Enhancing Forecasting Accuracy with a Moving Average-Integrated Hybrid ARIMA-LSTM Model

Sumalatha Saleti[1,2*], Lovely Yeswanth Panchumarthi[1,2*], Yogeshvar Reddy Kallam[2], Lavanya Parchuri[2], Shilpa Jitte[2†]

[1*]Computer Science Engineering, SRM AP University, Amaravati, Guntur, 522502, Andhra Pradesh, India.
[2]Computer Science Engineering, SRM AP University, Amaravati, Guntur, 522502, Andhra Pradesh, India.

*Corresponding author(s). E-mail(s): sumalatha.s@srmap.edu.in; lovelyyeswanth2002@gmail.com;
Contributing authors: yogeshvar_reddy@srmap.edu.in; srilayaparchuri10@gmail.com; jitteshilpasri@gmail.com;
†These authors contributed equally to this work.

Abstract

This research provides a time series forecasting model that is hybrid which combines Long Short-Term Memory (LSTM) and Autoregressive Integrated Moving Average (ARIMA) models with moving averages. For modelling stationary time series, LSTM models are utilized, while modelling non-stationary time series is done using ARIMA models. While LSTM models are more suited for capturing long-term dependencies, ARIMA models are superior in catching short-term relationships in time series data. The hybrid model combines the short-term dependency modeling of ARIMA utilising LSTM's long-term dependency modelling. This combination leads to greater accuracy predictions for time series data that are both stationary and non-stationary. Also, Triple Exponential Moving Average (TEMA), Weighted Moving Average (WMA), Simple Moving Average (SMA), and six other moving averages were examined to determine how well the hybrid model performed Kaufman Adaptive Moving Average (KAMA), MIDPOINT, MIDPRICE individually helps to know which methods give much precision. The study compares the hybrid model's predicting performance to that of standalone ARIMA and LSTM models, in addition to other prominent forecasting approaches like linear regression and random forest. The findings indicate that the hybrid model surpasses the individual models and other benchmark methods, achieving increased precision in terms of mean absolute percentage error

(MAPE) and Root mean squared error(RMSE). The research also investigates the impact of different hyperparameters and model configurations on performance forecasts, giving information about the ideal settings for the hybrid model. Overall, the proposed ARIMA-LSTM hybrid model with moving averages is a promising approach for accurate and reliable stock price forecasting, which has practical implications for financial decision-making and risk management.

Keywords: Time Series, Forecasting, Prediction, ARIMA, LSTM, Hybrid, Moving Averages.

Conflicts of interest: All authors declare that they have no conflicts of interest.
Data Avaialability: Data is openly available in a public repository.

1 Introduction

Time Series Analysis [1] is a valuable and comprehensive resource that explores the intricacies of temporal data examination. It presents a thorough exploration of the fundamental concepts and techniques used in analyzing time-dependent data [2]. It includes the analysis of trends and patterns in time series, the modelling and estimation of time series models such as Autoregressive Integrated Moving Average (ARIMA), state-space models, and spectral analysis. It is a vital reference for academics, statisticians, and practitioners seeking to deepen their comprehension of time series analysis and its wide-ranging applications in fields such as economics, finance, engineering, and environmental studies [3][4].

One widely utilized approach for time series analysis is the ARIMA [5][6] model, which effectively captures temporal patterns and the seasonality inherent. By decomposing the time series into autoregressive, integrated, and moving average components, the ARIMA model provides a comprehensive framework for forecasting. Model estimation involves determining appropriate parameters, including the order of differencing, terminology for moving averages and autoregressive, often using statistical methods like the Akaikmodellingation Criterion (AIC) [7] or Bayesian Information Criterion (BIC) [8].

The advantages of LSTM [9] networks in capturing complex temporal dependencies and their ability to handle long-term dependencies, make them well-suited for modelling and forecasting time series data such as river water levels during flood events. The methodology of LSTM neural networks, including the architecture and training process, emphasises their ability to learn and retain information over extended periods. The findings of the analysis showed the superior performance of the LSTM model compared to traditional approaches, showcasing its potential as a valuable tool for accurate flood forecasting [10].

A hybrid forecasting model combines the traditional ARIMA model using the deep learning LSTM model to forecast sales. They recognize the limitations of each model when used individually and propose the hybrid approach [11] to harness the complementary strengths of both methods. The ARIMA component captures the linear patterns and short-term dependencies in the export data, while the LSTM component handles the complex nonlinear relationships and long-term dependencies. They describe the process of integrating the models, where the ARIMA model is used to pre-process the data and generate residuals, which are then inputted to the LSTM model for further forecasting. The experimental Results demonstrate that the suggested hybrid model works significantly more than the individual models, indicating its effectiveness in capturing both short-term and long-term dynamics in Indonesia's export data.

The development of a methodology for estimating prediction intervals for PARMA [12] models, the field of time series forecasting by addressing the crucial issue of uncertainty estimation. By proposing a methodology for estimating prediction intervals in the context of PARMA models, it offers a valuable tool for decision-making processes that require a comprehensive understanding of the potential variability in future outcomes. Incorporating the findings and methodology would enhance the robustness and accuracy of your forecasting framework by considering prediction intervals, enabling a more comprehensive and reliable analysis.

