INTERNATIONAL JOURNAL OF RESEARCH IN ECONOMICS AND FINANCE, 2025, Vol. 2, No. 8, 233-255. https://doi.org/10.71420/ijref.v2i8.170

Forecasting the Moroccan All-Share Index: A Comparative Study of ARIMA and LSTM Deep Learning Approaches

Soufiane BENBACHIR
Laboratory of Studies and Research in Management Sciences, Faculty of Legal, Economic and Social Sciences Agdal, Mohammed V University of Rabat, Morocco.

Abstract. This study addresses the problem of accurately forecasting financial markets in emerging economies, specifically focusing on the Moroccan All-Share Index (MASI) which lacks dedicated research. Motivated by the need to provide a clear and comprehensive comparison for local and international investors, this paper evaluates two prominent time series models: the classical ARIMA model and the deep learning-based LSTM network. The methodology involves using a time series of daily MASI closing prices from January 4, 2010, to August 8, 2025, which was chronologically split into a training sample (80%) and a test sample (20%). The models' out-of-sample performance was then rigorously evaluated using key error metrics. The results reveal a significant disparity in performance. The LSTM model delivered drastically superior accuracy with an MAE of 178.52 and an RMSE of 272.69, vastly outperforming the ARIMA model which yielded an MAE of 1917.74 and an RMSE of 2751.31. This demonstrates that the LSTM's ability to capture complex, non-linear dependencies is far more effective for forecasting the MASI index than the linear assumptions of the ARIMA model. The study concludes that deep learning methods offer a more reliable approach for financial forecasting, with practical implications for investors who can use LSTMs to make better-informed trading decisions and for analysts who can incorporate these models for more nuanced market surveillance and risk management.

Keywords: *MASI Index, Forecasting, ARIMA Model, LSTM model, Deep Learning, MAE, RMSE.*

1. Introduction

Forecasting financial markets represents a fundamental challenge for investors, policymakers, and researchers, as it equips them with critical insights essential for making informed investment decisions, managing financial risks, and supporting the overall stability of economic systems. Accurate market predictions help to anticipate price movements, allocate resources efficiently, and formulate regulatory policies that mitigate systemic risks. However, the task of forecasting financial time series is notoriously difficult due to the markets' intrinsic characteristics, high volatility, nonlinear dependencies, structural breaks, and evolving dynamics that often defy simple modeling assumptions.

Traditional statistical methods, notably the Autoregressive Integrated Moving Average (ARIMA) model, have long been employed for financial time series forecasting. ARIMA's ability to capture linear relationships, seasonal effects, and short-term dependencies has made it a staple in econometric analysis. Yet, despite its extensive use and interpretability, ARIMA frequently struggles to model the complex, chaotic, and non-stationary features inherent in financial data, especially when market conditions shift abruptly or when long-range dependencies are present.

The rise of artificial intelligence, and in particular deep learning techniques, has transformed the landscape of financial forecasting. Among these, the Long Short-Term Memory (LSTM) neural network, a variant of recurrent neural networks, is distinguished by its capacity to learn long-term temporal dependencies and model nonlinear, non-stationary patterns effectively.

LSTM networks have demonstrated promising results in capturing the subtle and often hidden temporal structures within financial data that traditional models may overlook. This has sparked an active debate in the literature regarding the comparative advantages of statistical models versus deep learning frameworks across different market contexts and asset classes.

Despite the global proliferation of forecasting studies, most focus predominantly on well-established markets in developed economies, leaving emerging markets less explored. These emerging economies often exhibit unique market behaviors influenced by differing regulatory frameworks, market liquidity, investor composition, and macroeconomic conditions. In particular, region-specific indices like the Moroccan All-Share Index (MASI), a pivotal benchmark of the Casablanca Stock Exchange, have received limited scholarly attention in terms of advanced forecasting techniques. This paucity of research leaves investors and policymakers without tailored insights relevant to the Moroccan financial environment.

The importance of this study is therefore twofold. First, it addresses a significant research gap by evaluating the forecasting performance of both traditional and deep learning models on the MASI index, thereby providing empirical evidence from an emerging market context that remains underrepresented in the literature. Second, the outcomes of this research hold practical implications for investors and policymakers alike: improved forecasting accuracy can enhance investment strategies, optimize portfolio management, and strengthen risk mitigation in a market characterized by volatility and evolving dynamics. By bridging the divide between global forecasting methodologies and regional market realities, this study contributes both to academic knowledge and to the practical advancement of financial decision-making in Morocco. The originality of this work lies in its pioneering focus on the Moroccan All-Share Index (MASI) using a comparative framework between ARIMA and LSTM models, an approach rarely applied to North African markets. Its added value stems from providing context-specific insights that enrich the forecasting literature on emerging economies, while offering investors and policymakers actionable evidence to support more resilient financial strategies.

The remainder of the paper is structured as follows: Section 2 reviews the relevant literature; Section 3 presents the data and methodology; Section 4 discusses the empirical results; and Section 5 concludes with key findings and implications.

2. Literature review

This literature review provides a comprehensive, chronologically ordered analysis of research comparing time series forecasting models, with a primary emphasis on the traditional Autoregressive Integrated Moving Average (ARIMA) model and the deep learning-based Long Short-Term Memory (LSTM) network. The review is divided into two sections to distinguish between studies focusing on other markets and those in financial markets, highlighting the diverse performance and applicability of these models across different domains.

a. Non financial Markets

The performance of forecasting models in non-financial contexts often depends on the unique characteristics of the dataset, such as linearity and seasonality.

Elmasdotter and Nyströmer (2018) conducted a comparative study of LSTM and ARIMA for sales forecasting in the retail industry, motivated by a desire to reduce food waste. Their analysis, using RMSE and MAE, revealed no statistically significant difference between the models for a one-day-ahead forecast. However, for a seven-day-ahead prediction, the LSTM model demonstrated a statistically significant higher accuracy, suggesting its superiority for longer forecasting horizons in this domain. Nasser and Etem (2021) also focused on non-financial applications, comparing LSTM and ARIMA for hourly energy consumption forecasting in the western USA. Both models achieved a high performance, with LSTM and ARIMA yielding R-squared metrics of 97% and 98%, respectively.

Moving to the domain of environmental science, Khan (2022) performed a methods case study comparing LSTM and traditional (S)ARIMA models for forecasting indoor air pollution in Canada and Sweden. Khan noted that while (S)ARIMA models are limited to making linear predictions from a single variable, the deep learning LSTM model is capable of producing more precise nonlinear forecasts using multivariate inputs, thereby preventing memory loss.

In the real estate sector, Albeladi et al (2023) compared LSTM and ARIMA for time series forecasting using a dataset from Mulkia Gulf. Contrary to some other findings, their research concluded that the ARIMA model was better suited for this specific type of time series forecasting, based on the mean average of the basic evaluation parameters. Similarly, Trisya et al. (2024) compared the forecasting performance of ARIMA, LSTM, and Support Vector Machine (SVM) models for predicting monthly electricity consumption from 1985–2018. The study found that while ARIMA(1,0,1) provided reasonable forecasts with an RMSE of 7.659, it struggled with high-precision requirements. LSTM handled complex, nonlinear patterns better than ARIMA but had higher RMSE values (best case = 11.4183), while SVM outperformed both with the lowest RMSE (0.020) for short-term predictions. The authors concluded that SVM offered the highest accuracy for short-term forecasts, LSTM was useful for modeling nonlinear dependencies, and ARIMA provided interpretable results for simpler patterns.

