Research Article

Don Bosco Institute of Technology Delhi Journal of Research
Year 2024, Volume-1, Issue-2 (July - Dec)

A Comparative Analysis of LSTM, ARIMA, and MCMC Models for Forecasting Financial Market

Deepika Kirti

(Ph.D. Scholar, Department of Computer Science & Engineering, Delhi Skill and Entrepreneurship University)

ARTICLE INFO

Keywords: ARIMA, LSTM, MCMC, Bayesian Model, Stock prediction, Stock market

doi: 10.48165/dbitdjr.2024.1.02.04

ABSTRACT

Machine learning, with its sophisticated predictive capabilities, has fundamentally transformed numerous fields, including stock trading. This research presents a comparative study of Bayesian inference using the Markov Chain Monte Carlo (MCMC) method in stock price prediction, the new paradigm of long short-term memory (LSTM) networks, and conventional auto-regressive integrated moving average (ARIMA) models. The study uses historical stock price data from a diverse array of companies to evaluate these models concerning their accuracy, robustness, and computational efficiency. The findings contribute to the ongoing discourse on effective forecasting methodologies in financial markets, offering valuable insights to stakeholders, practitioners, and researchers. These insights are particularly pertinent for navigating financial environments' inherently unpredictable and dynamic nature.

Introduction

A key component of international finance is the stock market, a complicated ecology of businesses, investors, and economic factors. Forecasting its future trends has been a recurring task for analysts, investors, and financial organizations. Precisely predicting stock prices can result in well-informed investing choices, enhanced risk mitigation tactics, and optimized portfolio management. However, it is still very challenging to predict stock prices with any level of accuracy, especially in efficient markets. (Shahwan et.al., 2006).

New developments in statistical modeling and machine learning have made it possible to use more advanced instruments for stock market forecasting. Autoregressive integrated moving average (ARIMA) models are considered to be among the most widely used linear models in time series forecasting due to their theoretical complexity and accuracy in short-term forecasting (Jhee et.al, 1996). Since financial markets operate under the assumption of bounded rationality, non-linear patterns are difficult for ARIMA models to represent (McNelis et.al, 2005).

However, the emergence of deep learning architectures such as Long Short-Term Memory (LSTM) networks and the continuous refinement of Bayesian approaches such as Markov Chain Monte Carlo (MCMC) systems have opened up new avenues for stock price prediction. LSTM models perform better than other models in forecasting over previous years, as a substantial amount of studies has shown. In time series data, LSTM systems can identify long-term relationships and nonlinear patterns particularly well. For this reason, they are frequently used in the analysis

*Corresponding author.
E-mail address: Email id- deepika.btts@gmail.com

of historical stock prices. Their capacity to "remember" previous knowledge comes in handy when figuring out intricate connections between financial data.

Utilizing historical data and residuals (errors), the Autoregressive Integrated Moving Average (ARIMA) statistical technique predicts future values. In financial data, seasonality— predictable fluctuations based on timeframes— and trends—upward or downward movements—are best captured by ARIMA models (Hyndman, R. J et.al. 2018). Their power comes from their capacity to simulate stationary data, which is a feature that historical stock prices frequently display after trend and seasonality corrections.

MCMC is a sampling-based simulation technique that creates a dependent sample from a certain distribution of interest. This computational method provides a strong tool for quantifying uncertainty and probabilistic modelling. To estimate parameters and take into consideration the intrinsic volatility of the stock market, MCMC simulations can be combined with other models (Tse YK et.al., 2003). In Bayesian inference, a number of MCMC method implementation techniques are commonly employed. These include the Gibbs sampler, which was first introduced by (Roberts GO et.al., 2003), and the Metropolis-Hasting method, which was first created by Metropolis in 1953 (Chib et al.,1995) and then Then, it was generalized further. MCMC enables a more nuanced view of the range of options inside the market by simulating a large number of possible outcomes.