The utilization of moving averages in the realm of time series analysis can provide valuable insights into the inherent uncertainty and indeterminacy often encountered in real-world data [10]. Neutrosophic time series, which encompass the notions of truth, indeterminacy, and falsity simultaneously, offer a powerful framework for dealing with uncertain and imprecise information. Moving averages are widely recognized for their ability to capture underlying trends and filter out noise in time series data. The inclusion of moving averages can provide a solid foundation for analyzing and interpreting the behaviour of neutrosophic time series and enabling more accurate forecasting, anomaly detection, or decision-making processes [13] [14].

Typically, anomaly detection involves identifying abnormal behavioural patterns that are reflected in the network traffic analysis parameters. All instances that deviate from those patterns (models describing the behaviour of network traffic) are categorised as potentially harmful and may be signs of an attack or abuse attempt [13]. The capacity to identify unknown assaults, such as zero-day exploits, is intimately related to the high efficiency and efficacy of methods based on anomaly detection. The fundamentals of these kinds of actions are anomaly detection and/or identification techniques that depend not on understanding how a specific attack occurs but rather on what lies outside the predetermined moral model of network data [11] [15].

The organisation of the article is as follows. The related work is presented in section 2 after the introduction. The models and algorithms used in the research work are categorized and described in section 3. The proposed Moving Average Integrated Hybrid ARIMA and LSTM are then shown in section 4. A real-world experimental setup and the outcomes of the experiment are provided and analyzed in section 5. Conclusions and future works are discussed in section 6.

2 Related Work

Typically, anomaly detection involves identifying abnormal behavioural patterns that are reflected in the network traffic analysis parameters. All instances that deviate from those patterns (models describing the behaviour of network traffic) are categorised as potentially harmful and may be signs of an attack or abuse attempt. The capacity to identify unknown assaults, such as zero-day exploits, is intimately related to the high efficiency and efficacy of methods based on anomaly detection. The fundamentals of these kinds of actions are anomaly detection and/or identification techniques that depend not on understanding how a specific attack occurs but rather on what lies outside the predetermined moral model of network data.

There are reasons to evaluate both nonparametric regression and ARIMA models for the same sort of data, such as traffic condition data, despite what may initially seem like an incongruity. Even though traffic conditions are frequently seen as stochastic, there is evidence of "chaos-like" behaviour, especially during times of congestion when traffic flow becomes unstable. According to studies, traffic movement behaves erratically, and the existence of weekly patterns lends credence to the notion that past neighbour states can produce accurate short-term forecasts. It seems sensible to compare the effectiveness of nonparametric regression techniques to ARIMA models when taking into consideration this expectation of reasonably accurate forecasts and the practical benefits of a data-driven approach [16].

To anticipate stock values, several researchers have run tests using various methods. Suhartono [17] and Kongcharoen [18] investigated the usage of Neural Networks for non-linear models and ARIMAX [19] for linear models. Their findings demonstrated that in terms of forecasting precision, ARIMAX surpassed ARIMA. The feature fusion LSTM-CNN [20] model is a hybrid model that combines CNN and LSTM. They fed time series data into the LSTM model and stock price charts into CNN. Their experiments with LSTM and CNN discovered that while increasing the number of CNN layers from 2 to 3 significantly increased running time, it did not significantly enhance performance based on the MSE success criterion [21].

On the other hand, performance did noticeably improve with more LSTM layers. By including known future features to forecast medicine sales in Germany, Helmini, Jihan, Jayasinghe, and Perera enhanced the LSTM model. Using financial time series data from Yahoo Finance, Namini, Tavakoli, and Namin evaluated the performance of BiLSTM, ARIMA, and LSTM. Their findings demonstrated that while LSTM had a faster training process than BiLSTM, the latter had a higher prediction accuracy. To predict stock prices, Selvin and their team researchers also tried out LSTM, CNN, and RNN. They discovered that CNN worked best because it used the current window to spot patterns and changes, whereas LSTM and RNN were better suited for long-term forecasts since they used historical data [22].

Neural networks have been found to consistently outperform time series analysis, as highlighted by Nanayakkara [15][23]. To enhance performance and select appropriate data types for prediction models when dealing with multivariate data, researchers have investigated the correlations between various metrics. An approach that has shown promise in forecasting future resource usage is the utilization of short-term memory (LSTM) neural networks. The outcomes of the study demonstrate the practicality and effectiveness of this system in real-world settings for predicting multivariate cloud resource utilization. To evaluate the efficiency of the proposed system, the trace data is utilized.

3 Methodology

3.1 ARIMA Model

In an autoregressive model, the relevant variable is estimated by combining predictors using a linear equation. This model forecasts the variable of interest by considering a linear combination of its previous values. The term "autoregression" indicates that the variable is regressed against itself. As a result, it is possible to create an autoregressive model of order p by specifying the appropriate parameters.

$$h_t = a + \phi_1 y_{t-1} + \phi_p y_{t-2} + \ldots + \phi_p y_{t-p} + \varepsilon_t \tag{1}$$

where ε_t is background noise, comparable to multiple regression, but using h_t's lag numbers as predictors. An AR(p) model, or an autoregressive model of order p, is what we're talking about.