Shifting to macroeconomic forecasting, Hamiane et al (2024) conducted a comparative study on forecasting quarterly U.S. GDP data from 1947 to 2022. They examined LSTM, ARIMA, and a hybrid ARIMA–LSTM approach. Their research revealed that the hybrid model delivered the highest predictive accuracy. This was attributed to its ability to merge the linear trend modeling capability of ARIMA with the nonlinear pattern detection strength of LSTM. The standalone LSTM model performed strongly, while the ARIMA model recorded a lower R^2.

Finally, Yanuar et al. (2024) conducted a comparative study of ARIMA and LSTM models to predict sea level rise in Jakarta. The study found that the ARIMA (1,1,4) model was more effective, producing lower error values MAE, MAPE and RMSE than the LSTM model. The authors concluded that the optimal predictive model is highly dependent on the dominant physical processes of the specific region.

b. Financial Markets

The literature on financial markets often emphasizes the challenges of forecasting due to market volatility and chaotic behavior, leading many studies to favor deep learning over traditional methods.

Fischer and Krauss (2018) were among the first to explore the effectiveness of LSTM networks for financial market predictions. They deployed LSTMs to predict out-of-sample directional movements for S&P 500 stocks from 1992 to 2015. The study found that LSTMs outperformed other memory-free classification methods, yielding daily returns of 0.46% and a Sharpe ratio of 5.8 before transaction costs. They noted, however, that the outperformance seemed to have been arbitraged away after 2010.

However, the majority of research leans toward deep learning models. Siami-Namini et al (2018), using historical monthly financial time series data from indices like Nikkei 225 and S&P 500, found that LSTM significantly outperformed ARIMA, with an average reduction in error rates between 84% and 87%.

In a separate empirical study, Yamak et al (2019), who compared ARIMA, LSTM, and GRU for forecasting Bitcoin prices. Their study found that the ARIMA model produced the best results, outperforming both deep learning models for this specific forecasting task.

A different conclusion was reached by Siami-Namini et al (2019) who conducted a comparative analysis of ARIMA, LSTM, and BiLSTM models for financial time series forecasting. They found that while ARIMA was adequate for stationary data, it failed to capture the nonlinear structures inherent in financial markets. Both LSTM and BiLSTM outperformed ARIMA in accuracy, with BiLSTM delivering the best results by leveraging information from both past and future time steps. This trend continued with Hua (2020) who conducted a comparative study of LSTM and ARIMA to predict Bitcoin prices. The author found that while both models could perform well, the LSTM model achieved a better performance, though it required significant training time. The ARIMA model's precision, in contrast, decreased as the forecasting time grew. Putri and Halim (2020) also compared ARIMA and LSTM for predicting EUR/USD exchange rates. The results showed that LSTM produced a lower RMSE than ARIMA, indicating better prediction results due to its ability to learn from and utilize a large amount of data. Similarly, Ma (2020), in a comparative study of ARIMA, ANN, and LSTM for stock price prediction, found that while ARIMA and ANN have their own advantages and disadvantages, the LSTM model may have the best predictive ability, but its performance is highly dependent on data processing.

Moving to the development of hybrid models, Li et al (2022) introduced the ARIMA -LSTM framework, which incorporates an ARIMA model as a technical feature within an LSTM network. The model was tested on the CSI 300, Nikkei 225, and S&P 500 indices and found to outperform other models, although the performance gain from adding the single ARIMA vector was not significant.

Xiao et al. (2022) further reinforced the superiority of deep learning models in their comparison of ARIMA and LSTM for stock price forecasting using daily data from 50 listed companies. The study found that while ARIMA is computationally efficient, it struggles with the nonlinear and volatile nature of stock prices, and LSTM outperformed ARIMA in accuracy, especially in modeling short-term fluctuations.

In 2023, several studies continued this line of inquiry. García, et al(2023) compared ARIMA, LSTM, and a hybrid model for foreign exchange prediction. They found that while ARIMA is effective for short-term forecasting, its effectiveness decreases for longer periods, and concluded that hybrid models, which combine linear (ARIMA) and non-linear (LSTM) methods, can increase the accuracy of predictions. In a similar finding, Wu et al (2023) compared ARIMA and LSTM for time series prediction and found that the LSTM neural network model provided better predictions, attributing this to its ability to learn from past data.

Additionally, Gonçalves et al (2023) compared ARIMA and LSTM for forecasting Brazil IBX50 closing prices. They found that ARIMA performed better for data points closer to the training data, while LSTM was a more reliable source of prediction for longer forecast windows. Latif et al. (2023), who assessed the prediction of Bitcoin prices using ARIMA and LSTM, found that when the forecast model was re-estimated at each step, the LSTM model consistently surpassed ARIMA. Unlike ARIMA, which only tracked the trend, LSTM was able to predict both the direction and value of prices.

Yavasani and Wang (2023) also compared ARIMA, LSTM, and GRU for predicting stock prices across three economic sectors. The study found that recurrent neural networks (LSTM and GRU) were more effective than the statistical ARIMA model, with GRU consistently performing the best.

Recent research in 2024 has continued to highlight the strengths of deep learning and hybrid approaches. Abu Talib et al (2024) conducted a comparative analysis of ARIMA, LSTM, and GRU for financial time series forecasting of companies like Apple and Tesla. Their findings indicate that while ARIMA is effective for linear trends, deep learning models, particularly LSTM, demonstrated superior predictive accuracy in handling the nonlinear and volatile dynamics of financial markets. In a similar study, Holm and Åkesson (2024) conducted a comparative study of ARIMA and LSTM for forecasting the OMX Stockholm 30 (OMXS30) index. Their findings showed that for a 7-day forecast, both models had similar accuracy. However, for a 30-day forecast, the LSTM model outperformed ARIMA, while for a 60-day forecast, ARIMA yielded better results. García et al (2024) conducted a comparative analysis between LSTM and BiLSTM for foreign exchange forecasting. They noted that while ARIMA has been widely used, neural networks have now surpassed it, and that recurrent neural networks like LSTM perform better than classical econometric models for short-term predictions. In the Moroccan context, Lahboub and Benali (2024) tested ARIMA, LSTM, and transformers for forecasting stock prices. The results showed that the LSTM model achieved high accuracy, with R-squared values over 0.99 for two of the companies and over 0.95 for the third.

The hybrid model approach was further validated by Mochurad and Dereviannyi (2024), who proposed an ensemble forecasting procedure that integrated LSTM and ARIMA models. This hybrid approach was shown to be superior to using either method individually, achieving a significant 15% improvement in root mean square error (RMSE) compared to the standalone LSTM model.

Similarly, Anilkumar et al (2024) conducted a comparative analysis of ARIMA and LSTM for stock price prediction. They concluded that while ARIMA models are computationally efficient and interpretable, LSTM networks are better at capturing complex temporal relationships in time series data. In another study, Kirelli (2024) compared LSTM and ARIMA for forecasting Oracle stock prices. The study found that while the LSTM model had lower error values for MSE, MAE, and RMSE, the ARIMA model had a lower MAPE and was found to produce proportionally more accurate forecasts. The author concluded that the final choice of model should depend on the prioritized metrics.

