outperformed baseline models giving an MSE value up to 60% less than SingleNet(R).

In [30], a low complexity FLANN (Functional link artificial neural network) is proposed. Varied amounts of datasets are used, each serving a different purpose. One of the methods is predicting stock market trends. FLANN successfully reduces computational costs to obtain faster error convergence in comparison to other models.

In [31], the authors compare the LSTM model with MA and XGBoost models. Data used is Infosys stock prices from May 2018 to May 2019. RMSE and MAPE are used to evaluate the model. We use this paper as a baseline to compare our model's performances on the same factors.

The general structure expected for this article includes sections such as Introduction, Materials and Methods, Results and Discussion, Conclusion, Acknowledgments and References.

2 Materials and Methods

2.1 Data Description

The data used in this work are collected from Yahoo Finance. They consist of stock price details of Apple, TCS, and Google from September 2012 to September 2022. The closing prices of the stocks are visualized in Figures 1, 2, and 3 respectively.

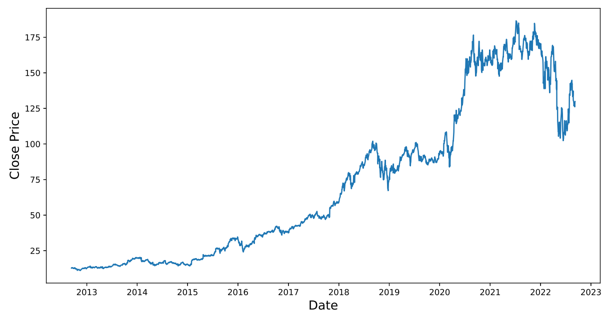

Fig. 1 Apple Closing Prices

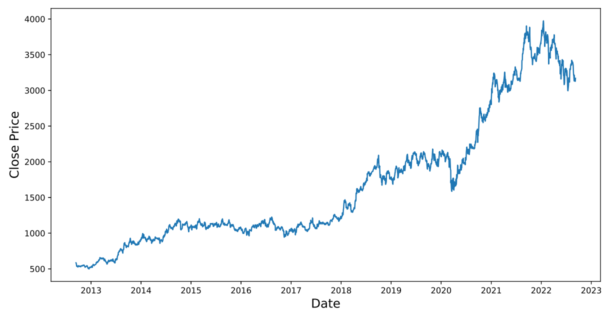

Fig. 2 TCS Closing Prices

2.2 ADF Test

One of the most popular statistical tests is the Dickey-Fuller test. It can be used to determine

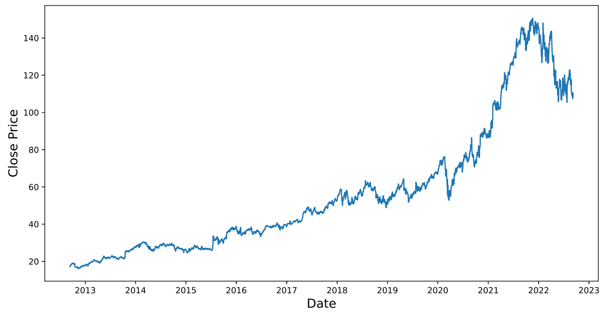

Fig. 3 Google Closing Prices

whether or not a series is stationary by recognizing whether or not it contains a unit root. The series has a unit root (a = 1), contradicting the null hypothesis. Alternate Hypothesis: There is no unit root in the series. If the null hypothesis is not rejected, we can say that the series is non-stationary. Thus, the series may be linear or difference stationary. If both the mean and standard deviation are flat (constant mean and constant variance), then the series is considered stationary.

2.3 Implementation

Our process involves a 90:10 train:test split for model training and evaluation. The test data is after the train data in the timeline. This is because time series forecasting is progressive and can only learn to predict the future if trained in the right direction of time, towards the future. Figure 4 shows the process flow used to carry out the trials.

Variables we consider to trace and predict were determined by feature importance done using XGBoost. 'Close' has a significant effect on making predictions for the stock market of a given company.