Moving average models, as opposed to forecast variables, leverage prior prediction errors in a regression-like manner and previous values as in a regression model.

$$l_t = a + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q} \tag{2}$$

This is an MA(q), or moving average model, of order q. The term "decline" used here does not refer to a conventional decline since the values of t are unknown. It is important to note that each value of it can be seen as a weighted moving average of the most recent prediction errors. Forecasting future values is accomplished using a moving average model while moving average averaging is employed to detect the trend cycle in recent data.

ARIMA model is created by combining differences with a moving average model and autoregression. ARIMA stands for Autoregressive Integrated Moving Average.

$$l'_t = k + \phi_1 l'_{t-1} + \ldots + \phi_p l'_{t-p} + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q} + \varepsilon_t \tag{3}$$

The backshift notation makes it much simpler to work with components once we begin integrating them in this way to create more complex models.

$$(1 - \phi_1 C - \ldots - \phi_p C^p)(1 - C)^d h_t = c + (1 + \theta_1 C + \ldots + \theta_q C^q)\varepsilon_t \tag{4}$$

where B represents the lag operator, $(1-C)^d$ represents differencing of order d, and $h'_t = (1 - C)^d h_t$. Hence, the SARIMA model, where the extra S stands for Seasonal and write

$$X_t = \mathrm{SARIMA}(p, d, q)(P, D, Q)_m \tag{5}$$

The autoregressive, integrated, and moving average parameters in a seasonal time series model are denoted by P, D, and Q, respectively. The frequency of the seasonal events, represented by "m," determines the values of these parameters. For example, in the weekly time series, m is equal to 24, while in the monthly time series, m is equal to 12.

To quantify the degree of a time series with observations separated by a lag p and linear dependence between them, several methods can be used, for instance, the inverse autocorrelation function, partial autocorrelation function, and the autocorrelation function (IACF). By analyzing these functions, we can make value estimates of autoregressive order p, moving average order q, and differencing order d, along with their correlation [3].

To measure the level of linear dependence and assess the relationship between observations in a time series with a lag of p, various techniques can be employed. These include the inverse autocorrelation function, partial autocorrelation function, and autocorrelation function (IACF). By analyzing these functions, we can estimate the values of autoregressive order (p), moving average order (q), and differencing order (d), along with their corresponding correlations. The ARIMA model is widely utilized for time series forecasting. However, its primary limitation is that it assumes linear relationships among the time series data, which is a linear functional form. Consequently, the model fails to capture nonlinear characteristics, and it may not be suitable for complex situations where linear approximations alone are insufficient. In such cases, fitting purely linear models may not always be feasible.

3.2 LSTM Model

The recurrent neural network (RNN) is currently a popular choice for time series prediction. However, Graves introduced the Long Short-Term Memory (LSTM) model as a solution to the issues faced by RNNs regarding gradient disappearance and explosion. The LSTM unit [24], proposed by Graves, comprises three distinct gates: the input gate, the forget gate, and the output gate. These gates play a crucial role in updating the stored information within the LSTM unit. The specific steps involved in LSTM calculations are as follows:

- The forget gate calculation involves obtaining the hidden state h_{t-1} from the previous sequence and the current series data x_t. The activation function f_t, typically sigmoid, is applied to calculate the output f_t of the forget gate. The sigmoid output f_t, ranging between 0 and 1, determines the probability of forgetting the previous hidden cell state. The calculation of the forget gate can be expressed as follows:

$$f_t = \sigma(K_f h_{t-1} + U_f x_t + b_f) \tag{6}$$

  where K_f, U_f, and b_f are the coefficients and bias of the linear relationship, respectively. The sigmoid function is denoted by σ.
- The input gate's computation: There are two parts to the input gate. The first part, which is represented by h_{t-1} and x_t, defines which data should be updated using the sigmoid function. The second part makes use of the tanh function to retrieve fresh potential cell data using h_{t-1} and x_t. The two elements cooperate to update the cell's status. The equations in algebra are:

$$p_t = \sigma(K_p h_{t-1} + U_p x_t + b_p) \tag{7}$$

$$a_t = \tanh(K_a x_t + b_a) \tag{8}$$

  where K_p, U_p, b_p, K_a, U_a, and b_a represent the coefficients and bias of the linear relationships, respectively.

- The cell state, denoted as C_k, is updated based on the combined impact of the previous input gate and forget gate. The calculation of the cell update involves determining the difference between the previous cell state, C_{k-1}, and the current cell state C_k.

$$C_k = C_{k-1} \cdot f_k + i_k \cdot a_k \tag{9}$$

- The update of the hidden state h_t involves two components. The first component, denoted as O_k, is composed of the previous hidden state h_{k-1} and the current sequence data x_k. It controls the output of the current cell. The second component includes the activation functions applied to the tanh function and the cell state C_k.

$$O_k = \sigma(K_0 h_{k-1} + U_0 x_k + b_0) \tag{10}$$

$$h_k = O_k \cdot \tanh(C_k) \tag{11}$$

Similar instances of structural issues in access control and version control In the communities of structural computing and hypermedia, our work has been extensively debated.