Zhang (2024), in a comparative study of LSTM and ARIMA for stock price forecasting using five major global stock indices, found that the LSTM model consistently outperformed the ARIMA model, achieving significantly lower MSE and RMSE values. This was attributed to LSTM's ability to handle non-linear relationships and “abnormal fluctuations”.

In another study, Asha et al (2024) analyzed the performance of ARIMA, LSTM, and FBProphet for predicting stock market values. The study concluded that each method has its own strengths: ARIMA for steady linear data, LSTM for nonlinear relationships, and FBProphet for trend and seasonality.

Looking ahead, Alharbi (2025) examined forecasting the stock price of Saudi Basic Industries Corporation (SABIC) using a hybrid ARIMA–LSTM model. The study found that this hybrid approach consistently outperformed both individual ARIMA and LSTM models across all evaluation metrics, demonstrating its superior accuracy and robustness.

Pilla and Mekonen (2025) conducted a comparative study of ARIMA and LSTM to forecast the S&P 500 index. The LSTM model significantly outperformed ARIMA, achieving an accuracy of 96.41% compared to ARIMA's 89.8%. The authors concluded that LSTM networks are a powerful tool for handling the volatile, nonlinear, and complex nature of financial data.

Finally, Zheng et al (2025) evaluated the forecasting effectiveness of the ARIMA and LSTM models for financial time series prediction. They found both models to be effective, noting that the LSTM model's strength lies in its ability to analyze nonlinear relationships, while the ARIMA model is suitable for simpler, linear markets. They also highlighted that an LSTM network ensemble improved prediction and reduced overfitting, with an average relative error compared to ARIMA's average relative error.

3. Data and Methodology

a. Data

The forecasting of the MASI index using both the ARIMA and LSTM models was conducted on a time series dataset of daily closing prices. The data covers a comprehensive period from January 4, 2010, to August 8, 2025. The period 2010–2025 was selected as it ensures reliable and consistent data coverage for the MASI index, provides a sufficiently long horizon to capture both short- and long-term market dynamics, and includes recent events such as the COVID-19 crisis and subsequent recovery. This extensive timeframe provides also a sufficiently large sample size for both model training and robust out-of-sample evaluation. The MASI (Moroccan All-Share Index) is the main stock market index for the Casablanca Stock Exchange (Bourse de Casablanca) in Morocco.1 It is a capitalization-weighted index that tracks the performance of all listed companies on the exchange, providing a comprehensive measure of the overall health and performance of the Moroccan stock market.2 As such, forecasting the MASI index is a relevant task for market participants, investors, and economic analysts interested in the Moroccan financial landscape. For model validation, the dataset was split chronologically. The training sample consisted of the first 80% of the data, while the remaining 20% was designated as the test sample. This approach ensured that the models were trained exclusively on past information and then evaluated on unseen, future data, providing a realistic assessment of their predictive performance.

The daily closing prices of the Moroccan All-Share Index (MASI) covering the period 2010–2025 were obtained from Investing.com, a widely used financial data platform that provides free access to historical market data for global equity indices, commodities, currencies, and other financial instruments.

b. Methodology

This study employs two prominent approaches for forecasting the Moroccan All-Share Index (MASI): the Autoregressive Integrated Moving Average (ARIMA) model and the Long Short-Term Memory (LSTM) neural network. Both models are formulated mathematically and implemented to capture different characteristics of financial time series.

We used the daily closing prices of the MASI index for both ARIMA and LSTM models, as the objective is to forecast the future level of the index rather than short-term returns.

i. ARIMA Model

The Autoregressive Integrated Moving Average (ARIMA) model is a classical statistical method used for time series forecasting (Box and Jenkins, 1970). It combines three core components to capture different aspects of a dataset's behavior: autoregression, differencing, and moving average. The model is typically denoted as $ARIMA(p, d, q)$, where each parameter represents the order of its respective component.

Components of ARIMA

- Autoregressive (AR) Component (p): This part of the model assumes that the current value of the time series, (y_t) , is a linear combination of its own past values. The order refers to the number of lagged observations included in the model.
- Integrated (I) Component (d): This component involves a differencing operation applied to the time series to make it stationary. A stationary series is one whose statistical properties (mean, variance, and autocorrelation) are constant over time. The order d indicates the number of times the data has been differenced.
- Moving Average (MA) Component (q): This part of the model assumes that the current value is a linear combination of past forecast errors. The order q represents the number of lagged forecast errors in the model.

Mathematical Formulation

The full $ARIMA(p, d, q)$ model is formulated by combining these components into a single equation. The general form is:

$$\phi(B)(1-B)^d y_t = \theta(B)\varepsilon_t$$

Where:

y_t is the value of the time series at time t.

B is the backshift operator, such that $By_t = y_{t-1}$

d is the degree of differencing. The term $(1-B)^d$ represents the differencing operation.

$\phi(B)$ is the autoregressive polynomial of order p :

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$$

$\theta(B)$ is the moving average polynomial of order q:

$$\theta(B) = 1 + \theta_1 B \mp \theta_2 B^2 \mp \cdots + \theta_q B^q$$

ε_t is the white noise error term at time t.

The coefficients ϕ_i and θ_j are the model parameters that are estimated from the data.

Forecasting

To effectively evaluate a forecasting model, a time series dataset must be partitioned into a training sample and a test sample. This method, often referred to as a train-test split, is fundamental for assessing how well a model generalizes to unseen data.

Data Splitting

The first step is to divide the historical data into two distinct, non-overlapping sets.

- Training Sample: This portion of the data (e.g., 80% of the total dataset) is used to fit the model. The model learns the underlying patterns, trends, and seasonal components from this historical data. For ARIMA model, this is where the parameters are estimated and the model's weights are optimized.

- Test Sample: This is the remaining portion of the data (e.g., the last 20%). The model has no prior exposure to this data. It is used to evaluate the model's performance by comparing its forecasts against the actual values in this test set. This provides an unbiased measure of the model's predictive accuracy.

Model Fitting

The ARIMA model is fitted to the training sample. The optimal parameters (p, d, q) are determined using methods like the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots or by minimizing information criteria like AIC or BIC. The model learns the coefficients (ϕ_i and θ_j) that best describe the training data's behavior.

Forecasting and Evaluation

Once the models are fitted, they are used to generate predictions.

- Forecasting: The trained model is used to forecast future values for the period covered by the test sample. For example, if the test set contains data for the last 100 days, the model will generate a 100-step-ahead forecast.
- Comparison: The forecasted values are then compared to the actual, observed values in the test sample.
- Performance Metrics: The difference between the forecasted and actual values is quantified using various performance metrics, such as, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), Coefficient of Determination (R^2)

These metrics provide a quantitative assessment of each model's predictive accuracy. A model is considered to have better forecasting performance if it consistently yields lower error values (MAE, RMSE, MAPE) and a higher R^2 on the unseen test data.

ii. LSTM model

The Long Short-Term Memory (LSTM) model, introduced by Hochreiter and Schmidhuber (1997), is a specialized type of neural network designed for processing and forecasting time series data.

Unlike traditional neural networks, LSTMs have a unique architecture that allows them to remember important information from long sequences of data while forgetting irrelevant details. This is achieved through a core component called the memory cell, which acts like a conveyor belt, carrying information forward through time.