Background

MCMC Model

In the past few years, statisticians have become more interested in using MCMC techniques to simulate complex, nonstandard multivariate distributions. Among these, the Gibbs sampling technique is one of the most well-known, and its influence on Bayesian statistics (Tanner et.al.,1987) has been immense as detailed in many articles.

The challenge of taking a sample from a multidimensional probability distribution p(x) arises when using the Monte Carlo method (Robert et.al., 2004). The foundation of MKML techniques is the resolution of this issue. There have been attempts, in example, to combine random simulations with extremely sophisticated functions. These efforts resulted in the Metropolis-Hastings algorithm.

To extract appropriate values from the posterior distribution P (ĩ=θ/D), this approach necessitates the usage of a basic distribution known as the proposal distribution Q (θ'/θ). The Metropolis-Hastings technique uses Q to walk randomly across the distribution space, accepting or rejecting jumps to new positions according to the likelihood of the sample (Chib et.al., 1995). The "Markov Chain" component of MCMC is this "memoryless" random walk.

The function f determines each new sample's probability. The function f needs to be proportionate to the supplied distribution to obtain samples. This proportionality is typically expressed by the function f, which is selected as a probability density function. Simply take our current value, θ, and propose a new value, θ', that is chosen at random from our distribution Q(θ'/θ), to get a new parameter position (Robert et.al., 2004).

The distribution is typically symmetric, such as a normal distribution with a mean of θ and a standard deviation of σ: Q (θ'/θ) = N (θ, σ). The accepted criterion for the sample θ' is the formula

$$P(\text{accept}) = \begin{cases} \dfrac{\prod_i^n f(d_i/\theta=\theta')P(\theta')}{\prod_i^n f(d_i/\theta=\theta)P(\theta)}, & \prod_i^n f(d_i/\theta=\theta)P(\theta) > \prod_i^n f(d_i/\theta=\theta')P(\theta') \\ 1, & \prod_i^n f(d_i/\theta=\theta)P(\theta) \leq \prod_i^n f(d_i/\theta=\theta')P(\theta') \end{cases}$$

When θ' is more likely than the current θ, we always accept it. If it is less likely than the current θ, we can arbitrarily accept or reject it with decreasing probability (Hamiane S. et al.,2024)

A stock's price is represented by the Black-Scholes formula as St, which follows a geometric Brownian motion with constant volatility and drift. The formula for

$$S_{t+\Delta t} = S_t \exp\left(\left(r - \frac{\sigma^2}{2}\right)\Delta t + \sigma z_t \sqrt{\Delta t}\right)$$

the stock price at time t + Δt is given by:

Where:

- St is the stock price at time t
- r is the risk-free interest rate (10% or 0.10)

σ is the volatility (25% or 0.25)

Δt is the time increment (1/22, representing 1 day in 22 trading days in a month)

- Zt is a random variable generated using the MCMC method

LSTM Model:

Virtual neural networks excel at extracting hidden insights from time-series data due to their capacity to capture intricate and non-linear patterns. The learning process for an artificial neural network involves ingesting past time-series datasets and transforming these inputs through multiple hidden layers within the network. Each layer applies mathematical operations to the data, enabling the network to learn intricate patterns and temporal dependencies (Staudemeyer, R. C. et.al., 2019). Recurring Neural Networks are a subdivision of Artificial Neural Networks, and LSTM is a subdivision of Recurring Neural Networks (Selvin, S et.al., 2017).

In order to improve intrusion detection by comprehending sequential linkages within network data and increasing efficiency using attention methods for feature extraction, the study makes use of an LSTM model and Transformer architectures (Nandam & Reddy, 2024).

By examining past data, the LSTM model is a kind of neural network intended to increase prediction accuracy in financial forecasting. In comparison to traditional models, this study's average accuracy of 95.53% proved its efficacy (Bi, C., & Wang, Y. 2024).

LSTM has greatly helped forecasting scenarios that contain long-term data due to its incompatibility with RNNs, which do not have long-term memory based on "memory line. Using integrated gates along the memory line, an LSTM can memorize previous phases. The structure of LSTM nodes is shown in the diagram below (Moghar, A. et.al., 2020).