3.3 Moving Averages

A Simple Moving Average (SMA)[25] refers to the calculation of the average stock price over a specific time period. It is obtained by taking the sum of n values and dividing it by the number of observations, represented by m, after considering the observation y_t.

$$m = \frac{1}{n} \sum_{i=0}^{n-1} y(t-i) \tag{12}$$

The steady-state model allows for the expression of prior data in terms of the current level as

$$Y_{t-i} = \theta_1 - \sum w_{t-j} + v_{t-i} \quad (j = 1 \text{ to } i) \tag{13}$$

Given that all error terms have an anticipated value of zero, the estimate will be unbiased according to the definition of an SMA [26]. It is crucial to recognise the covariance introduced by the accumulation of the disturbance terms to calculate the variance of the average, whereupon the variance of m, here labelled C_n (to signify it has originated from an SMA of length n), is equal to

$$\begin{aligned} C_n = \frac{1}{n^2}\left[nV + W - w\sum J^2\right] &= \frac{1}{n^2}\left[nV + \frac{W(n-1)n(2n-1)}{6}\right] \\ &= \frac{V}{n} + \frac{W(2n^2-3n+1)}{6n} \end{aligned} \tag{14}$$

$$C_n = \frac{V(2n-3/2)}{n^2-1/2} \tag{15}$$

The exponential marginal distribution of the intervals (X_i), also known as the synchronous distribution of intervals in point process nomenclature, is the most basic component of the EMA1 model. It describes how the interval between one event and the next is distributed. We write the following for its probability density function (p.d.f.) $f_X(x)$'s Laplace transform:

$$f_X^*(s) = E\{\exp(-sX_i)\} = E\{\exp(-s\beta\epsilon_i)\} + E\{\exp(-s\beta\epsilon_i - s\epsilon_{i+1})\}(1-\beta) \quad (16)$$

The exponential distributions of the i.i.d. random variables, let's say, have parameters of A, hence their Laplace transform is $A/(A+s)$.

$$f_X^*(s) = \frac{A}{A+\beta s} \cdot \frac{\beta A}{A+s} \cdot \frac{A}{A+s}(1-\beta) = \frac{A}{A+s} \quad (17)$$

This proves the X_i have the same exponential distributions as claimed. Thus, the number of events per unit of time, or the rate of the point process, is the parameter A [27].

The Weighted Moving Average (WMA)[28] algorithm calculates the moving average by incorporating confidence data obtained from sensors. The temporal moving average is defined in a specific manner as follows:

$$\bar{x}_p^q = \frac{(w_{p,q-k} + 1x_{p,q-k+1} + \cdots + w_{p,q}x_{p,t})}{(w_{p,q-k+1} + \cdots + w_{p,q})} \quad (18)$$

where $w_{p,q}$ denotes the degree of certainty associated with the value $x_{p,q}$, weighted by the sensor i and timestamp t. The algorithm also considers the spatial moving average, which takes into account the sensors that are geographically connected. This spatial moving average is utilized in the calculation.

$$\bar{x}_p^8 = \frac{\sum_{i\in \text{Neighbor}(p)} b_{i,t}w_{i,t}x_{j,t}}{\sum_{i\in \text{Neighbor}(p)} b_{i,t}w_{i,t}} \quad (19)$$

The weight $b_{j,t}$, determines the significance of incorporating the value of a neighboring node when computing the moving average. It controls whether or not the neighbouring node's value should be considered in the calculation [29].

Holt's linear exponential smoothing, is a popular technique for double exponential smoothing. It involves smoothing the trend and slope separately using distinct smoothing constants.

$$Y_p = \alpha Z_p + (Y_{p-1} + T_{p-1})(1-\alpha) \quad (20)$$

$$T_p = \beta(Y_p - Y_{p-1}) + (1-\beta)T_{p-1} \quad (21)$$

$$F_{p+k} = Y_p + kT_p \quad (22)$$

The above equations describe the different constants used for calculating the current time period values in a smoothed constant process value.

A moving average that modifies its sensitivity to price fluctuations based on market volatility is known as Kaufman's Adaptive Moving Average (KAMA)[30].

$$ER = \frac{|Y_b - Y_{b-m}|}{m\sum_{b=1}^{m} |Y_m - Y_{m-1}|} \quad (23)$$

$$C = (ER(0.6667 - 0.0645) + 0.0645)^2 \tag{24}$$

$$KAMA(k) = KAMA(k-1) + C(Y_k - KAMA(k-1)) \tag{25}$$

A market efficiency ratio, or ratio of direction to volatility, is used by KAMA to modify alpha. The EMA alpha value's flexibility [31] is represented by both 0.0645 and 0.6667 as values, which range from 2 to 30 bars. We won't alter these constants because they were recommended by the author [17].

