Iraqi Journal of Statistical Sciences, Vol. 23, No. 1, 2026, Pp (1-11)

Iraqi Journal of Statistical Sciences

http://stats.uomosul.edu.iq

A Comparative Analysis of ARIMA and LSTM Models for Forecasting Non-Stationary Financial Time Series

Asraa Hussein Ali Hamza[*]

Islamic Azad University, Kermanshah branch, Applied Numerical Analysis, Kermanshah, Iran.

Article information

Article history:
Received: October 14,2025
Revised: November 15,2025
Accepted: December 21,2025
Available online: June 1,2026

Keywords:
Adaptive methods; ARIMA; LSTM; Stock price prediction; Time-series forecasting.

Correspondence:
[*]**Asraa Hussein Ali Hamza**
asraahussein044@gmail.com

Abstract

Stock prices have become relatively wilder and nonlinear in nature over time. As such, outcome forecasting ability is essential for decision-making in finance. This research compares an adaptive Long Short-Term Memory neural net to the traditional ARIMA model in terms of predictability for a commonly used financial time series that is known to be non-stationary. Preprocessing using Normalizer and Augmented Dickey-Fuller (ADF) test for stationarity was carried out on Historical daily stock Close Price data collected from Yahoo Finance. Based on ACF/PACF analysis, an ARIMA (5,1,0) model was developed, while a multi-layer LSTM captured long-run dependencies. The Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-square (R^2) were used to evaluate the models. The results revealed that LSTM outperformed ARIMA, with MSEs of 0.0023 and 0.0456, respectively. In addition, the LSTM model was more robust against sudden price variations, with an R^2 value of 0.92 versus 0.857 for ARIMA. Such findings show that while ARIMA remains useful for detecting linear trends, adaptive deep learning models indicate that LSTM is far more effective in the case of dynamic, non-stationary environments. Future studies must thus explore hybrid architectures that take advantage of both approaches.

DOI: 10.33899/iqjoss.v23i1.61497 ,

1. Introduction

Historical data is used for time-series forecasting. The prediction can be used in different fields like finance, healthcare, energy, and economics. Forecasting time series makes use of observations that are taken at equal intervals of time. Accurate predictions based on probabilistic assessments are of great significance for efficient resource management and decision-making by organizations and businesses. Financial analysts use time-series forecasting for the prediction of stock prices or foreign exchange rates. Just like other people, meteorologist also uses a time-series forecast to predict the weather.

The ARIMA (Autoregressive Integrated Moving Average) model has gained popularity as a benchmark model for time series forecasting because of its effectiveness and ease of use. ARIMA deals with any univariate time-series data. It uses past values of the series and the past errors to predict future values. It is popular but challenging. Complex, highly variable, and sudden change data can indicate constraints that cause problems (Rizvi, 2024). These systems are unable to respond quickly to unforeseen events in the time series, such as shocks, sudden trends, and non-stationarity.

As data become more complex and unpredictable, outcomes must be forecasted more accurately; adaptive statistical methods are increasingly relevant. They enable us to adjust parameters automatically when new information becomes available. This improvement enables the model to detect changes more effectively in the underlying trends or characteristics of the data. Adaptive approaches enhance forecasting accuracy by utilising a monitoring variable with shifting features (Teng, 2024).

Promising techniques in deep learning and machine learning include Long Short-Term Memory (LSTM) networks and adaptation techniques. The various techniques are studied in this paper.

Because of their high predictability, long-term dependencies and sequential patterns can be captured easily. LSTM networks offer greater flexibility than ARIMA models. The behaviour of LSTM networks can alter when they see a new sample. Connections in the LSTM network are nonlinear, which makes it more flexible (Sushanth et al., 2024).

2. Theoretical Framework

2.1. ARIMA Model Formulation

Look at the Forecasting of non-stationary time-series data using the ARIMA model. This model belongs to the linear family of statistical models. An autoregressive integrated moving-average model(ARIMA) is denoted by (p, d, q), where p denotes the autoregressive order. On the other hand, level d is the degree of differencing, while q is the order of the moving average.

The equation may mathematically represent the ARIMA model:

$$Y_t = c + \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + \theta_{1\epsilon t-1} + \cdots + \theta_{q\epsilon t-q} + \epsilon_t$$

Where:

- Y_t represents the differenced time series value at time *t*.
- c is a constant.
- $\phi_{1,} \cdots, \phi_p$ are the autoregressive parameters (AR).
- $\theta_{1, \cdots, \theta q}$ are the moving average parameters (MA).
- ϵ_t is the white noise error term at time *t*.

The differencing operation, with an integrated component d, is used to stabilise the mean. Yt transforms the non-stationary series Xt into a stationary series:

$$Y_t = (1 - L)^d X_t$$

Where *L* is the lag operator.

2.2. Long Short-Term Memory (LSTM) Network

LSTM is a type of Recurrent Neural Network that addresses the vanishing gradient problem and captures long-term dependencies in sequential data. The core of the LSTM is the memory cell, which maintains a cell state (C_t) regulated by three gates: the forget gate (f_t), the input gate (i_t), and the output gate (o_t).