3 Results and Discussion

3.1 ARIMA Model

ARIMA (Auto Regressive Integrated Moving Average) is used after tuning the following parameters: order of AR term (p), order of MA term (q), and minimum differencing period (d). The model results are compiled in table 3.1. Using auto_arima, we arrive at the following optimal values: p = 1, d = 1, q = 0. These values represent a first-order auto-regressive model with a constant

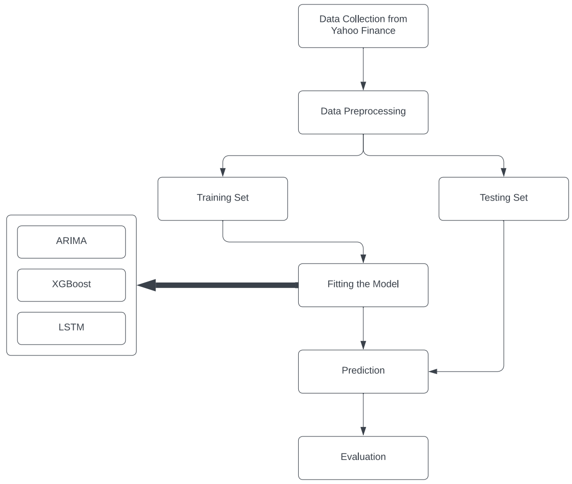

Fig. 4 Implementation of Trials

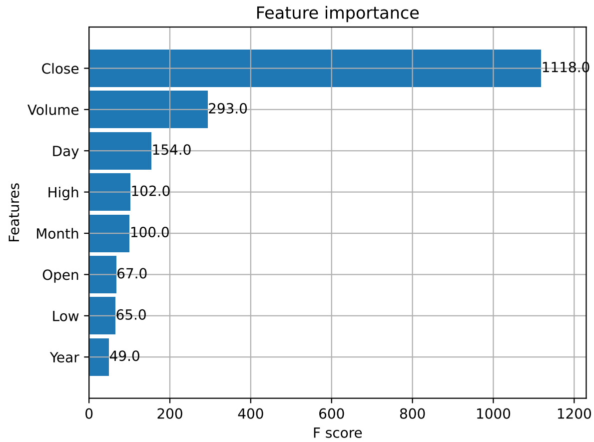

Fig. 5 Feature Importance

term and one order of nonseasonal differencing. It can be visualized in formula 1, where $\hat{Y}_i$ is the predicted price value at time t, u is the long term drift(period to period change), and P is the slope coefficient constant.

$$\hat{Y}_t = u + Y_{t-1} + P(Y_{t-1} - Y_{t-2}) \tag{1}$$

3.2 XGBoost Model

XGBoost (Extreme Gradient Boosting) is used after tuning the following parameters: Number of gradient boosted trees (n_estimators), Maximum tree depth for base learners (max_depth), Minimum sum of instance weight(hessian) needed

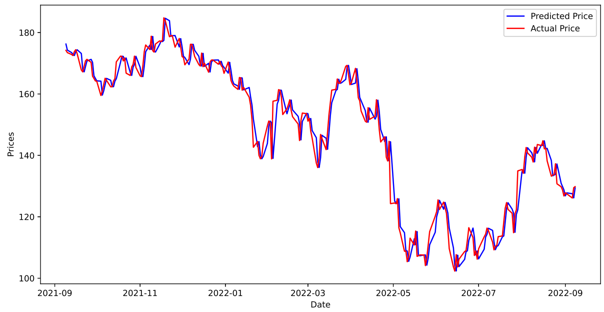

Fig. 6 ARIMA: Forecasting Apple Close Prices

Fig. 7 ARIMA: Forecasting TCS Close Prices

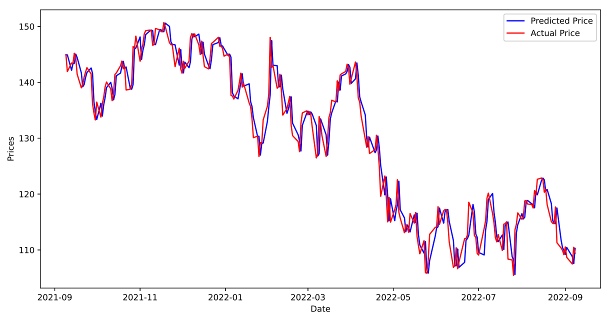

Fig. 8 ARIMA: Forecasting Google Close Prices

in a child(min_child_weight), Boosting learning rate(learning_rate). We manually tested and arrived the following optimal values: n_estimators = 1000, max_depth = 5, min_child_weight = 2, learning_rate = 0.3. The model results are compiled in table 3.2. The equation 2 shows the objective function for XGBoost. [32] describes the function in detail.