A midpoint moving average is calculated by averaging the high and low prices over a specific time period. In contrast, a mispriced moving average assigns a higher weight to the high price compared to the low price over a given period. Triple Exponential Moving Average (TEMA)[32][33] is a technique that combines three exponential moving averages to reduce lag and generate faster signals.

$$TEMA_n(X) = EMA_n(EMA_n(EMA_n(X))) \tag{26}$$

4 Proposed Model (ARIMA-LSTM Hybrid Model)

To estimate stock prices, this research suggests a hybrid ARIMA-LSTM model that has been moving average-optimized. It is suggested that the linear and nonlinear components be trained using the ARIMA and LSTM models [22].

The initial step is optimizing the periods used in various moving average calculations to increase the precision of forecasting models. The code creates a list of moving averages to test, and for each moving average, a function is created using Python's Technical Analysis Library (TA-Lib) in order to compute the moving average. The kurtosis values are then calculated for each moving average, and the periods with kurtosis values are selected as optimized periods. These optimized periods can then be used in forecasting models to improve their accuracy. This step provides a framework for selecting the appropriate moving average period to use in forecasting refer to the graph Fig.1 below to understand the algorithm working flow.

Algorithm 1 calculate_kurtosis()

```
for i in range(period_range[0], period_range[1]) do
    kurtosis_results.append(i)
    for moving_average in moving_averages do
        moving_average_output ← moving_average(data, i)
        k ← kurtosis(moving_average_output)
        if ma not in kurtosis_results then
            kurtosis_results[moving_average] ← []
        end if
        kurtosis_results[moving_average].append(k)
    end for
end for
```

The low-volatility and high-volatility components are determined for each moving average. The low volatility component is obtained by subtracting the moving average

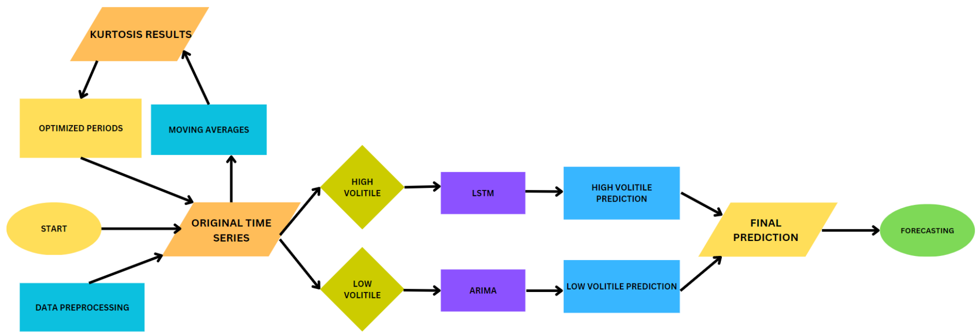

Fig. 1 *Proposed ARIMA-LSTM Model*

Algorithm 2 optimize_periods()

for moving_average in talib_moving_averages **do**
  $diff \leftarrow \text{abs}(kurtosis\_result[\text{moving\_average}] - 3)$
  $df \leftarrow \text{DataFrame}(\{'\text{difference}' : diff\})$
  $df \leftarrow df.\text{sort\_values}().\text{reset\_index}()$
  if $df.\text{at}[0, '\text{difference}'] < 3 \cdot \text{tolerance}$ **then**
    $opt[\text{moving\_averages}] \leftarrow df.\text{at}[0, '\text{period}']$
  else
    exit
  end if
end for

values from the stock's ending prices, and the ARIMA model is used for predicting future prices. The high volatility component is obtained by subtracting the low volatility component from the stock's closing prices, and the LSTM model is applied to forecast future prices. The ARIMA and LSTM models are used to forecast the low and high volatility components, respectively, thereby providing a more comprehensive forecast of future stock prices [34].
Consequently, there are two methods for doing prediction research using the network combination method described in the paper:

Combination model 1: To compute the error value, begin by making an initial sequence prediction using the ARIMA model. Then, employ the LSTM model to calculate the error value. Finally, combine the two values by adding the error predictability to the ARIMA prediction value.

Algorithm 3 Moving_Averages()

```
talib_moving_averages ← ['SMA', 'EMA', 'WMA', 'DEMA', 'KAMA', 'MIDPOINT', 'MIDPRICE', 'T3', 'TEMA', 'TRIMA']
functions ← {ma : abstract.Function(ma) for ma in talib_moving_averages}
kurtosis_results ← calculate_kurtosis(data)
optimized_period ← optimize_periods(kurtosis_results)
simulation ← {}
for ma, period in optimized_period.items() do
    low_volume ← functions[ma](data, period)
    high_volume ← data − low_volume
    low_volume_prediction, low_volume_mse, low_volume_rmse, low_volume_mape
← get_arima(low_volume)
    high_volume_prediction, high_volume_mse,
high_volume_rmse, high_volume_mape ← get_lstm(high_volume)
    final_prediction ← Series(low_volume_prediction)

    +Series(high_volume_prediction)
end for
```

Combination model 2: Since linear regression preserves both the nonlinear trend of the LSTM model and the linear trend of ARIMA, the anticipated result of ARIMA and the anticipated result of LSTM are linearly regressed to provide the anticipated result of the real value.