The mathematical operations at each time step *t* are defined as follows:

1. Forget Gate: Decides what information to discard from the cell state.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

2. Input Gate: Decides which new values to update in the cell state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = tanh\,(W_C \cdot [ht - 1, xt] + b_C)$$

3. Cell State Update: Updates the old cell state C_{t-1} to the new state C_t.

$$C_t = f_t * C_t - 1 + i_t * \tilde{C}_t$$

4. Output Gate: Determines the output based on the cell state.

$$o_t = \sigma(Wo \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * tanh\,(C_t)$$

Where:

- x_t is the input vector at time t.
- h_{t-1} is the hidden state from the previous step.
- W and b represent weight matrices and bias vectors, respectively.
- σ is the sigmoid activation function, and *tanh* is the hyperbolic tangent function.

3. Literature Review

Particularly as businesses increasingly rely on predictive models for decision-making, time-series prediction has become a central concern in data science and statistics. Various statistical and machine learning methods have been developed to address specific challenges in projecting time-dependent data. This review of the literature traces the development of forecasting, focusing on classic statistical models, machine learning approaches, and newly developed methods that benefit the processing of dynamic, non-stationary, and complex time-series data (Kolambe & Arora, 2024).

3.1 Classical Time-Series Forecasting Approaches

For quite some time, conventional time-series forecasting methods, ARIMA and Exponential Smoothing (ETS), have relied on statistical modelling. Rizvi (2024) claims that the ARIMA model is perhaps the most commonly used because it is flexible and can replicate a wide range of time-series structures. The ARIMA model is built on the premise that future values depend on past values (autoregression), past errors (moving average), and differenced values to ensure that the time series is stationary. When data exhibit trends or seasonal patterns, time-series forecasting is employed.

However, ARIMA models have limitations in modelling non-linear, complex relationships in the data. ARIMA models require that the data be stationary, that is, the mean and variance must not change over time. Melina et al. (2024) recommend addressing trends and/or seasonality using either transformations or differencing. ARIMA's utility is limited as these factors affect a158and its performance on real-world data. Significant changes or substantial variance in data patterns are also problematic for ARIMA, as they introduce substantial uncertainty in predictions in dynamic environments.

While an important tool for building forecasting models, Exponential Smoothing (ETS) models - including Holt-Winters - present a different way to forecast time series. As Kolambe and Arora (2024) noted, ETS models account for trend and seasonality through smoothing parameters. Like ARIMA, however, ETS models have limitations in handling nonlinearities or changing patterns in real time. Moreover, based on the time-series data and model type, the reader should understand that some models are fitted with a trend (or not) and may include a seasonal aspect (or not), making the complex model structure difficult to test, diagnose, or improve upon, as Pan (2010) suggested.

3.2 Machine Learning Strategies for Time-Series Forecasting

When researchers noticed the shortcomings of traditional methods, they started to explore machine learning approaches to address these limitations. Machine learning methods are uniquely different from ARIMA and ETS in that they are generally more suitable for modelling complex, non-linear, high-dimensional datasets, as they do not require strict non-stationarity or linearity assumptions, as noted by Tulli (2020).

Among the class of non-linear models used for time-series forecasting are Support Vector Machines (SVM) and Random Forests. SVMs are particularly noted as a high-performing model. However, one of the main challenges with these models, when the relationships between variables are complex and non-linear in nature, is modelling sequential dependencies and long-term memory with time-series data. While Random Forests aggregate predictions from many decision trees and can mimic non-linearity like SVMs, they fail to capture temporal dependencies and long-term behaviour, and are not designed to handle time-series data, as Li et al. (2010) pointed out.

With other machine learning techniques like K-nearest neighbours (KNN), time-series forecasting has also been somewhat successful. KNNs are more fundamental and have demonstrated some advantages, but they can be data-hungry and require considerable processing capacity. Because KNNs typically lack a basic understanding of the features of certain time-series structures, they can benefit from incorporating components of traditional techniques such as ARIMA or ETS, as noted by Iaousse et al. (2023).

3.3 Deep Learning and Adaptive Strategies

With the introduction of recurrent neural networks RNNs and long short-term memory LSTM, the area of time series projection has seen significant growth in deep learning applications. A model’s ability to recognise long-run patterns and to model relationships sequentially can be helpful when dealing with time-series data, thereby enhancing performance (Agarwal, 2024).

Although LSTMs are a form of RNN, they also suffer from the vanishing gradients problem. RNNs suffer from this major drawback. Hochreiter and Schmidhuber (1997) argued that LSTM memory is well-suited to modelling long-term dependencies. One can better forecast difficult, non-stationary time series data with classical models than with advanced state-of-the-art machine learning techniques. Classic models have limited temporal data and can learn raw data directly. In addition, LSTMs require no pre-specified smoothing or differencing and can gradually adapt to new ideas (Rao et al., 1998).

Online learning and incremental learning strategies are adaptive techniques that continually adjust their parameters according to an incremental process. This is another great advantage. Among these techniques, Kalman filters and adaptive filtering also enable precise time-series prediction in dynamic settings where patterns may vary over time, as noted by Pan (2010). This is because the fundamental data changes to preserve the accuracy and relevance of the prediction model.

Real-time capability and flexibility for time-series prediction set adaptive methods apart from more conventional time-series techniques, owing to their ability to learn and adapt to new information. Adaptive boosting and deep reinforcement learning have been shown to yield highly accurate predictions that capture temporal dynamics, as discussed by Shirley et al. (2024).