The flow of information within an LSTM is regulated by three main "gates":

- Forget Gate: This gate decides what information from the past is no longer needed and can be discarded.

- Input Gate: This gate determines which new information from the current time step is important enough to be stored in the memory cell.
- Output Gate: This gate controls what information from the memory cell is used to produce the output for the current time step.

This gated mechanism allows LSTMs to effectively handle time series data with complex, long-term dependencies, making them particularly well-suited for tasks such as financial market prediction, where patterns from the distant past can be crucial for understanding current behavior.

The following equations represent the flow of information through a single LSTM unit at time step t.

- **Forget Gate (f_t)**

The forget gate decides which information from the previous cell state (C_{t-1}) should be discarded. It takes the previous hidden state (h_{t-1}) and the current input (x_t), and outputs a value between 0 and 1, where 0 means "completely forget" and 1 means "completely keep".

$$f_t = \sigma\left(W_f(h_{t-1}, x_t) + b_f\right)$$

σ is the sigmoid function, which squashes the values to a range between 0 and 1.

W_f and b_f are the weight matrix and bias vector for the forget gate.

(h_{t-1}, x_t) is the concatenation of the previous hidden state and the current input.

- **Input Gate (i_t) and Candidate Cell State ($\tilde{C}_t$)**

The input gate decides which new information from the current input (xt) should be stored in the cell state. It has two parts:

- The input gate layer (i_t) uses a sigmoid function to determine which values to update.
- The tanh layer ($\tilde{C}_t$) creates a vector of new candidate values that could be added to the state.

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

$$\tilde{C}_t = tanh(W_C(h_{t-1}, x_t) + b_C)$$

- W_i, W_C and b_i, b_C are the weight matrices and bias vectors for the input gate and candidate cell state, respectively.

  tanh is the hyperbolic tangent function, which squashes the values to a range between -1 and 1.

- **New Cell State (C_t)**

The new cell state is created by combining the previous cell state, the forget gate's decision, and the input gate's decision. This is where the old information is forgotten and new information is stored.

$$C_t = f_t \circledast C_{t-1} + i_t \circledast \tilde{C}_t$$

$\circledast$ denotes element-wise multiplication.

d. Output Gate (o_t) and Hidden State (h_t)

The output gate decides what part of the new cell state will be the final output of the LSTM unit. The output is filtered and used to create the new hidden state ($\boldsymbol{h_t}$), which is then passed on to the next time step.

$$o_t = \sigma(W_o(h_{t-1}, x_t) + b_o)$$

$$h_t = o_t \circledast tanh(C_t)$$

W_o and b_o are the weight matrix and bias vector for the output gate.

h_t is the new hidden state, which also serves as the final output for the current time step.

Forecasting with the LSTM Model

To evaluate the forecasting performance of an LSTM (Long Short-Term Memory) model, a time series dataset must be partitioned into a training sample and a test sample. This method is fundamental for assessing how well the model generalizes to unseen data.

Data Splitting

First, the historical data is divided into two distinct, non-overlapping sets.

- Training Sample: This portion of the data is used to fit the model. The LSTM network learns the underlying patterns, trends, and seasonal components from this historical data. During this stage, the network's weights and biases are optimized through an iterative training process.
- Test Sample: This is the remaining, chronologically later portion of the data. The model has no prior exposure to this data. It is reserved for evaluating the model's performance by comparing its forecasts against the actual values in this set. This provides an unbiased measure of the model's predictive accuracy.

For time series data, it is crucial to use a chronological split, where the training data precedes the test data. Randomly splitting the data would introduce future information into the training process, leading to look-ahead bias, which results in unrealistically optimistic performance metrics.

Model Fitting

The LSTM network is trained on the training sample. The preprocessed data is formatted into sequences of input-output pairs. The network then learns the weights and biases through an optimization process (e.g., Adam optimizer) by minimizing a loss function (e.g., Mean Squared Error) over a defined number of epochs.

Forecasting and Evaluation

After fitting the model, it is used to generate predictions, which are then evaluated against the test sample.

a. Forecasting: The trained model generates forecasts for the future period covered by the test sample. For example, if the test set contains data for the last 100 days, the model will generate a 100-step-ahead forecast.

b. Comparison: The forecasted values are compared to the actual, observed values in the test sample.

c. Performance Metrics: The difference between the forecasted and actual values is quantified using various performance metrics to provide a quantitative assessment of the model's predictive accuracy. A model is considered to have better forecasting performance if it consistently yields lower error values (MAE, RMSE, MAPE) and a higher R^2 on the unseen test data.

4. Results and discussion

a. Results of ARIMA model applied to MASI index series

- **Results of the ADF test applied to MASI index series**

The objective of this section is to examine the stationarity of the MASI index using the ADF test. The analysis will focus on both the original series and the differenced series to determine whether a transformation is necessary to achieve stationarity. The results are presented in table 1.

Table 1: ADF test applied to MASI index series

Test	ADF Statistic	1% Critical Value	5% Critical Value	10% Critical Value
Original Series	1.4316	-3.43	-2.86	-2.57
Differenced Series	-15.5039	-3.43	-2.86	-2.57

The ADF test for the original MASI series produced an ADF Statistic of 1.4316. This value is greater than all the critical values at 1%, 5% and 10% (-3.43, -2.86, and -2.57). This result leads us to fail to reject the null hypothesis, which states that the time series has a unit root and is non-stationary. Therefore, we conclude that the original MASI series is non-stationary.

The ADF test for the differenced MASI series yielded an ADF Statistic of -15.5039. This value is significantly smaller than all the critical values at 1%, 5% and 10% (-3.43, -2.86, and -2.57). This allows us to reject the null hypothesis and conclude that the differenced series is stationary. This means that taking the first difference of the data successfully removed the non-stationarity, making the series suitable for analysis with ARIMA model that requires this assumption. Based on this result, we'll set the differencing order, d, to 1.

- **Selecting optimal parameters of ARIMA**

When a forecasting time series using ARIMA model, it's essential to select the best set of parameters. A common challenge is balancing a model's goodness of fit (how well it explains the historical data) with its complexity (the number of parameters used). A model that fits the data perfectly might be too complex and perform poorly on new, unseen data, a phenomenon known as overfitting. To address this, we use statistical criteria like the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). The goal is to find the model with the lowest AIC and BIC values, as this indicates the most efficient and robust model.

The following table lists the AIC and BIC scores for each ARIMA(p,1,q) model where $0 \leq p \leq 3$ and $0 \leq q \leq 3$.. The results of the grid search are as follows.