5 Results

Each ARIMA model implementation was suitable for its respective component due to its linear behaviour. The Auto Arima procedure was executed with seasonality set to false to determine the optimal set of parameters. The ARIMA model (0, 1, 1) was selected based on the lowest AIC value and used for making predictions for the next year. Although the individual ARIMA model's performance was below expectations, the forecasts achieved an excellent performance with a MAPE value of 2.096%, considering the small size of the ARIMA model. The results are presented in Table 1.

Model	Metric	Value
ARIMA	MSE	29.030
	RMSE	5.388
	MAPE	2.096%
LSTM	MSE	11.549
	RMSE	7.663
	MAPE	1.0251%

Table 1 Comparison of ARIMA and LSTM Models

To achieve the best result, the seasonality component is trained using various counts of LSTM epochs. A lower mean squared error (MSE) indicates better accuracy. The MSE score of 11.549 in this instance suggests that there is a significant squared difference between the anticipated and actual values. This could be attributed to the presence of outliers or the model's inability to capture certain patterns or trends in the data.

Metric	SMA	WMA	KAMA
Prediction vs Close Accuracy	49.8%	47.81%	48.61%
Prediction vs Prediction Accuracy	45.82%	45.82%	45.02%
MSE	2.6337364039	3.2187415708	3.0357476327
RMSE	1.6228790478	1.7940851626	1.7423397007
MAPE	0.6584307245	0.7684413999	0.7570769026

Table 2 Prediction of SMA, WMA & KAMA Tables

Similarly, better accuracy is indicated by a lower root mean squared error (RMSE). The RMSE value of 7.663 in this case indicates a relatively high absolute difference between the anticipated and actual values, further highlighting the challenges faced by the model in capturing the underlying patterns or trends in the data.

Metric	MIDPOINT	MIDPRICE	TRIMA
Prediction vs Close Accuracy	47.81%	52.19%	47.41%
Prediction vs Prediction Accuracy	45.42%	46.22%	45.42%
MSE	3.2430643953	2.790664792	2.9339888295
RMSE	1.8008510197	1.6705282973	1.7128890301
MAPE	0.7900534151	0.7057112134	0.6928058743

Table 3 Prediction of MIDPOINT, MIDPRICE & TRIMA Tables

Lower MAPE values are better accuracy. The MAPE value of 1.0251 suggests, on average, that the percentage difference between the predicted and actual values is around 1.03%. This indicates that the model's predictions, on average, deviate by approximately 1.03% from the actual stock prices. This is represented in Table 2.

According to the presented measurements, it seems that KAMA and MIDPOINT have the lowest MSE and RMSE values, demonstrating that their predictions are closer to the actual values. SMA, WMA, and TRIMA have higher values for MSE and RMSE, indicating that their predictions are further from the actual values. MAPE compares expected and actual values, measuring the percentage difference between the two. In this case, all models have relatively similar MAPE values, with WMA having the highest and KAMA having the lowest.

Among the indicators, KAMA and MIDPOINT showed the lowest RMSE and MSE values, showing that their forecasts were more accurate than those of the other models. As a result, it can be concluded that KAMA and MIDPOINT are more effective at capturing the underlying trends and volatility of the stock under study.

On the other hand, the higher MSE and RMSE values of SMA, WMA, and TRIMA indicate that their forecasts were more disconnected from the actual values. Yet, these

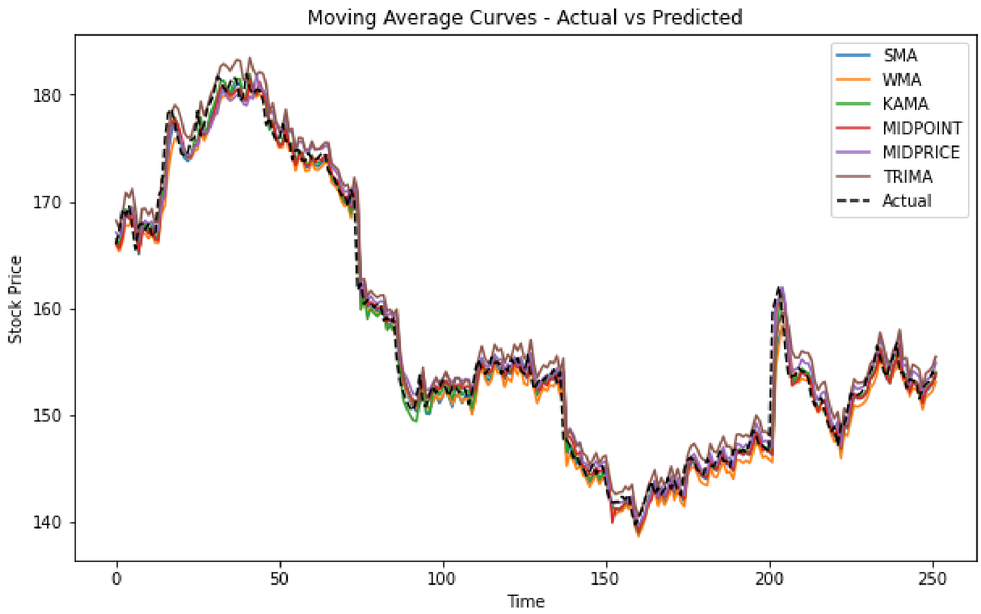

Fig. 2 Moving Average Curve - Stock Price vs Time

models still made largely accurate predictions, and there is potential for improvement to make their performance more in line with that of KAMA and MIDPOINT.