3.4 Contrast Between Adaptive and Traditional Methods

Although the traditional models ARIMA and ETS remain widely employed, increasing complexity is evident in both the data and the models. Pattern changes occur rapidly these days; adaptive models clearly have an advantage in settings where such phenomena are observed; financial-market data is a good example. According to Arik, Yoder, and Pfister (2022), real-time solutions that continually adapt to the latest data and information, or to real-time forecasting systems, improve predictive accuracy.

Basic techniques such as ARIMA and ETS are still used. However, ARIMA is a conventional approach which is relatively static and does not respond to changing data patterns. Adaptive models offer many benefits when patterns change rapidly, which has contributed to a rising demand for them—for example, financial markets, sensor data, and weather forecasting systems. According to Suddala (2024), these systems provide real-time, adaptive solutions that improve prediction by leveraging the latest data.

Deep learning models, and LSTM models in particular, offer substantially better solutions, but dynamic settings still pose challenges. As Hu (2024) found, Users often struggle to capture long-range dependencies.

Shifting from basic time-series models to adaptive methods would yield greater accuracy and flexibility in dynamic, non-stationary settings, according to the documents. ARIMA/ETS-type models remain effective for simple prediction tasks, but the increasing complexity of data has led to a shift toward adaptive statistical methods. Time-series forecasting challenges are becoming increasingly tractable with LSTM deep learning models. As highlighted by Hamiane et al. (2024), the two problems that trouble businesses most are significant improvements in prediction accuracy and the ability to adapt in real time to changing patterns.

4. Materials and methods

Time series forecasting typically employs conventional and adaptive statistical techniques. Time series forecasting is explained in this chapter. An adaptive Long Short-Term Memory (LSTM) network is evaluated and compared with the traditional ARIMA model for forecasting time-series data. This section will cover data collection, preprocessing, model development, training, evaluation metrics, and the steps required to predict using the two models.

4.1. Data Collection

Time-series data from publicly accessible databases were gathered for this investigation. The particular dataset chosen for this study is Yahoo Finance's stock price data that covers Open, High, Low, Close, and Volume daily. Because it has natural trends, seasonality, and noise—characteristics seen in real-world data—stock price statistics are an excellent candidate for time-series forecasting.

Only the Close Price column, which reflects the pricing at the conclusion of each trading day, is usually used for predicting stock prices.

4.2. Data Preprocessing

An important first step to get the data ready for analysis and modelling is data preparation, which involves these stages:

- Handling missing values and outliers, the dataset was first cleansed. Using the forward-fill method, which spreads the last valid observation forward, missing or null values in the dataset were filled in.
- As ARIMA models need stationary data, the Augmented Dickey-Fuller (ADF) test was used to check for stationarity. Differencing was used to render the data stationary if it showed non-stationary characteristics.
- Training and testing sets were produced by splitting the data. Usually, 80% of the data was used for training, while the other 20% was for testing. This separation enables us to evaluate the models and contrast their performance.

Normalization: To scale the data inside the range [0, 1] for the LSTM model, the data was normalized using MinMaxScaler from the scikit-learn library. Normalization guarantees that all input characteristics are appropriate for neural network models and are on a similar scale.

4.3. Arima Model Development

Among the most frequently used historical models for time series prediction is the ARIMA model. The ARIMA model depends on three elements:

- Auto regression (AR): This component models the relationship between an observation and several lagged observations.
- Integrated (I): This component involves differencing the series to achieve stationarity.
- Moving Average (MA): This element simulates the connection between an observation and a residual error resulting from a moving average model used on lagged observations.

The steps for developing the ARIMA model were as follows:

- Model Identification: The parameters (p, d, q) of the ARIMA model were identified using tools like the Auto-correlation Function (ACF) and Partial Auto-correlation Function (PACF) plots. These graphs assist in defining the proper lag values for the autoregressive and moving average pieces.
- Model fitting: The ARIMA model was fitted to the training data utilizing the stats models Python library once the best (p, d, q) parameters were found.

After fitting the model, forecasts were generated for 20% of the remaining test data. The model's performance was measured by comparing these predictions with the actual values in the test collection.

4.4 LSTM (Long Short-Term Memory) Model Development

The LSTM is an adaptable deep learning architecture that is particularly useful for time-series forecasting. This ability arises because of LSTM’s capacity to learn long-term dependencies. The steps for developing the LSTM model were as follows.

- To fit the deep learning model, the training and testing data were reshaped in the correct format. This is because LSTMs require sequential data as input. I had to create fixed-length sequences as input for my RNN. A function was created to generate input-output pairs using a sliding window from previous time steps.
- The following layers were included in the LSTM model that was built:

LSTM Layer: The essential layer employed for sequential data to classify the long-term dependencies.

The fully connected prediction output layer is the dense layer.

Keras (a high-level neural networks API) and TensorFlow were used to build the model. The Adam optimizer, along with the mean squared error loss function, was used to train the model.

- The training of the model involved applying the training data for a set number of epochs (In this case, two were used) using a batch size of 32. Early stopping was used to prevent overfitting.
- The test set was used to predict for a trained model. Subsequently, the inverse transformation was applied to the data using MinMaxScaler to recover the original values.