Table 2: AIC and BIC scores for ARIMA(p,1,q) with $0 \leq p \leq 3$ and $0 \leq q \leq 3$

Model	AIC Score	BIC Score
ARIMA(0,1,0)	45303.99	45310.26
ARIMA(0,1,1)	45255.67	45268.20
ARIMA(0,1,2)	45210.67	45229.46
ARIMA(0,1,3)	45212.50	45237.57
ARIMA(1,1,0)	45245.53	45258.07
ARIMA(1,1,1)	45229.41	45248.21
ARIMA(1,1,2)	45212.50	45237.56
ARIMA(1,1,3)	45214.37	45245.70
ARIMA(2,1,0)	45214.92	45233.71
ARIMA(2,1,1)	45213.56	45238.62
ARIMA(2,1,2)	45214.50	45245.83
ARIMA(2,1,3)	45215.71	45253.30
ARIMA(3,1,0)	45212.40	45237.47
ARIMA(3,1,1)	45213.05	45244.38
ARIMA(3,1,2)	45214.99	45252.58
ARIMA(3,1,3)	45216.96	45260.82

The results of the grid search for optimal ARIMA parameters show that:

- ✓ ARIMA(0,1,2) has the lowest AIC score of 45210.67 and the lowest BIC score of 45229.46. This makes it the optimal model according to both criteria among all the tested configurations.
- ✓ The AIC and BIC scores for ARIMA(0,1,2) are significantly lower than those of simpler models like ARIMA(0,1, 0) (AIC: 45303.99, BIC: 45310.26) and ARIMA(1, 1, 0) (AIC: 45245.53, BIC: 45258.07). This suggests that including a moving average component of order 2 is crucial for accurately modeling the time series.
- ✓ The results also show that ARIMA(3,1,0) and ARIMA(2,1,0) are the second-best options based on AIC and BIC, respectively. However, their scores are still higher than ARIMA(0, 1, 2), confirming that the latter is the most suitable choice.

In conclusion, based on the lowest AIC and BIC values, the ARIMA(0,1,2) model is the most optimal configuration for forecasting the MASI index.

- **Forecasting the MASI Index with ARIMA(0,1,2) model**

This section presents the results of a time series forecast for the MASI index. The forecasting was performed using an ARIMA(0,1,2) model, which was selected as the optimal model in a previous analysis based on its low AIC and BIC scores. The objective of this exercise is to evaluate the model's performance by comparing its predictions to actual market data. The forecast is generated for a specific test period, allowing us to assess how well the ARIMA(0,1,2) model captures the dynamic movements of the MASI index.

The following figure illustrates the model's performance, displaying the historical data, the actual prices in the test set, and the prices forecasted by the model.

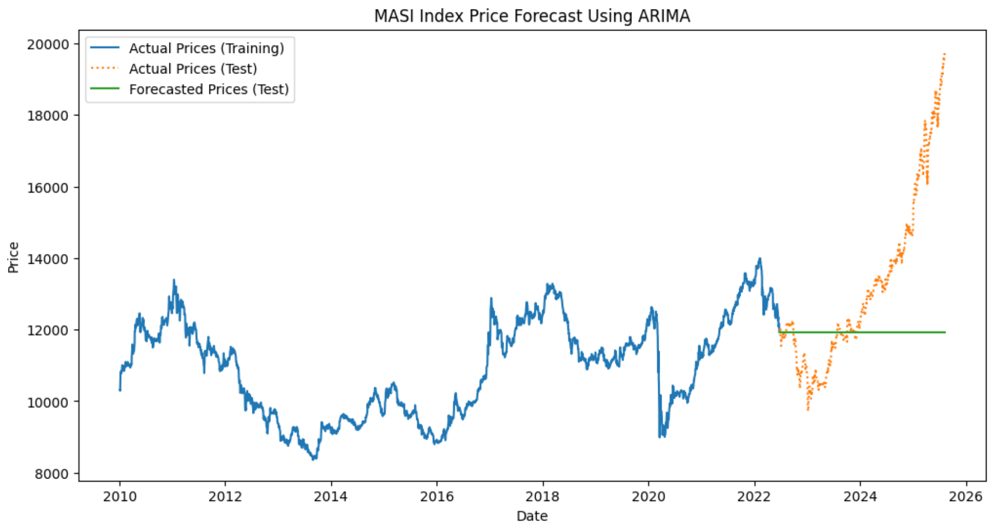

Figure 1: Forecasting MASI index using ARIMA(0,1,2) model

This figure shows the actual MASI prices over time, split into training and test sets, and the forecasted prices for the test set. Initially, the forecast seems to roughly follow the trend, but it quickly flattens out and remains relatively constant. This suggests that the ARIMA(0,1,2) model, while capturing the differencing aspect ($d = 1$), may not be fully capturing the underlying patterns and volatility in the MASI index to provide accurate long-term forecasts. The significant divergence between the actual test prices and the forecasted prices in the later part of the test set indicates that the model's predictive power is limited over this horizon with the current parameters.

- **Forecasting Errors (MAE and RMSE)**

To assess the predictive accuracy of the ARIMA(0,1,2) model, we calculated the two error metrics, the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE). These metrics provide a quantitative measure of how closely our model's forecasts align with the actual values, which is essential for determining its reliability. The results are presented in table 3.

Table 3: Accuracy of the ARIMA(0,1,2) model measured by MAE and RMSE

Mean Absolute Error (MAE)	Root Mean Squared Error (RMSE)
1919.67	2754.37

The high values for both the MAE (1919.67) and RMSE (2754.37) indicate that the ARIMA(0,1,2) model's predictions are quite high relative to the scale of the MASI index prices.

- **Residuals of the ARIMA(0,1,2) model**

After fitting a time series model ARIMA(0,1,2), it's crucial to analyze the residuals (the errors between the actual values and the model's predictions). This step is essential to ensure the model is valid and has captured all the relevant information in the data. The following figure presents a plot of the residuals from the fitted ARIMA(0,1,2) model.

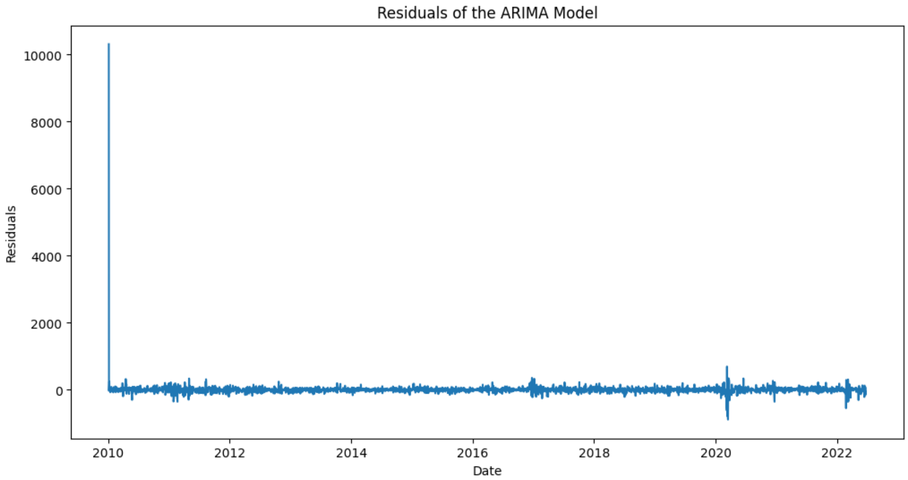

Figure 2: Plot of the residuals of the ARIMA(0,1,2) model

This plot shows the difference between the actual training prices and the prices predicted by the fitted ARIMA(0,1,2) model on the training data. Ideally, the residuals should look like white noise, random, centered around zero, and with no discernible patterns. In this plot, the residuals appear mostly centered around zero after an initial large value, but there might be some clustering or periods of higher volatility. Examining the ACF and PACF of the residuals will provide more formal insights into whether they are truly random.