Additionally, when examining the MAPE values, it was found that all models produced relatively close outcomes, with WMA having the greatest MAPE and KAMA having the lowest MAPE. This implies that, on average, the percentage difference between the predicted and actual values was consistent across all models, with KAMA exhibiting the smallest variation.

6 Conclusions and Future Work

It is significant to note that each moving average's performance can be affected by many other variables, including the stock's underlying trend and volatility, the size of the moving average window, and the particular model parameters employed. Additionally, specific sorts of data or market circumstances could suit particular models more effectively than others. We found that the graphs from Figure 2 were strikingly comparable by visually contrasting the projected values with the actual values. This resemblance suggests that our algorithms were quite accurate in their stock value predictions.

In conclusion, our analysis shows that KAMA and MIDPOINT performed superior to the competing models. since they had lower MSE and RMSE values, which indicated more precise predictions. SMA, WMA, and TRIMA, despite slightly underperforming, nonetheless produced results that were pretty accurate. All models' consistent MAPE values point to a constant degree of relative inaccuracy. These results show the potential of using moving averages as reliable stock price forecasting tools. We may improve each model's predictive power by looking at its advantages and disadvantages as well as by investigating other variables and methods. These findings support the ongoing search for accurate and trustworthy forecasting techniques in the volatile world of financial markets.

7 Declarations

*Funding - No funding was received for conducting this study.
*Conflicts of interest/Competing interests - The authors have no competing interests to declare that are relevant to the content of this article.
*Ethics approval - Not Applicable
*Consent to participate - We have obtained informed consent from all individuals who participated in the study described in this manuscript.
*Consent for publication - We grant the right to reproduce, distribute, and publicly display the manuscript, including any figures or supplementary materials.
*Availability of data and material - Data openly available in a public repository
*Code availability - Code available on request from the authors
*Authors' contributions -
Sumalatha Saleti: Conceptualization, Methodology, Validation, Writing - review & editing, Supervision, Project administration.
Lovely Yeswanth Panchumarthi: Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Writing - original draft. Writing - review & editing, Visualization, Project administration.
Yogeshvar Reddy Kallam: Software, Validation, Investigation, Data curation.
Lavanya Parchuri: Software, Validation, Investigation, Data curation.
Shilpa Jitte: Software, Validation, Investigation, Data curation.

References

[1] Granger, C., Newbold, P.: Forecasting economic time series (1986)

[2] Esling, P., Agon, C.: Time-series data mining. ACM Computing Surveys (CSUR) **45**, 12 (2012) https://doi.org/10.1145/2379776.2379788

[3] Guha, B., Bandyopadhyay, G.: Gold price forecasting using arima model. Journal of advance Management Journal (2016) https://doi.org/10.12720/joams.4.2.117-121

[4] Canova, F., Hansen, B.E.: Are seasonal patterns constant over time? a test for seasonal stability. Journal of Business and Economic Statistics **13**, 237–252 (1995)

[5] Shumway, R., Stoffer, D.: Time Series Analysis and Its Applications With R Examples vol. 9, (2011). https://doi.org/10.1007/978-1-4419-7865-3

[6] Lin, T., Horne, W., Giles, C.: How embedded memory in recurrent neural network architectures helps learning long-term temporal dependencies. Neural networks : the official journal of the International Neural Network Society **11** (1998)

[7] Bozdogan, H.: Model selection and akaike's information criterion (aic): The general theory and its analytical extensions. Psychometrika **52**, 345–370 (1987) https://doi.org/10.1007/BF02294361

[8] Burnham, K.: Understanding aic and bic in model selection. Sociological Methods & Research - SOCIOL METHOD RES **33**, 93 (2004) https://doi.org/10.1177/0049124104268644

[9] McKenzie, E.: General exponential smoothing and the equivalent arma process. Journal of Forecasting **3** (1984) https://doi.org/10.1002/for.3980030312

[10] Xu, W., Peng, H., Zeng, X., Zhou, F., Tian, X., Peng, X.: Deep belief network-based ar model for3 nonlinear time series forecasting. Appl. Soft Comput. **77**(C) (2019) https://doi.org/10.1016/j.asoc.2019.02.006

[11] Kim, H., Won, C.: Forecasting the volatility of stock price index: A hybrid model integrating lstm with multiple garch-type models. Expert Systems with Applications **103** (2018) https://doi.org/10.1016/j.eswa.2018.03.002

[12] Vecchia, A.: Maximum likelihood estimation for periodic autoregressive moving average models. Technometrics **27**, 375–384 (1985) https://doi.org/10.1080/00401706.1985.10488076