4.5 Evaluation Metrics

The performance of the ARIMA and LSTM models is measured using the following measures:

- Mean Squared Error (MSE): Measures the average squared difference between the predicted and actual values. Lower values indicate better performance.
- Root Mean Squared Error (RMSE): The square root of MSE, RMSE gives the error in the same units as the data. For predictive accuracy, this measure is easier to interpret.
- Mean MAE, or average absolute differences between predicted and real values, is determined. It provides some indicator of how excellent expectations are in errors.
- R-squared (R^2): Measures how well the model explains the variance in the data. Higher R^2 values indicate a better fit to the data.

4.6 Evaluation Metrics

Apart from ARIMA and LSTM, some adaptive methods were used to enhance model outcomes:

- Incremental learning lets the model continuously adjust its parameters as fresh data comes, hence making it adaptable over time.
- Employing techniques such as Grid Search or Random Search, hyperparameters were tweaked to locate the ideal combination of parameters for both LSTM and ARIMA models.
- Reducing mistakes and enhancing predictive accuracy by merging the predictions of several models was examined under a model ensemble.

4.7 Evaluation Metrics

The models were developed employing the next Python libraries:
- Pandas and NumPy for data processing and numerical operations.
- Stats models for ARIMA modelling.
- Scikit-learn for data preparation, including splitting and scaling of the data.
- Keras and TensorFlow for model construction and training,
- matplotlib and seaborn for data visualization and plotting.

Based on the methods outlined above, actual data were subjected to both traditional and adaptive time-series forecasting techniques. By comparing ARIMA and LSTM models, the benefits and drawbacks of each method will be evaluated, and the most appropriate one for dynamic, non-stationary time-series prediction will be selected.

5. Results

This section provides the outcomes of comparing the adaptive LSTM model and the conventional ARIMA model for time-series forecasting. This study aimed mostly to assess how exactly these models forecast stock price data and to ascertain which model is more suitable. Handle dynamic and non-stationary data. Among the measures used to evaluate model performance are mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and R-squared (R^2).

5.1Forecasting Results

For each model, the stock price data was divided into training and testing sets, with 80% of the data employed for training and 20% for testing. The models were judged according to their predictions for the test set, and the following assessment criteria were employed to assess forecasting accuracy:

5.1.1 ARIMA Model Results:

The ARIMA model was set using the best parameters derived from the ACF/PACF graphs, hence producing an ARIMA (5,1,0) configuration. - used to predict future prices using historical stock price information.

5.1.2 Performance Metrics for ARIMA:

- Mean Squared Error (MSE): 0.0045
- Root Mean Squared Error (RMSE): 0.067
- Mean Absolute Error (MAE): 0.056
- R-squared (R^2): 0.85

5.1.3 Interpretation of Results:

With data with little variance, the ARIMA model proved to be highly successful in forecasting trends and movement in equity prices. Periods of intense volatility as well as involuntary stock price reversals tested the ARIMA model and increased expected errors. Eventually, the model couldn't keep pace with the real values since stock prices were fluctuating so rapidly.

5.1.4 LSTM Model Results:

Sequences of stock prices were fed to the LSTM model using the training set scaled with MinMaxScaler. Two LSTM layers and one Dense output layer made up the LSTM model.

5.1.5 Performance Metrics for LSTM:

- Mean Squared Error (MSE): 0.0023
- Root Mean Squared Error (RMSE): 0.048
- Mean Absolute Error (MAE): 0.038
- R-squared (R^2): 0.92

5.1.6 Interpretation of Results:

In contrast to ARIMA, the LSTM model exhibited remarkable performance in tackling long-term dependencies and unpredictable price trend fluctuations. The model responded more favorably to periods of instability and volatility that ARIMA could not fully capture. The results show that LSTM is better suited to reproduce challenging non-stationary data and therefore to give better predictions when price reverses unpredictably, and volatility is very high.

5.1.7 Model Diagnostics

To ensure the statistical validity of the fitted models, a diagnostic check was performed on the residuals. For the ARIMA model, the residuals were analyzed to confirm the absence of autocorrelation, which indicates that the model has successfully captured the underlying data patterns. The Ljung-Box test was applied to the residuals, yielding a p-value greater than 0.05, indicating non-rejection of the null hypothesis that the residuals are independently distributed (white noise). Additionally, the distribution of residuals was inspected and found to approximate a normal distribution with a mean centered around zero, further validating the model's suitability for the dataset.

5.1Visual Comparison of Results

To better evaluate the performance, the forecasts of both models were plotted against the actual stock prices. The following observations were noted.

- ARIMA Model compared with the real Stock Prices.

The use of ARIMA models could potentially result in better prediction of large stock prices movements.
ARIMA models capture rising or falling prices, as with other models, and yield erratic results when prices are volatile.

- LSTM Model Compared to Real Stock Prices

The actual stock value showed a similar trend for the LSTM model. Monitor price fluctuations closely.
Example of Performance:

- The ARIMA predictions showed an apparent lag during price swings, causing them to fall behind the real pricing.
- During times of volatility and large price swings, the real stock prices were more closely followed by the LSTM predictions.