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y_i}^{(t-1)} + f_t(X_i)) + \Omega(f_t) \quad (2)$$

Table 1 ARIMA Model Evaluation

Stock	**Model**	**Metrics**
Apple	ARIMA	MSE: 14.847 MAE: 2.833 RMSE: 3.853 MAPE: 0.020 Acc: 97.97%
TCS	ARIMA	MSE: 2760.976 MAE: 38.509 RMSE: 52.544 MAPE: 0.010 Acc: 98.90%
Google	ARIMA	MSE: 7.186 MAE: 2.074 RMSE: 2.680 MAPE: 0.016 Acc: 98.36%

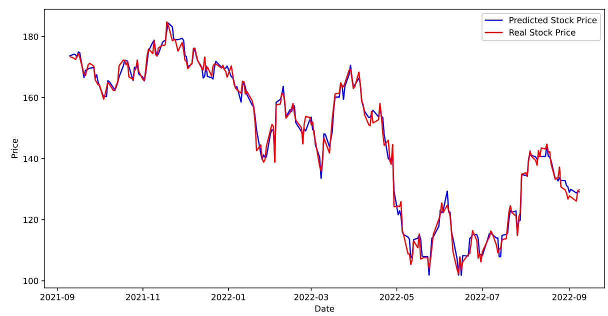

Fig. 9 XGB: Forecasting Apple Close Prices

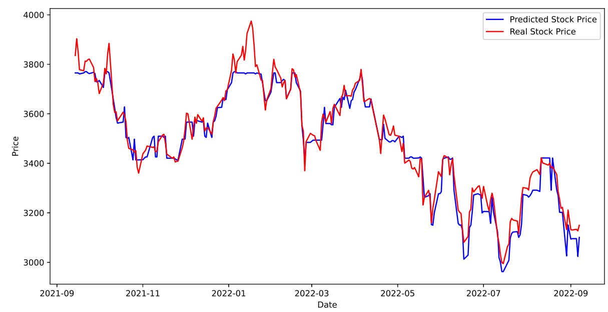

Fig. 10 XGB: Forecasting TCS Close Prices

3.3 LSTM Model

LSTM (Long Short-Term Memory) is run using 100 units and the default activation function (tanh). Using manual tuning, we arrived at the following optimal values for fitting the model: epochs = 20, batch_size = 10. The model results are compiled in table 3.3.

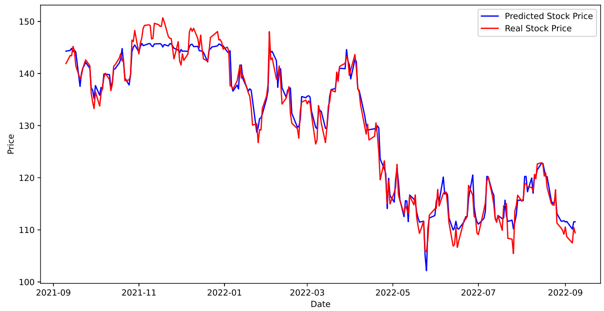

Fig. 11 XGB: Forecasting Google Close Prices

Table 2 XGB Model Evaluation

Stock	Model	Metrics
Apple	XGB	MSE: 4.748 MAE: 1.651 RMSE: 2.179 MAPE: 0.0118 Acc: 98.81%
TCS	XGB	MSE: 2442.076 MAE: 37.637 RMSE: 49.417 MAPE: 0.010 Acc: 98.92%
Google	XGB	MSE: 3.668 MAE: 1.519 RMSE: 1.915 MAPE: 0.011 Acc: 98.81%

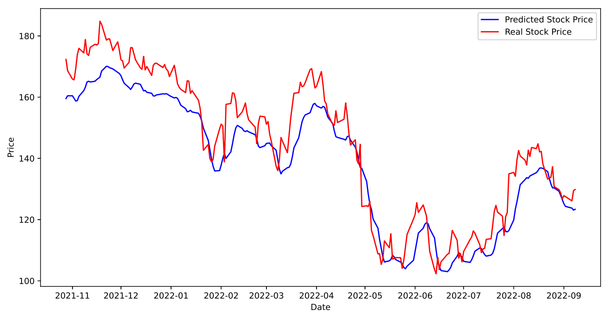

Fig. 12 **LSTM: Forecasting Apple Close Prices**

3.4 Evaluation Metrics

Mean squared error (MSE) is the average of the squares of the errors, i.e., the average squared difference between the predicted values and the actual values. The MSE for our trials varies drastically based on the dataset for the same model.