- **Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) of the residuals for ARIMA(0,1,2) model**

After fitting an ARIMA model, it's crucial to check if the residuals are random and contain no remaining patterns. If the residuals are not random, it means our model hasn't captured all the information in the data, and there's room for improvement. We use two important plots to analyze the randomness of residuals, the Autocorrelation Function (ACF) plot and the Partial Autocorrelation Function (PACF) plot. ACF Plot shows the correlation between residuals at different lag periods. PACF Plot shows the partial correlation between residuals at different lag periods, removing the influence of intermediate lags. Figures 1 and 2 show respectively the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) of the residuals for ARIMA(0,1,2) model.

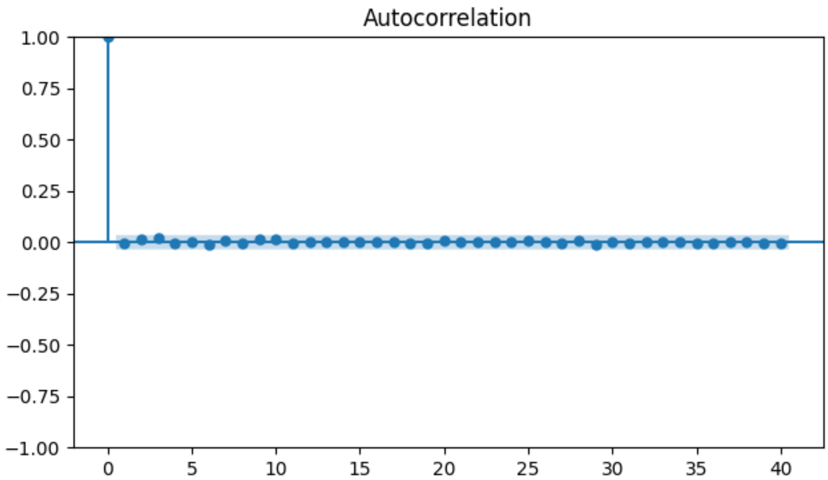

Figure 3: Autocorrelation Function (ACF) of Residuals for ARIMA(0,1,2) model

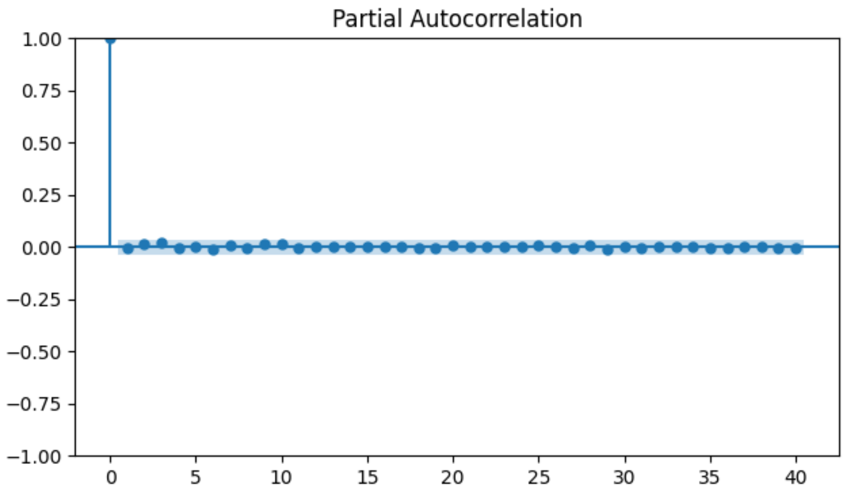

Figure 4 : Partial Autocorrelation Function (PACF) of residuals for ARIMA(0,1,2) model

Ideally, for a well-fitted ARIMA(0,1,2) model, the residuals should be uncorrelated, meaning most of the spikes in the ACF and PACF plots should fall within the blue shaded area (which represents the confidence interval for zero correlation). In figure 4, most of the lags are within the confidence intervals, suggesting that the ARIMA(0,1,2) model has captured most of the linear dependencies in the differenced series. There might be a few small spikes just outside the confidence interval, which could warrant further investigation or consideration of different model orders.

- **QQ-plot of residuals**

After fitting a time series model, it's important to check the assumptions about the residuals (the model's errors). One key assumption in many statistical analyses is that the residuals are normally distributed. The QQ-plot (Quantile-Quantile plot) is a graphical tool used to assess this assumption. The QQ-plot works by comparing the quantiles of our residuals' distribution against the quantiles of a theoretical normal distribution. If the residuals are normally distributed, the points on the plot will fall roughly along a straight diagonal line. Any significant departure from this line, especially at the ends (the “tails”), indicates that the residuals may not be normally distributed. The following plot shows whether the residuals from our

ARIMA(0,1,2) model align with a normal distribution. Figure 5 displays the QQ-plot of Residuals for ARIMA(0,1,2) model.

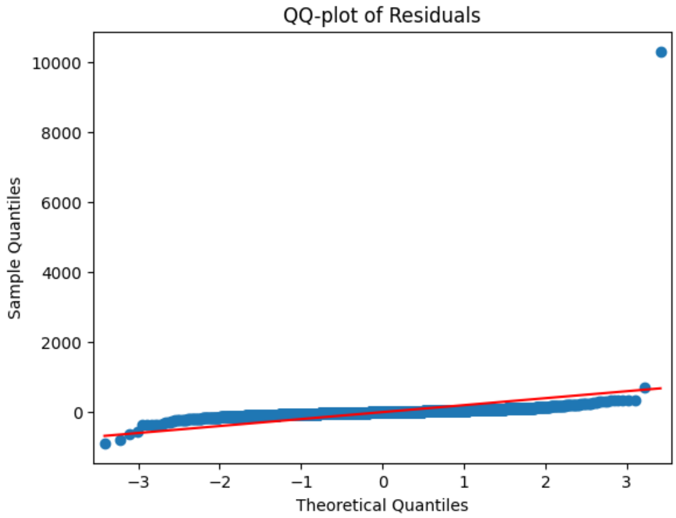

Figure 5: QQ-plot of Residuals for ARIMA(0,1,2) model

The QQ-plot shows some deviation from the red line, particularly at the tails (the points in the upper right and lower left corners are further away from the line). The point in the upper right corner is notably far off the line. This suggests that the residuals may not be perfectly normally distributed, which is an assumption of some statistical tests applied to ARIMA models. However, deviations from normality in residuals are not uncommon in financial time series.

- **Ljung-Box test on residuals**

After fitting a time series model, it's crucial to ensure that the residuals are random. The Ljung-Box test is a formal statistical method used to check for any remaining autocorrelation in these residuals. The test's null hypothesis (H0) states that the residuals are independently distributed, meaning there is no significant autocorrelation. The alternative hypothesis (Ha) suggests that there is autocorrelation. We perform this test to formally confirm what we might observe from the ACF and PACF plots. If the test's p-value is high (typically greater than 0.05), we fail to reject the null hypothesis, which gives us confidence that our model has successfully captured all the linear patterns in the data.

The Ljung-Box test statistic, also known as the Q-statistic is equal to 4.3787. The p-value is equal to 0.999903. This is the probability of observing a test statistic as extreme as 4.3787, assuming the residuals are truly random. In hypothesis testing, we compare the p-value to a significance level (e.g., 0.05). Since our p-value (0.999903) is much greater than 0.05, we fail to reject the null hypothesis. The conclusion is that there is no significant autocorrelation in the residuals up to lag 20. This supports the idea that the ARIMA(0,1,2) model has adequately captured the linear dependencies in the data, leaving uncorrelated residuals.