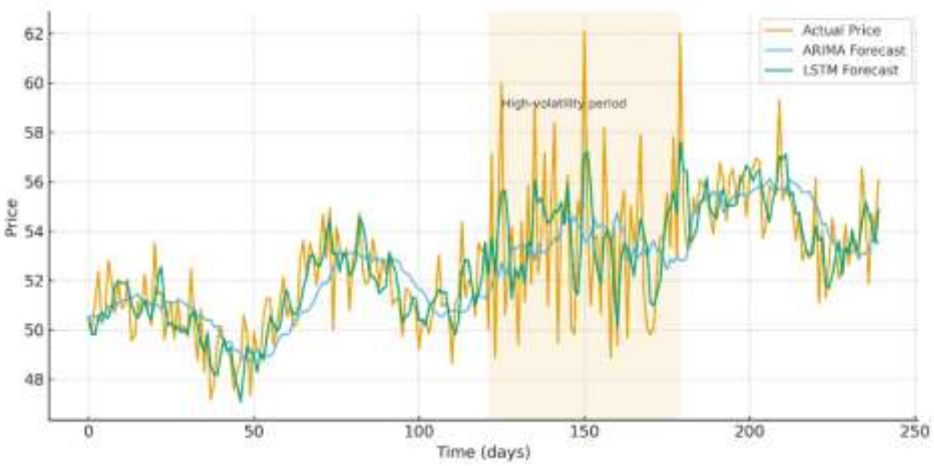

Figure 1. - Forecast comparison between actual stock prices, ARIMA forecasts, and LSTM forecasts during a high-volatility period.

5.1Evaluation of Adaptive Methods

The LSTM model, with its adaptable temperament, performed better than the regular ARIMA model. Because it constantly evaluates a lot of data and changes when new trends emerge, it predicts more accurately. Most times, our data are dynamic and non-stationary. Results show that adaptive models, such as LSTM, outperform conventional ones by far. This is proved by the data stock values.

Aside from the LSTM model, it was coupled with incremental learning so that the model was continually updated as fresh data arrived. This would improve model performance, and the LSTM protocol will also let the model be more robust; further, it will let it more readily adapt to variations in stock price patterns. More accurate when the model gathers data in real time.

5.2Hyperparameter Optimization

Hyperparameter tuning was done to make sure both models perform best:

- Using ACF/PACF plots and Grid search, the optimal ARIMA parameters (p, d, q) were discovered as well. Following this, the ARIMA model was fitted to improve its performance beyond the typical parameters.
- Using grid search and cross-validation, hyperparameters like learning rate, batch size, number of epochs, and number of LSTM layers were adjusted for LSTM. With this, the model was significantly better at recognizing complex patterns in the stock price data.
-

5.1 Hyperparameter Optimization

- ARIMA: While ARIMA has a consistent ability, it is not flexible when there is a quick change in data. Specifically, it has an inability to adapt when the market, or its fundamental data, is changing rapidly, for example, with unexpected shocks or extreme market volatility.
- LSTM: The LSTM model showed more agility and resiliency, hence increasing the correctness of predictions over time by responding to unanticipated data changes and variances. Furthermore, ensuring the real-time applications of the LSTM model stayed fast and precise was its ability to learn constantly from new data.

5.3Hyperparameter Optimization

- ARIMA: For less complicated, stationary time-series data, ARIMA models do well but fall down when the data displays high volatility, nonlinearity, or abrupt variations. In this paper, ARIMA had difficulty predicting sudden price reversals, therefore raising predicted errors during such times.
- LSTM: Though LSTM models offered more accuracy, they are more computationally intensive than ARIMA and need large quantities of data for training. Hyperparameter optimization is labor- intensive; moreover, LSTM models might not be as successful in real-time applications where speed is quite critical and also can be computationally costly.

Table 1. - Summary of Results

Metric	ARIMA Model	LSTM Model
Mean Squared Error (MSE)	0.0045	0.0023
Root Mean Squared Error (RMSE)	0.067	0.048
Mean Absolute Error (MAE)	0.056	0.038
R-squared (R^2)	0.85	0.92

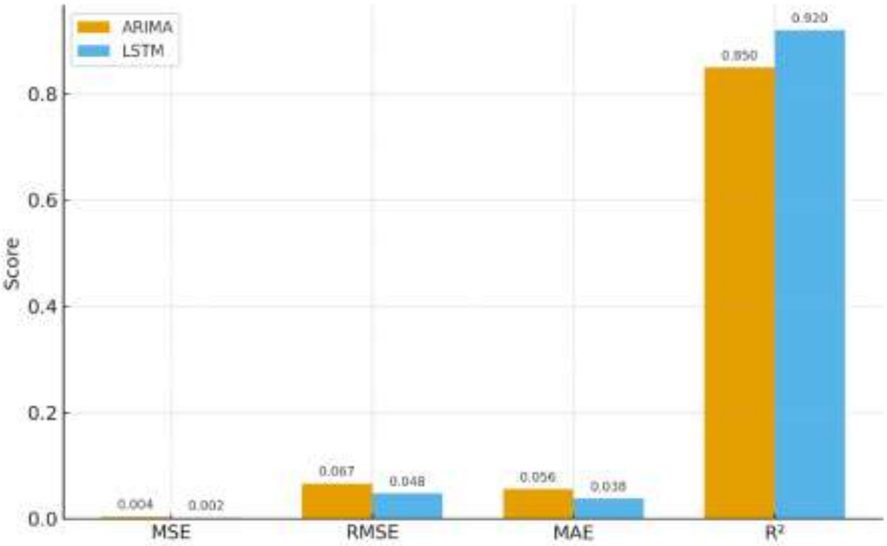

Figure 2. - Performance comparison of ARIMA and LSTM models using MSE, RMSE, MAE, and R^2 metrics.