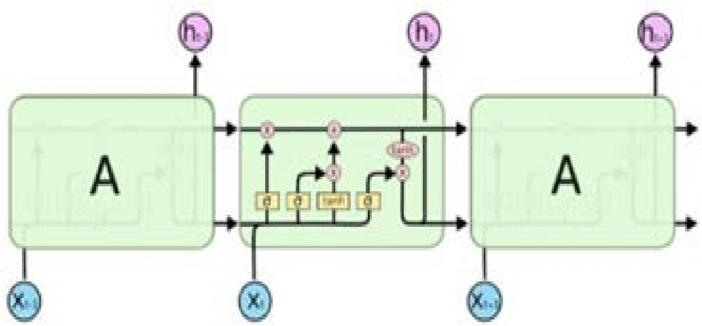

Figure 1. The internal structure of an LSTM (Olah, C. et.al. **,2015)**

The ability to memorize a data sequence sets the LSTM apart from other RNNs. Each cell's top line transports data from the past to the present and acts as a transport line between the models (Moghar, A. et.al. ,2020). The model can add values from one cell to another or remove filters because the cells are independent. A collection of cells that hold passed data streams make up the majority of an LSTM node. In the end, the sigmoidal neural network layer that comprises the gates drives the cell to an optimum value by either discarding or permitting data to pass through. Each sigmoid layer has a binary value (0 or 1) assigned to it, where 0 denotes "let nothing pass through" and 1 denotes "let everything pass through."

The goal here is to control the condition of each cell, and the gates are operated as explained below (Moghar, A. et.al. ,2020):

The ability to memorize a data sequence sets the LSTM apart from other RNNs. Each cell's top line transports data from the past to the present and acts as a transport line between the models (Moghar, A. et.al. ,2020). The model can add values from one cell to another or remove filters because the cells are independent. A collection of cells that hold passed data streams make up the majority of an LSTM node. In the end, the sigmoidal neural network layer that comprises the gates drives the cell to an optimum value by either discarding or permitting data to pass through. Each sigmoid layer has a binary value (0 or 1) assigned to it, where 0 denotes "let nothing pass through" and 1 denotes "let everything pass through."

The goal here is to control the condition of each cell, and the gates are operated as explained below (Moghar, A. et.al. ,2020):

A number between 0 and 1 represents the forget gate's output, where 0 denotes "completely ignore this" and 1 denotes "completely keep this."

The Memory Gate chooses which fresh data should be stored in the cell. First, a sigmoid layer that serves as a "input door layer" chooses which values need to be changed. A tan layer then generates a vector of new candidate values that could be added to the state.

The output of each cell is determined by the output gate. The output value will be based on the cell status and the most recent, filtered additions to the data.

ARIMA Model:

In 1970, the ARIMA model was introduced by Box and Jenkins. The Box-Jenkins methodology, as it is also known, is a set of procedures for identifying, estimating, and diagnosing ARIMA models with time series data. The model is among the most popular in financial forecasting (P. Pai et.al., 2005), (N. Merh et.al. , 2010), (Ayodele A. et.al., 2014).

The ability of the ARIMA models to generate short-term projections has been proven to be successful. It continuously outperforms complex structural models in short-term prediction (A. Meyler et.al., 1998).

In the ARIMA method, quasi-information is converted into static information before being used to predict time-series data (Xiao, et al., 2022). In the 1970s, the British statistician Jenkins GM and the American statistic Box GE P created the ARIMA (p, d, q) time stream analytics paradigm. ARIMA, also referred to as the Box-Jenkins approach, routinely produces more accurate short-term forecasts than intricate algorithms (Verma, S. et.al., 2022).

The analysis of five technology stocks shows how well the ARIMA model predicts equities (Xu, J. 2024). It analyses stock movements and projects future prices using parameters that have been optimized using the Bayesian Information Criterion (BIC) and the Akaike Information Criterion (AIC). The ARIMA model is popular for short-term stock price forecasting because of its autoregressive and moving average components (Liang, K., Wu, H., & Zhao, Y.,2024). This study demonstrates how well it predicts the values of Chinese A-share stocks over a 24-day period.