The ARIMA(0,1,2) model is statistically adequate for capturing the MASI's linear dependencies, with uncorrelated residuals and optimal AIC/BIC scores. However, its forecasting power, particularly for capturing volatility and longer-term patterns, is limited, as shown by the high error metrics and flattening forecasts.

b. Results of LSTM model

To prepare the MASI index data for the LSTM model, the time series was first preprocessed. The data was split into training and test sets, with 80% of the data used for training and the remaining 20% for testing. To improve model stability and performance, the price data was scaled to a range between 0 and 1 (using the MinMaxScaler).

The LSTM model architecture consisted of a sequential stack of layers. The input layer was configured with a time step of 60, meaning the model was trained to predict the next day's price based on the previous 60 days of data. The network itself was composed of two LSTM layers, each with 50 units, followed by a single dense output layer with one unit to produce the final price prediction.

For training, the model was compiled using the "Adam" optimizer and the "mean squared error" loss function. The training process was run for 10 epochs with a batch size of 32. These parameters were chosen to ensure the model had sufficient opportunities to learn the underlying patterns in the time series data.

- **Price forecasting by LSTM model**

To evaluate the forecasting performance of the LSTM model on the MASI index, a visual comparison between actual and forecasted prices was conducted. The following plot illustrates how well the model captures the index's dynamics over the test period by juxtaposing historical training data, test data, and the model's forecasts. This graphical representation provides an intuitive assessment of the model's ability to track the underlying trends and movements of the MASI index. Figure 6 shows the actual and forecasted prices of MASI index using LSTM model.

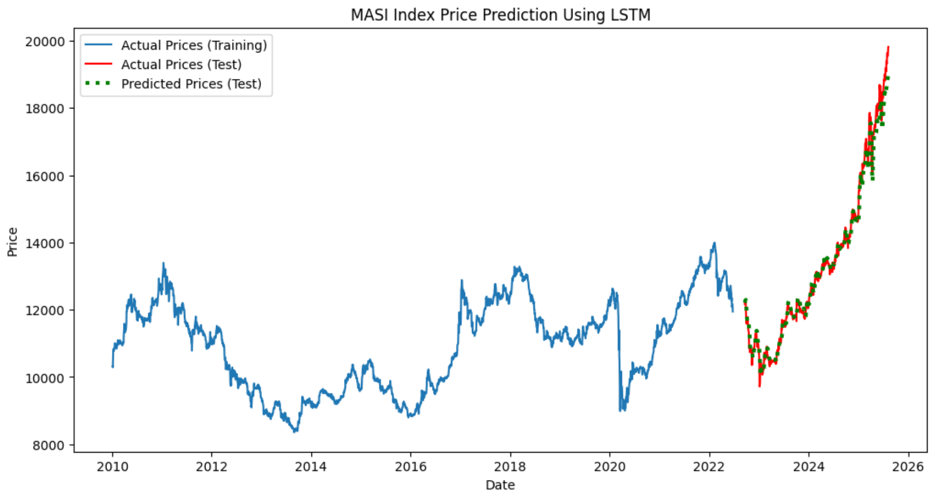

Figure 6: Forecasting MASI index using LSTM model

This plot visually compares the actual MASI prices (training and test) with the predicted prices on the test set. The LSTM model demonstrates a reasonable ability to forecast the MASI index, as reflected in the visual alignment between predicted and actual prices during the test period. The predicted values generally follow the direction and overall trend of the actual test set prices, indicating that the model effectively captures the broad movements of the index. However, some deviations are observed, particularly in periods of higher volatility, where the model tends to smooth out sharp price changes. This behavior suggests that while the LSTM can model the underlying temporal patterns and trend dynamics, it may have limitations in fully replicating short-term fluctuations or abrupt market shifts, which are common in financial time series.

- **Error metrics:**

The LSTM model's predictive accuracy was assessed using two widely used error measures, the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). The results are presented in table 4.

Table 4: Accuracy of the LSTM model measured by MAE and RMSE

Mean Absolute Error (MAE)	Root Mean Squared Error (RMSE)
178.52	272.69

The value MAE =178.52 reflects that on average, the model's predictions deviate from the actual MASI prices by about 178.52 points during the test period. This metric treats all errors equally, regardless of their direction or magnitude, making it a straightforward indicator of overall accuracy. The value RMSE = 272.69 reflects the typical size of prediction errors but penalizes larger errors more heavily due to the squaring process before averaging. An RMSE of 272.69 means that, in general, the magnitude of the model's prediction errors is around 272.69 points, with bigger mistakes contributing disproportionately to this measure.

The relatively low values of both metrics, especially when compared to the overall scale of the MASI index, suggest that the LSTM model provides a strong fit to the test data. However, the higher RMSE compared to MAE also indicates that occasional larger prediction errors occurred, likely during periods of heightened volatility or abrupt price movements.

- **Training Loss Curve Plot**

To monitor the learning progress of the LSTM model, the evolution of the training loss; measured as the mean squared error, was tracked across 10 epochs. This curve provides insight into how quickly and effectively the model adapted to the training data. By examining the loss trend, we can assess whether the model converged to an optimal solution, overfitted, or underfitted during the training process. Figure 7 displays the training loss curve of the LSTM model.

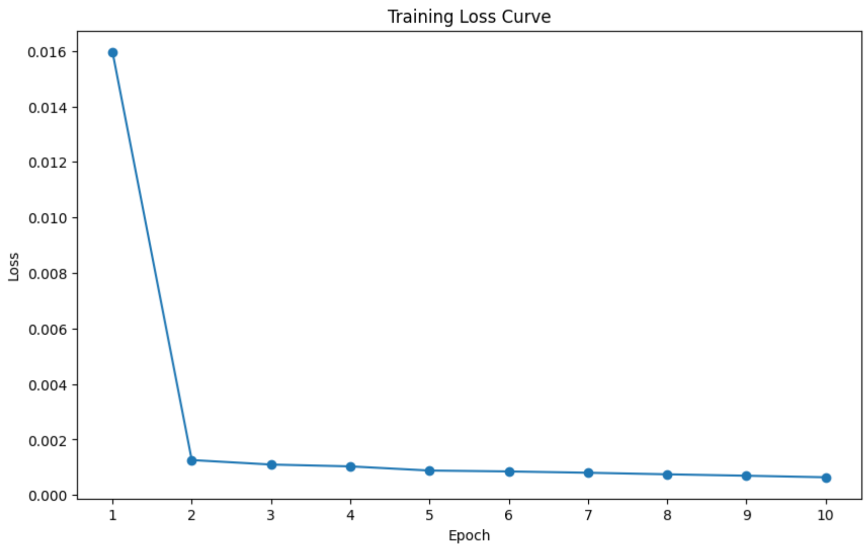

Figure 7: Training Loss Curve of the LSTM model

The training loss curve displays a steep decline during the initial epochs, indicating that the LSTM model quickly captured the most prominent patterns in the MASI index data. This rapid early improvement is typical in deep learning, as the model first learns broad structural relationships before refining finer details. In subsequent epochs, the loss decreases more gradually, suggesting that the model shifted from learning general patterns to optimizing smaller, more complex relationships in the data. The absence of any upward trend in the loss indicates that overfitting was not evident during the 10-epoch training period, and the model achieved stable convergence toward a satisfactory solution.