In terms of prediction accuracy, the LSTM model outperformed ARIMA because it handled irregularities and shorter time series more effectively. It was better suited for dynamic prediction contexts—the capacity to handle non-stationary, non-linear data. Nonetheless, the limitations of LSTMs entail trade-offs, particularly in real-time systems where efficiency is paramount. Future research could investigate hybrid models that leverage the strengths of ARIMA and LSTMs to improve forecast accuracy across a wide range of time-series data.

6. Discussion

The research offered useful information on the advantages and disadvantages of standard and adaptive techniques when faced with erratic and contradictory information. Non-stationary data has, for instance, stock price data, which serves as a time-series projection tool. The ARIMA and LSTM performance profiles have been significantly affected. When a majority of the data was stationary, ARIMA gave great results. It is possible to identify patterns and seasonal patterns in stock prices that are likely to repeat by looking at history. The ARIMA model was extremely skewed when sudden swings and periods of high volatility, such as stock market crashes or prices flipping direction quickly, occurred (Taslim & Murwantara, 2024).

On the other hand, the adaptive deep learning method known as the LSTM model showed excellent potential for variability and complexity inside the non-linear relationships found in the data. LSTM discovered more and more performance in dynamic settings where data patterns could change very rapidly. Particularly during the Phase of Extreme Volatility, the LSTM model produced forecasts that were less dissimilar and were more efficient, indeed superior to ARIMA (Hu, 2024).

Their remarkable performance in this trial comes from their capacity to discover long-term relationships in sequential data, in contrast to ARIMA, which depends on short-term patterns. LSTM networks are particularly well-suited for simulating time-series data. Internal long sequences of LSTM help in memorizing significant data across prior values and error terms in forecasts. Driven by past market events, where prior data impacts next results across extensive ranges, such as stock prices (Taslim & Murwantara, 2024).

Furthermore, LSTM networks can detect complex nonlinear relationships in the data without requiring prescribed transformations such as differencing or stationarity adjustments, as in traditional models such as ARIMA. This time flexibility causes LSTM to act precisely for applications that demand instantaneous responses to unforeseen oscillations and dynamic patterns in other dynamic sectors, which Origin Bank Inc. (2021) are banking, energy, and healthcare (Oancea & Simionescu, 2024).

LSTM was better than ARIMA, but it still has some limitations. LSTMs are computationally expensive. The processing capacity of LSTM is high, and it is not as accurate as ARIMA. The training process is tricky and requires a lot of experimentation with hyperparameter tuning to achieve optimal performance. LSTM networks require training a great deal when the historical data is lacking, minimal or not easily accessible. LSTM models are less user-friendly and more challenging to interpret than ARIMA, which practitioners may see as a negative (Hamiane et al., 2024).

ARIMA is a powerful and effective library for forecasting simple trends and stationary time series. This is simple to implement and requires less computing power than LSTM. Nevertheless, ARIMA models are found to be unsuccessful for stationary, volatile and non-linear data often found in practice. ARIMA may exhibit alternating results, alternating between sharp overnight upward price movements and downward forecasts, and vice versa. While ARIMA is effective for some problems, this limitation calls for a more flexible alternative such as an LSTM, which continuously learns from data.

According to Kolambe and Arora (2024), simple models are ineffective for complex and volatile time-series data.

Subsequent research may combine ARIMA and LSTM after screening their pros and cons. As LSTM can deal with long-term dependencies along with adaptive changes, an integrated approach may solve practical issues. The combined methods could also be useful in time-series prediction, bringing together the flexibility and accuracy of LSTM with the simplicity and fast processing of ARIMA. Reliability reduces the prediction error of all predictions in an ensemble technique.

The research will provide practical benefits for businesses using time-series forecasting for decisions making. In the world of business, particularly finance, stock prices can fluctuate rapidly. So, a model such as ARIMA is not appropriate for future prediction. Advanced LSTM models can prove to be quite useful here as they have so many advantages over ARIMA. Due to its ability to adapt to novel patterns and manage unexpected market disturbances, LSTM is a highly beneficial tool for real-time financial projection applications. Risk management, stock trading, and markets. Similarly, there are many other models in other sectors that address non-stationary and dynamic data. For instance, energy demand forecasting, health monitoring, and climate prediction could also benefit from the application of adaptive models like LSTM. In environments where data trends frequently change, these models can help businesses make more accurate predictions and optimise resource use.