The time stream technique is ambiguous or inefficient. Many regression analyses are made possible by the ARIMA process, such as multiple regression analysis, cyclical, subgroup, and scaled ARIMA ideas, interference or stopped time sequence systems, suitable transferable functional features of any sophistication, and ARIMA errors (Varaprasad, B. N., et al., 2022). For the specified output set, idea identification and selection determine potential ARIMA algorithms. Furthermore, following real-time reading, it evaluates auto-correlation, component-independent grouping, reverse-independent grouping, and bridge, which can be utilised to differentiate those in subsequent claims. To determine if variance is necessary, you can do normalcy checks (N. Merh, et.al., 2010), (Ayodele A. et.al. 2014).

In order to match the parameter supplied in the recognition step, it calculates the model's parameters while simultaneously defining the ARIMA framework. the demand for movement averages (MA) in addition to the interpolation (I), integrating, or auto-regressive (AR) criterion. It also produces analytical statistical data that is useful for evaluating the adequacy of the model (Xiao et al, 2022). The outcomes of relevance checks on parametric estimations indicate whether any parameters are potentially unnecessary.

Prediction scheduling ARIMA algorithms cannot be fitted with uniform data or datasets that could be transformed into uniform statistics by non-seasonal or seasonally interpolation (Kobiela, et al, 2022). The majority of constant sequences, or sequences with perfect uniformity, like data with a sawtooth graph or a perfect border, do not provide an ARIMA framework match because of this feature (Varaprasad, B. N., et al., 2022).

The equations for the AR(p) paradigm, or any p-th level auto-regressive (AR) model, are as follows:

$$y_t = C + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \varepsilon_t$$

Here, "yt" is the data that will be subjected to the ARIMA algorithm. It suggests that there's a chance the order and strength have been altered in the past. The AR coefficients are denoted by the numbers φ1, φ2, and so on (Kobiela, et al, 2022).

The q-th level's movable averages (MA) linear framework, or MA(q) paradigm, is as follows:

$$y_t = C + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \ldots - \theta_q \varepsilon_{t-q}$$

Yt appears to be as previously defined, while MA factors are represented by θ1, θ2, and so forth (Siami-Namini et.al., 2018).

$$y_t = C + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \ldots - \theta_q \varepsilon_{t-q}$$

The steady factor of the ARIMA equation embeds parameter estimations into the framework and maintains it for the foreseeable future (Xiao et al, 2022). The pattern of a paradigm with one distinction and no seasonal or temporal standard error seems to be regular, while the pattern of a structure with two differences is polynomial. If a parameter has one or more inter-annual changes in addition to seasonality, the parameter is removed from the equation using the Auto-Select parameter of ARIMA coefficients in the Estimator ARIMA Parameters dialogue (Varaprasad, B. N., et al., 2022).

Results and Analysis

The present research used historical stock market data from Yahoo Finance to assess the efficacy of three forecasting models: LSTM, MCMC, and ARIMA. Python was used in Visual Studio Code to handle and analyse the data. The effectiveness of each model was assessed using Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE).

The RMSE (Root Mean Square Error) formula is indeed used to measure the error in predictions.

$$\mathrm{RMSE} = \sqrt{\tfrac{1}{n}\sum_{i=1}^{n}(\mathrm{predictions}_i}$$

Machine learning algorithms frequently employ Mean Absolute Percentage Error (MAPE) to assess prediction accuracy. The formula for MAPE is:

$$\mathrm{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_{\mathrm{test},i} - \mathrm{predictions}_i}{y_{\mathrm{test},i}}\right|$$

Table 1. Forecasting errors of LSTM, MCMC, and ARIMA

Model	RMSE	MAPE
LSTM	2.5517	1.9575
MCMC	4.4323	3.1496
ARIMA	10.6662	8.6652

Bar graph displaying the RMSE values for each prediction method. The LSTM approach has the lowest RMSE, as shown, and is followed by the MCMC and ARIMA approaches.