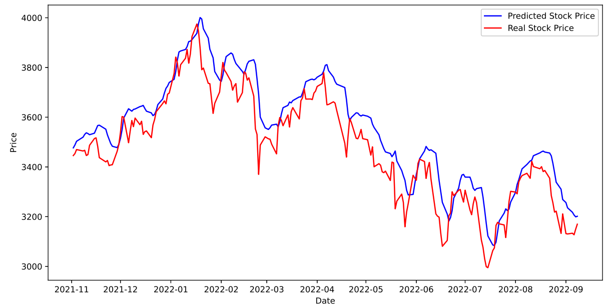

Fig. 13 LSTM: Forecasting TCS Close Prices

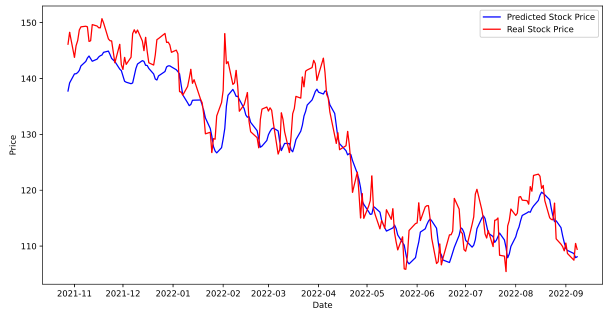

Fig. 14 LSTM: Forecasting Google Close Prices

Table 3 LSTM Model Evaluation

Stock	Model	Metrics
Apple	LSTM	MSE: 42.300 MAE: 5.374 RMSE: 6.503 MAPE: 0.037 Acc: 96.22%
TCS	LSTM	MSE: 4726.423 MAE: 52.888 RMSE: 68.748 MAPE: 0.015 Acc: 98.45%
Google	LSTM	MSE: 37.227 MAE: 5.120 RMSE: 6.101 MAPE: 0.038 Acc: 96.12%

This is because the difference between the predicted and actual values in some parts of the data tends to vary more than the general variation causing the squares to be significantly high. The general formula for MSE is stated in equation 3,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \quad (3)$$

where n is the number of data points, Y_i is the observed values, and $\hat{Y}_i$ is the predicted values.

Mean absolute percentage error (MAPE) is commonly used for measuring the prediction accuracy of forecasting methods in statistics. It is critical for model evaluation. It is more consistent compared to MSE when there are predicted values that are unusually off from the actual values. The general formula for MAPE is stated in equation 4,

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \quad (4)$$

where A_t is the actual value, F_t is the forecasted value, and n is the number of fitted points.

The accuracy of the models is calculated using MAPE values which happen to produce results that can be visually affirmed with the graphs. The formula for calculating accuracy is stated in equation 5,

$$Acc = (1 - MAPE) \times 100 \quad (5)$$

3.5 Baseline Comparison

We use [31] as a baseline to compare performance of our models. We used the exact same data as the paper (Infosys) for the same dates and the same evaluation metric (RMSE) for a fair comparison. We found that our model and hyper parameters perform better, as demonstrated in table 3.5.

Table 4 Baseline Model Comparison

Model	Novel RMSE	Baseline RMSE
ARIMA	8.99	10.094
XGB	5.11	8.807
LSTM	1.89	7.834

4 Conclusion

Through our paper, we successfully compare the Machine Learning models for Time Series Forecasting. Based on our trials, we observe that XGBoost has the highest accuracy among the 3 models worked upon. In some instances, XGBoost fails to make the right predictions at key turning points in the stock price. LSTM shows better consistency in making optimal predictions with a slight reduction in accuracy, which can be favorable depending on the investment strategy. ARIMA showcases high accuracy due to the short forecasting range. It is suitable for long-term trend recognition but the forecasting is not reliable.