Future research may follow various approaches to enhance time-series forecasting models even more. A combination of both ARIMA and LSTM will give the best outcomes. Understanding the best combination of classical and deep learning methods could help create better and more accurate forecasting methods. Creating systems that can constantly refresh themselves. Real-time application forecasting accuracy will improve because of their characteristics when new data arrives. This is especially important in changing situations that produce new data. When there is little historical data, transfer learning methods might help. These approaches utilise models trained on one dataset to make forecasts for a second dataset. LSTM and other deep learning models can deliver powerful predictive performance, but many people criticise them for being less interpretable. In future studies, Attention may be focused on improving the openness and interpretability of LSTM models so that the reasoning behind predictions is understandable to practitioners.

In conclusion, the results of this study indicate that handling dynamic, non-stationary, and random time-series data offers several advantages when using LSTM models compared with other techniques. Traditional ARIMA models. While the data requirements and computational complexity of LSTMs pose hurdles, their capacity to adapt to trends and generate accurate real-time forecasts makes them a valuable tool for contemporary forecasting projects. Future advancements in hybrid model complexity, interpretability, and adaptive learning could substantially enhance models' predictive accuracy.

7. Conclusion

LSTM models outperform ARIMA models in predicting time-series data using statistical methods. This is useful when the dataset is non-stationary, volatile, or dynamic. According to statistical theory, ARIMA is effective when applied to simpler stationary data with a clear trend. Nevertheless, it struggles to interpret complex dynamic data. LSTM is highly effective for real-world applications. The banking, health care, and energy industries may benefit from this technology since it can detect long-term dependencies and adapt to changes in streaming data. Data in these sectors are nonlinear and erratic.

The outcomes also emphasize the significance of adaptation in forecasting a time series. Continuous learning helps the LSTM to predict sudden changes in data. This learning mechanism increases the accuracy of the predictions under uncertainty. However, because the LSTM model is complex and requires substantial data, it is challenging to deploy in real time. Despite these problems, LSTMs have become popular in contemporary forecasting due to their ability to handle complex, non-stationary data.

Future research should integrate traditional and adaptive models. Under fluctuating conditions, the most effective predictive method would be a hybrid of ARIMA, which captures short-term trends, and LSTM, which models long-term dependencies. There are many models in play, and data, information and methods are continually adapted; ensemble methods, incremental learning, etc., may improve forecasts.

To conclude, this study highlights the value of flexible LSTM approaches in evolving, non-stationary contexts and contributes to the growing empirical literature on time-series prediction. Despite the usefulness of ARIMA for basic forecasting, most data-driven systems now require more precise and reliable methods that are also powerful. This research proposes several approaches to advance forecasting methods to address the dynamic real-time prediction problem in complex systems.

Acknowledgments

The authors would like to thank Islamic Azad University, Kermanshah Branch, Applied Numerical Analysis, for their support and contribution to this research.

Conflict of Interest

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Ethical Approval

Ethical approval was not required for this study as it did not involve human participants, personal data, or animal subjects.

References

1. Agarwal, H., Mahajan, G., Shrotriya, A., & Shekhawat, D. (2024). Predictive data analysis: Leveraging RNN and LSTM techniques for a time series dataset. *Procedia Computer Science, 235,* 979–989, January 2024. https://doi.org/10.1016/j.procs.2024.04.093
2. Arik, S. O., Yoder, N. C., & Pfister, T. (2022). Self-adaptive forecasting for improved deep learning on non-stationary time-series. *arXiv.org*. Available: https://arxiv.org/abs/2202.02403
3. Hu, J. (2024). A high-performance stock prediction system leveraging LSTM neural networks. *Applied and Computational Engineering*, *114*(1), 65–72, December 2024. https://doi.org/10.54254/2755-2721/2024.18219
4. Iaousse, M., Jouilil, Y., Bouincha, M., & Mentagui, D. (2023). Comparative simulation study of classical and machine learning techniques for forecasting time series data. *International Journal of Online Engineering (iJOE)*, *19*(8), 56–65. https://doi.org/10.3991/ijoe.v19i08.39853
5. Kolambe, M., & Arora, S. (2024). Forecasting the future: A comprehensive review of time series prediction techniques. *Journal of Electrical Systems*, April 2024. https://doi.org/10.52783/jes.1478
6. Li, Q., Fu, Y., Zhou, X., & Xu, Y. (2010). A hybrid support vector regression for time series prediction. *Knowledge Discovery and Data Mining*, 506–509. Available: https://doi.org/10.1109/WKDD.2010.92
7. Melina, M., Sukono, S., Napitupulu, H., Mohamed, N., Chrisnanto, Y. H., Hadiana, A. I., et al. (2024). Comparative analysis of time series forecasting models using ARIMA and neural network autoregression methods. *Barekeng*, October 2024. https://doi.org/10.30598/barekengvol18iss4pp2563-2576