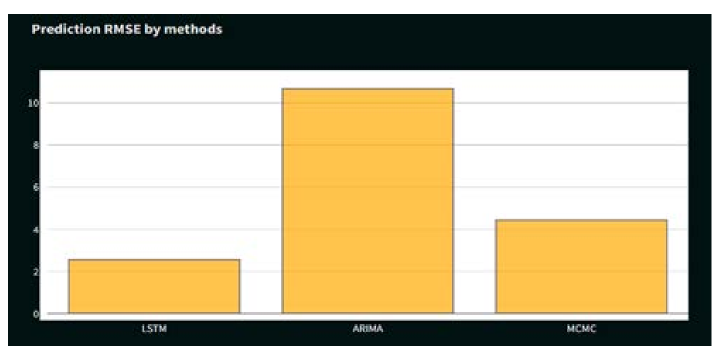

Figure 2: Bar graph for Prediction RMSE by methods

LSTM outperformed the others, with the lowest MAPE (1.9575) and RMSE (2.5517). This shows that the LSTM is quite good at forecasting accurately and reducing prediction mistakes. Its ability to recognise complex patterns and long-term connections in stock market data accounts for its high accuracy. The effectiveness of the LSTM model is demonstrated by how well it manages non-linear interactions and makes more accurate predictions of volatile patterns.

MCMC performed better than ARIMA, with an RMSE of 4.4323 and a MAPE of 3.1496, but not being as precise as LSTM. Based on its performance, MCMC can manage uncertainty quantification and probabilistic modeling. Even while MCMC isn't as accurate as LSTM, it's still a useful substitute when it comes to knowing how uncertain projections are.

With the highest RMSE (10.6662) and MAPE (8.6652), ARIMA performed the worst. The complex, non-linear patterns typical of stock market data seem to be beyond the scope of the model's linear assumptions and structure. This restriction implies that although ARIMA might work well for simpler datasets or linear scenarios, it is not a good fit for predicting the highly volatile and non-linear tendencies of the stock market.

Conclusion

The analysis shows that, due to its lower RMSE and MAPE, LSTM is the most useful model for stock market forecasting out of the three. The ability of LSTM to identify intricate patterns and enduring dependencies present in stock market data. Its proficiency in managing non- linear relationships makes it a valuable tool for high-accuracy forecasting in volatile markets.

While ARIMA is less successful in capturing the nuances of stock market behavior. Consequently, while ARIMA might be suitable for simpler datasets or scenarios where linear relationships dominate, it falls short in the context of stock markets' volatile and intricate nature., MCMC provides a reasonable compromise between accuracy and uncertainty quantification. MCMC offers a viable alternative when probabilistic modeling and uncertainty quantification are essential. Its performance highlights its effectiveness in scenarios where capturing uncertainty is as critical as prediction accuracy. These results highlight how crucial it is to choose a forecasting model that considers the unique properties and needs of the data.

References

Tanner, M. and Wong, W. (1987). "The calculation of posterior distributions by data augmentation." J. American Statist. Assoc., 82: 528–550.

Chib, S., & Greenberg, E. (1995). Understanding the Metropolis-Hastings Algorithm. The American Statistician, 49(4), 327–335. https://doi.org/10.1080/00031305.1995.10476177

JHEE W, SHAW M. Neural Networks in Finance and Investing. Revised Edition, Irwin, Chicago. 1996.

Meyler, A., Kenny, G., & Quinn, T. (1998). Forecasting Irish inflation using ARIMA models.

Roberts GO, Rosenthal JS. Markov chain monte carlo. Encyclopedia of the Actuarial Sciences. 2003.

Tse YK, Zhang X, Yu J. Estimation of hyperbolic diffusion using the Markov chainMonte Carlo method. Quantitative Finance. 2003 Dec 3;4(2):158.