In conclusion, XGBoost is desirable for long-term investments as small inconsistencies can be ignored, whereas LSTM is favorable for short-term investments where consistency is important. ARIMA may be more accurate compared to LSTM, but this is due to it depending on recent trends in the prices, making it more suitable for applications such as day trading.

Declarations

Funding No funds, grants, or other support was received.

Conflicts of interest The authors declare that there is no conflict of interest.

Author contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Sarthak Khare and Gaurav Goverdhan. The first draft of the manuscript was reviewed by Manoov R and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

References

[1] Yang, J., Zhao, J., Song, J., Wu, J., Zhao, C., Leng, H.: A hybrid method using havok analysis and machine learning for predicting chaotic time series. Entropy **24**, 408 (2022). https://doi.org/10.3390/e24030408

[2] Parray, I.R., Khurana, S.S., Kumar, M., Altalbe, A.A.: Time series data analysis of stock price movement using machine learning techniques. Soft Computing **24**, 16509–16517 (2020). https://doi.org/10.1007/s00500-020-04957-x

[3] Xu, W., Peng, H., Zeng, X., Zhou, F., Tian, X., Peng, X.: A hybrid modelling method for time series forecasting based on a linear regression model and deep learning. Applied

Intelligence **49**, 3002–3015 (2019). https://doi.org/10.1007/s10489-019-01426-3

[4] Iqbal, M., Iqbal, M., Jaskani, F., Iqbal, K., Hassan, A.: Time-series prediction of cryptocurrency market using machine learning techniques. EAI Endorsed Transactions on Creative Technologies **8**, 170286 (2021). https://doi.org/10.4108/eai.7-7-2021.170286

[5] Yeung, J.F.K.A., Wei, Z.-k., Chan, K.Y., Lau, H.Y.K., Yiu, K.-F.C.: Jump detection in financial time series using machine learning algorithms. Soft Computing **24**, 1789–1801 (2020). https://doi.org/10.1007/s00500-019-04006-2

[6] Chou, J.-S., Truong, D.-N., Le, T.-L.: Interval forecasting of financial time series by accelerated particle swarm-optimized multi-output machine learning system. IEEE Access **8**, 14798–14808 (2020). https://doi.org/10.1109/ACCESS.2020.2965598

[7] Parmezan, A.R.S., Souza, V.M.A., Batista, G.E.A.P.A.: Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. Information Sciences **484**, 302–337 (2019). https://doi.org/10.1016/j.ins.2019.01.076

[8] Fernández, C., Salinas, L., Torres, C.E.: A meta extreme learning machine method for forecasting financial time series. Applied Intelligence **49**, 532–554 (2019). https://doi.org/10.1007/s10489-018-1282-3

[9] Ramadevi, B., Bingi, K.: Chaotic time series forecasting approaches using machine learning techniques: A review. Symmetry **14**, 955 (2022). https://doi.org/10.3390/sym14050955

[10] Li, Z., Han, J., Song, Y.: On the forecasting of high-frequency financial time series based on arima model improved by deep learning. Journal of Forecasting **39**, 1081–1097 (2020). https://doi.org/10.1002/for.2677

[11] Sheeba, S.L., Gupta, N., Ragavender, R.A., Divya, D.: Time series model for stock market prediction utilising prophet. Turkish Journal of Computer and Mathematics Education **12**(6), 4529–4534 (2021)

[12] Singh, A., Gupta, P., Thakur, N.: An empirical research and comprehensive analysis of stock market prediction using machine learning and deep learning techniques. IOP Conference Series: Materials Science and Engineering **1022** (2021). https://doi.org/10.1088/1757-899X/1022/1/012098

[13] Chan, W.N.: Time series data mining: Comparative study of arima and prophet methods for forecasting closing prices of myanmar stock exchange. Journal of Computer Applications and Research **1**, 75–80 (2020)

[14] Moghar, A., Hamiche, M.: Stock market prediction using lstm recurrent neural network. Procedia Computer Science **170**, 1168–1173 (2020). https://doi.org/10.1016/j.procs.2020.03.049

[15] Selvin, S., Vinayakumar, R., Gopalakrishnan, E.A., Menon, V.K., Soman, K.P.: Stock price prediction using lstm, rnn and cnn-sliding window model. 2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017 **2017-Janua**, 1643–1647 (2017). https://doi.org/10.1109/ICACCI.2017.8126078