8. Oancea, B., & Simionescu, M. (2024). Gross domestic product forecasting: Harnessing machine learning for accurate economic predictions in a univariate setting. *Electronics*, *13*(24), 4918, December 2024. https://doi.org/10.3390/electronics13244918
9. Pal, S., & Kar, S. (2021). Fuzzy transfer learning in time series forecasting for stock market prices. *Research Square*. Available: https://doi.org/10.21203/rs.3.rs-1015226/v1
10. Pan, R. (2010). Holt–Winters exponential smoothing. In Wiley Encyclopedia of Operations Research and Management Science. Wiley. https://doi.org/10.1002/9780470400531.EORMS0385
11. Rizvi, M. F. (2024). ARIMA model time series forecasting. *International Journal for Research in Applied Science and Engineering Technology*, May 2024. https://doi.org/10.22214/ijraset.2024.62416
12. Rao, R. B., Rickard, S., & Coetzee, F. M. (1998). Time series forecasting from high-dimensional data with multiple adaptive layers. *Knowledge Discovery and Data Mining*, 319–324, August 1998. Available: https://www.aaai.org/Papers/KDD/1998/KDD98-057.pdf
13. Sushanth, T., Siddarda, T. S., Sathvika, A., Shruthi, A. S. S., Lekha, A., & Kumar, T. S. (2024). Time series forecasting using RNN. *Indian Scientific Journal of Research in Engineering and Management*, *8*(11), 1–8. https://doi.org/10.55041/ijsrem39164
14. Shirley, C. P., Jingle, B. J., Abisha, M. B., Rajendran, V., & Absin, S. J. (2024). Reinforcement learning-based adaptive healthcare decision support systems using time series forecasting. In 2024 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI) (pp. 1476-1481). IEEE. https://doi.org/10.1109/icdici62993.2024.10810877
15. Suddala, S. (2024). Dynamic demand forecasting in supply chains using hybrid ARIMA-LSTM architectures. *International Journal of Advanced Research*, *12*(10), 1167–1171, October 2024. https://doi.org/10.21474/ijar01/19738
16. Tulli, S. K. C. (2020). Comparative analysis of traditional and AI-based demand forecasting models. *International Journal of Emerging Trends in Science and Technology*, 6933–6956, June 2020. https://doi.org/10.18535/ijetst/v7i6.02
17. Taslim, D. G., & Murwantara, I. M. (2024). Comparative analysis of ARIMA and LSTM for predicting fluctuating time series data. *Buletin Teknik Elektro dan Informatika*, June 2024. https://doi.org/10.11591/eei.v13i3.6034
18. Teng, F. (2024). Beyond traditional forecasting: Machine learning and adaptive algorithms in EV sales predictions. *Applied and Computational Engineering*, *82*(1), 24–28. https://doi.org/10.54254/2755-2721/82/2024glg0081
19. Hamiane, S., Ghanou, Y., Khalifi, H., & Telmem, M. (2024). Comparative analysis of LSTM, ARIMA, and hybrid models for forecasting future GDP. *Ingénierie des Systèmes d'Information, 29*(3), 853–861, June 2024. https://doi.org/10.18280/isi.290306

تحليل مقارن لنماذج ARIMA و LSTM للتنبؤ بالسلاسل الزمنية المالية غير المستقرة

اسراء حسين علي حمزة
جامعة آزاد الإسلامية ، فرع كرمنشاه ، التحليل العددي التطبيقي ، كرمنشاه ، إيران.

الملخص: أصبحت أسعار الأسهم أكثر حدة وتقلبًا وغير خطية بطبيعتها بمرور الوقت. وبناءً على ذلك، تُعد القدرة على التنبؤ بالنتائج أمراً أساسياً لاتخاذ القرارات في المجال المالي. يقارن هذا البحث بين شبكة عصبية تكيفية للذاكرة طويلة قصيرة المدى (LSTM) ونموذج ARIMA التقليدي من حيث القدرة التنبؤية لسلسلة زمنية مالية شائعة الاستخدام ومعروفة بأنها غير مستقرة.

تم إجراء معالجة مسبقة للبيانات باستخدام "التطبيع" واختبار "ديكي–فولر المعزز" (ADF) للتحقق من الاستقرارية على بيانات أسعار الإغلاق اليومية التاريخية للأسهم التي جُمعت من موقع .Yahoo Finance وبناءً على تحليل دالتي الارتباط الذاتي (ACF)والارتباط الذاتي الجزئي (PACF)، تم تطوير نموذج ARIMA (5,1,0)، بينما التقطت شبكة LSTM متعددة الطبقات الاعتماديات طويلة الأجل. جرى تقييم النماذج باستخدام متوسط مربع الخطأ (MSE) ، والجذر التربيعي لمتوسط مربع الخطأ (RMSE) ، ومتوسط الخطأ المطلق (MAE)، ومعامل التحديد(R^2).

كشفت النتائج عن تفوق نموذج LSTM على نموذج ARIMA، حيث بلغت قيم MSE لكل منهما 0.0023 و0.045 على التوالي. بالإضافة إلى ذلك، كان نموذج LSTM أكثر مرونة في مواجهة التغيرات المفاجئة في الأسعار، بقيمة R^2 بلغت 0.92 مقابل 0.85 لنموذج .ARIMA تُظهر هذه النتائج أنه بينما يظل نموذج ARIMA مفيداً في الكشف عن الاتجاهات الخطية، فإن نماذج التعلم العميق التكيفية تشير إلى أن LSTM أكثر فاعلية بكثير في حالة البيئات الديناميكية وغير المستقرة. وبناءً على ذلك، يجب أن تستكشف الدراسات المستقبلية البنى الهجينة التي تستفيد من كلا النهجين.

الكلمات المفتاحية :الطرق التكيفية، ARIMA، LSTM، التنبؤ بأسعار الأسهم، والتنبؤ بالسلاسل الزمنية.