Robert, C. P., & Casella, G. (2004). Monte Carlo Statistical methods. In Springer texts in statistics. https://doi.org/10.1007/978-1-4757-4145-2

McNelis PD. Neural networks in finance: gaining predictive edge in the market. Elsevier; 2005 Jan 20.

Pai, P. F., & Lin, C. S. (2005). A hybrid ARIMA and support vector machines model in stock price forecasting. Omega, 33(6), 497-505.

Shahwan T. Forecasting commodity prices using artificial neural networks. Mensch-und-Buch- Verlag; 2006.

Merh, N., Saxena, V. P., & Pardasani, K. R. (2010). A comparison between hybrid approaches of ANN and ARIMA for Indian stock trend forecasting. Business Intelligence Journal, 3(2), 23-43.

Graves A, Graves A. Long short-term memory. Supervised sequence labelling with recurrent neural networks. 2012:37-45.

Akandwanaho, S. M., Adewumi, A. O., & Adebiyi, A. A. (2014). Solving dynamic traveling salesman problem using dynamic Gaussian process regression. Journal of Applied Mathematics, 2014(1), 818529.

Dai, A. M. (2015). Document embedding with paragraph vectors. arXiv preprint arXiv:1507.07998.

Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017, September). Stock price prediction using LSTM, RNN and CNN-sliding window model. In 2017 international conference on advances in computing, communications and informatics (icacci) (pp. 1643-1647). IEEE.

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2018, December). A comparison of ARIMA and LSTM in forecasting time series. In 2018 17th IEEE international conference on machine learning and applications (ICMLA) (pp. 1394-1401). Ieee.

Talagala TS, Hyndman RJ, Athanasopoulos G. Meta-learning how to forecast time series. Monash Econometrics and Business Statistics Working Papers. 2018 May;6(18):16.

Staudemeyer, R. C., & Morris, E. R. (2019). Understanding LSTM--a tutorial into long short-term memory recurrent neural networks. arXiv preprint arXiv:1909.09586.

Moghar, A., & Hamiche, M. (2020). Stock market prediction using LSTM recurrent neural network. Procedia computer science, 170, 1168-1173.

Xiao, R., Feng, Y., Yan, L., & Ma, Y. (2022). Predict stock prices with ARIMA and LSTM. arXiv preprint arXiv:2209.02407.

Verma, S., Prakash Sahu, S., & Prasad Sahu, T. (2022, March). Ensemble approach for stock market forecasting using ARIMA and LSTM model. In Proceedings of Third International Conference on Intelligent Computing, Information and Control Systems: ICICCS 2021 (pp. 65-80). Singapore: Springer Nature Singapore.

Varaprasad, B. N., Kanth, C. K., Jeevan, G., & Chakravarti, Y. K. (2022, March). Stock Price Prediction using Machine Learning. In 2022 International Conference on Electronics and Renewable Systems (ICEARS) (pp. 1309-1313). IEEE.

Kobiela, D., Krefta, D., Król, W., & Weichbroth, P. (2022). ARIMA vs LSTM on NASDAQ stock exchange data. Procedia Computer Science, 207, 3836-384

Hamiane, S., Ghanou, Y., Khalifi, H., & Telmem, M. (2024). Comparative analysis of LSTM, ARIMA, and hybrid models for forecasting future GDP. Journal homepage: http://iieta.org/journals/isi, 29(3), 853-861.

Bi, C., & Wang, Y. (2024, August). Long Short Term Memory Network (LSTM) Model Based on Neural Networks in Financial Forecasting. In 2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON) (pp. 1-5). IEEE.

Nandam, M. D., & Reddy,M. B. S. (2024). Detection of intrusions using transformer-LSTM model. Interantional Journal of Scientific Research in Engineering and Management, 8(008).

Xu, J. (2024). Technology Company Stock Trend Prediction Based on ARIMA Model. Advances in Economics, Management and Political Sciences, 94, 240-247.

Liang, K., Wu, H., & Zhao, Y. (2024). Study of effectiveness of the ARIMA method in forecasting stock process in China. Theoretical and Natural Science, 43, 33-44