[13] Tang, Z., Fishwick, P.: Feedforward neural nets as models for time series forecasting. INFORMS Journal on Computing **5** (1993) https://doi.org/10.1287/ijoc.5.4.374

[14] Raudys, A., PABARSKITE, Z.: Optimising the smoothness and accuracy of moving average for stock price data. Technological and Economic Development of Economy **24** (2018) https://doi.org/10.3846/20294913.2016.1216906

[15] Nanayakkara, S., Chandrasekara, V., Jayasundara, M.: Forecasting exchange rates using time series and neural network approaches. European International Journal of Science and Technology **3** (2014)

[16] Kalpakis, K., Gada, D., Puttagunta, V.: Distance measures for effective clustering of arima time-series. (2001). https://doi.org/10.1109/ICDM.2001.989529

[17] Stoean, C., Paja, W., Stoean, R., Sandita, A.: Deep architectures for long-term stock price prediction with a heuristic-based strategy for trading simulations.

PLOS ONE **14** (2019) https://doi.org/10.1371/journal.pone.0223593

[18] Kongcharoen, C., Kruangpradit, T.: Autoregressive integrated moving average with explanatory variable (arimax) model for thailand export. (2013)

[19] Anggraini, W., Vinarti, R., Kurniawati, Y.: Performance comparisons between arima and arimax method in moslem kids clothes demand forecasting: Case study. Procedia Computer Science **72** (2015) https://doi.org/10.1016/j.procs.2015.12.172

[20] Mehtab, S., Sen, J., Dasgupta, S.: Robust analysis of stock price time series using cnn and lstm-based deep learning models. (2020). https://doi.org/10.1109/ICECA49313.2020.9297652

[21] Zhang, P.: Zhang, g.p.: Time series forecasting using a hybrid arima and neural network model. neurocomputing 50, 159-175. Neurocomputing **50** (2003) https://doi.org/10.1016/S0925-2312(01)00702-0

[22] Siami Namini, S., Tavakoli, N., Siami Namin, A.: A comparative analysis of forecasting financial time series using arima, lstm, and bilstm (2019)

[23] Hwarng, H.: Insights into neural-network forecasting of time series corresponding to arma(p,q) structures. Omega **29** (2001) https://doi.org/10.1016/S0305-0483(01)00022-6

[24] Salman, A., Heryadi, Y., Abdurahman, E., Suparta, W.: Weather forecasting using merged long short-term memory model (lstm) and autoregressive integrated moving average (arima) model. Journal of Computer Science **14** (2018) https://doi.org/10.3844/jcssp.2018.930.938

[25] Johnston, F., Boyland, J., Meadows, M., Shale, E.: Some properties of a simple moving average when applied to forecasting a time series. Journal of The Operational Research Society - J OPER RES SOC **50** (1999) https://doi.org/10.1057/palgrave.jors.2600823

[26] Zhuangt, Y., Chen, L., Wang, X., Lian, J.: A weighted moving average-based approach for cleaning sensor data. https://doi.org/10.1109/ICDCS.2007.83

[27] Lawrance, A., Lewis, P.: An exponential moving-average sequence and point process (ema1). Journal of Applied Probability **14** (1977) https://doi.org/10.1017/S0021900200104693

[28] Holt, C.: Forecasting seasonals and trends by exponential weighted moving averages. International Journal of Forecasting **20**, 5–10 (2004) https://doi.org/10.1016/j.ijforecast.2003.09.015

[29] Wang, F., Xuan, Z., Zhen, Z., Li, K., Wang, T., Shi, M.: A day-ahead pv

power forecasting method based on lstm-rnn model and time correlation modification under partial daily pattern prediction framework. Energy Conversion and Management **212** (2020) https://doi.org/10.1016/j.enconman.2020.112766

[30] Pomorski, P., Gorse, D.: Improving on the Markov-Switching Regression Model by the Use of an Adaptive Moving Average, (2023). https://doi.org/10.1007/978-3-031-23844-4_2

[31] Aimran, A.: A comparison between single exponential smoothing (ses), double exponential smoothing (des), holt's (brown) and adaptive response rate exponential smoothing (arres) techniques in forecasting malaysia population. Global Journal of Mathematical Analysis **2** https://doi.org/10.14419/gjma.v2i4.3253

[32] Alevizakos, V., Chatterjee, K., Koukouvinos, C.: A nonparametric triple exponentially weighted moving average sign control chart. Quality and Reliability Engineering **37** (2020) https://doi.org/10.1002/qre.2810

[33] Peleg, R., Weiss, R., Hoogi, A.: Leveraging the triple exponential moving average for fast-adaptive moment estimation (2023)

[34] Hilborn, R., Amoroso, R., Anderson, C., Baum, J., Branch, T., Costello, C., Moor, C., Faraj, A., Hively, D., Jensen, O., Kurota, H., Little, L., Mace, P., Mcclanahan, T., Melnychuk, M., Minto, C., Osio, G., Parma, A., Pons, M., Ye, Y.: Effective fisheries management instrumental in improving fish stock status. Proceedings of the National Academy of Sciences **117**, 201909726 (2020) https://doi.org/10.1073/pnas.1909726116