[16] Vuong, P.H., Dat, T.T., Mai, T.K., Uyen, P.H., Bao, P.T.: Stock-price forecasting based on xgboost and lstm. Computer Systems Science and Engineering **40**, 237–246 (2022). https://doi.org/10.32604/CSSE.2022.017685

[17] Kim, K., Park, J.H., Lee, M., Song, J.W.: Unsupervised change point detection and trend prediction for financial time-series using a new cusum-based approach. IEEE Access **10**, 34690–34705 (2022). https://doi.org/10.1109/ACCESS.2022.3162399

[18] Mir, A.A., Celebi, F.V., Rafique, M., Hussain, L., Almasoud, A.S., Alajmi, M., Al-Wesabi, F.N., Hilal, A.M.: An improved imputation method for accurate prediction of imputed

dataset based radon time series. IEEE Access **10**, 20590–20601 (2022). https://doi.org/10.1109/ACCESS.2022.3151892

[19] Wang, J., Li, X., Li, J., Sun, Q., Wang, H.: Ngcu: A new rnn model for time-series data prediction. Big Data Research **27**, 100296 (2022). https://doi.org/10.1016/j.bdr.2021.100296

[20] Lv, P., Wu, Q., Xu, J., Shu, Y.: Stock index prediction based on time series decomposition and hybrid model. Entropy **24**, 146 (2022). https://doi.org/10.3390/e24020146

[21] Hussain, S.N., Aziz, A.A., Hossen, M.J., Aziz, N.A.A., Murthy, G.R., Mustakim, F.B.: A novel framework based on cnn-lstm neural network for prediction of missing values in electricity consumption time-series datasets. Journal of Information Processing Systems **18**, 115–129 (2022). https://doi.org/10.3745/JIPS.04.0235

[22] Li, H., Cui, Y., Wang, S., Liu, J., Qin, J., Yang, Y.: Multivariate financial time-series prediction with certified robustness. IEEE Access **8**, 109133–109143 (2020). https://doi.org/10.1109/ACCESS.2020.3001287

[23] Zhang, Y., Yan, B., Aasma, M.: A novel deep learning framework: Prediction and analysis of financial time series using ceemd and lstm. Expert Systems with Applications **159**, 113609 (2020). https://doi.org/10.1016/j.eswa.2020.113609

[24] Pattanayak, R.M., Sangameswar, M.V., Vodnala, D., Das, H.: Fuzzy time series forecasting approach using lstm model. Computacion y Sistemas **26**, 485–492 (2022). https://doi.org/10.13053/CyS-26-1-4192

[25] Arslan, S.: A hybrid forecasting model using lstm and prophet for energy consumption with decomposition of time series data. PeerJ Computer Science **8** (2022). https://doi.org/10.7717/PEERJ-CS.1001

[26] Lv, J., Wang, C., Gao, W., Zhao, Q.: An economic forecasting method based on the lightgbm-optimized lstm and time-series model. Computational Intelligence and Neuroscience **2021** (2021). https://doi.org/10.1155/2021/8128879

[27] Pal, S.S., Kar, S.: Fuzzy transfer learning in time series forecasting for stock market prices. Soft Computing **26**, 6941–6952 (2022). https://doi.org/10.1007/s00500-021-06648-7

[28] Maître, E., Dousset, B., Chemli, Z., Gitto, J.P., Chevalier, M., Teste, O.: Event detection and time series alignment to improve stock market forecasting. CEUR Workshop Proceedings **2621** (2020)

[29] Lee, S.W., Kim, H.Y.: Stock market forecasting with super-high dimensional time-series data using convlstm, trend sampling, and specialized data augmentation. Expert Systems with Applications **161**, 113704 (2020). https://doi.org/10.1016/j.eswa.2020.113704

[30] Mohapatra, U.M., Majhi, B., Satapathy, S.C.: Financial time series prediction using distributed machine learning techniques. Neural Computing and Applications **31**, 3369–3384 (2019). https://doi.org/10.1007/s00521-017-3283-2

[31] Kumar, R., Kumar, P., Kumar, Y.: Analysis of financial time series forecasting using deep learning model. Proceedings of the Confluence 2021: 11th International Conference on Cloud Computing, Data Science and Engineering, 877–881 (2021). https://doi.org/10.1109/Confluence51648.2021.9377158

[32] Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016)