Comparison the performance of ARIMA and LSTM models

For forecasting the MASI index, the ARIMA(0,1,2) and LSTM models show distinct performance characteristics. ARIMA(0,1,2) captures linear dependencies well, as evidenced by uncorrelated residuals and optimal AIC/BIC scores, but its forecasts tend to flatten and produce high errors (MAE: 1919.67, RMSE: 2754.37), indicating limited ability to track volatility and complex patterns. In contrast, the LSTM model, a deep learning approach designed to capture nonlinear temporal dependencies, demonstrates superior predictive accuracy (MAE: 178.52, RMSE: 272.69), effectively following the overall trend of the index while smoothing extreme short-term fluctuations. The LSTM's training loss curve shows rapid convergence without overfitting, reflecting efficient learning of underlying patterns. Overall, while ARIMA(0,1,2) provides a reliable linear baseline, the LSTM model outperforms it in capturing the dynamic and nonlinear behavior of the MASI index, making it better suited for forecasting in volatile financial markets.

c. Comparison with previous studies

Our study's findings, which show the superiority of the LSTM model over the ARIMA model for forecasting the MASI index, aligns with a significant body of recent literature on financial time series forecasting.

As our results demonstrated, the LSTM model was more effective at capturing the complex, non-linear dynamics of the MASI index. This finding is consistent with the conclusions of several studies. For example, Siami-Namini et al (2019) found that LSTM and BiLSTM models significantly outperformed ARIMA for various financial indices, with an average error rate reduction of up to 87%. Similarly, Xiao et al. (2022) concluded that LSTM's ability to handle the non-linear and volatile nature of stock prices makes it superior to the computationally efficient but linear-dependent ARIMA model. The findings of Zhang (2024), who also found that LSTM consistently outperformed ARIMA for major global stock indices, further reinforce our conclusion that deep learning models are better suited for this type of data.

Our observation that the ARIMA model struggled to capture the volatility and abrupt movements in the MASI index is also a common theme in the literature. While ARIMA models are effective for linear trends and stationary data, as noted by Siami-Namini et al (2018), they often fall short in financial markets. Pilla and Mekonen (2025)'s study on the S&P 500 index supports this, with LSTM significantly outperforming ARIMA due to the latter's inability to handle the "volatile, non-linear, and complex nature of financial data".

However, it's important to acknowledge studies that have yielded different results. Our findings contrast with some research, particularly in non-financial contexts or for specific financial assets. For instance, Yamak et al (2019) found that ARIMA produced the best results for forecasting Bitcoin prices, a finding that runs counter to our own. Similarly, Khulood Albeladi et al. (2023) concluded that ARIMA was more suitable for a specific real estate dataset. These discrepancies highlight that the optimal model is often highly dependent on the specific characteristics of the dataset, as suggested by Yanuar et al. (2024) in their study on sea level rise.

The literature also points to a growing trend of using hybrid models to combine the strengths of both linear and non-linear approaches. Our conclusion that future research could explore hybrid models is supported by studies like those of Hamiane et al. (2024) and Mochurad and Dereviannyi (2024), both of which found that hybrid ARIMA-LSTM models delivered superior predictive accuracy by leveraging the linear trend modeling of ARIMA and the non-linear pattern detection of LSTM. This suggests a promising avenue for future work to potentially improve upon the results of our standalone LSTM model.

Finally, our study contributes to the limited literature on the Moroccan financial market. While studies like Lahboub and Benali (2024) have shown the high accuracy of LSTM for forecasting specific Moroccan stock prices, our research provides a comparative analysis on a broader index level (MASI), further solidifying the case for adopting deep learning methods in this specific market context.

5. Conclusion

This study aimed to forecast the MASI (Moroccan All Shares Index) using both classical and modern approaches, namely the ARIMA(0,1,2) model and a Long Short-Term Memory (LSTM) deep learning neural network. Preliminary analysis revealed that the MASI index series is non-stationary and becomes stationary after first differencing, indicating that it is integrated of order 1. The ARIMA model parameters were selected through a systematic grid search, optimizing both the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) to achieve the best balance between model fit and complexity. While ARIMA(0,1,2) model successfully captured the linear dependencies in the MASI series, as confirmed by residual diagnostics, its forecasting performance was limited, producing high error metrics and forecasts that smoothed out the index's volatility and abrupt movements.

In contrast, the LSTM model demonstrated strong predictive capability, accurately capturing the overall trends and temporal dependencies in the MASI index. The relatively low MAE and RMSE, combined with a stable training loss curve, indicate that the model learned efficiently without overfitting. Although the LSTM tended to smooth extreme short-term fluctuations, it outperformed ARIMA in forecasting accuracy and in handling the nonlinear and complex dynamics inherent in financial time series.

Overall, the results suggest that deep learning models like LSTM are more suitable for modeling and forecasting volatile stock market indices than traditional linear models such as ARIMA.

The findings of this study have several important practical implications for market participants and researchers focused on the Moroccan financial market. For investors and traders, the superior performance of the LSTM model suggests that strategies relying on forecasting the MASI index should leverage deep learning methods for more reliable predictions. A more accurate predictive model can lead to better-informed decisions, potentially increasing profitability and reducing risk. Financial analysts at institutions in Morocco should also consider incorporating deep learning models into their analytical toolkit for a more nuanced understanding of market movements, which is essential for risk management and portfolio optimization. Furthermore, this study serves as a foundational step for future research into forecasting the MASI index, encouraging further exploration of more advanced deep learning architectures, such as hybrid models that combine the strengths of both traditional and deep learning approaches.

Despite the promising results, this study has several limitations. First, it focuses exclusively on the MASI index, which may limit the generalizability of the findings to other Moroccan or emerging market indices. Second, only historical daily data were used, so short-term intraday patterns, high-frequency fluctuations, and long-term structural changes were not captured.

Future research could explore the development of a hybrid ARFIMA–LSTM model, which combines the strengths of both approaches. While ARFIMA effectively captures long-memory linear dependencies commonly present in financial time series, LSTM excels at modeling nonlinear and complex temporal patterns. By applying ARFIMA to extract the long-memory structure and then using LSTM to learn the residual nonlinear dynamics, such a hybrid model could improve forecast accuracy and better capture abrupt market movements.

6. References

- Abu Talib, N., Ahmad, R., & Noor, S. N. (2024). A comparative analysis of machine learning models for time series forecasting in finance. *International Journal of Applied Mathematics and Computing, 1*(1), 20–26. https://doi.org/10.62951/ijamc.v1i2.71
- Albeladi, K., Zafar, B., & Mueen, A. (2023). Time series forecasting using LSTM and ARIMA. *International Journal of Advanced Computer Science and Applications, 14*(1), 241–252. https://doi.org/10.14569/IJACSA.2023.0140133
- Alharbi, M.H. (2025). Prediction of the stock market using LSTM, ARIMA, and hybrid of LSTM-ARIMA models. *J. Kno.Mana.App.Pra. 7, No. 1, 15-22 (2025)*. http://dx.doi.org/10.18576/jkmap/070102
- Andréasson, D., & Blomquist, J.M. (2020). *Forecasting the OMXS30 – a comparison between ARIMA and LSTM*. Master's thesis, Department of Statistics. Uppsala University.
- Anilkumar P.S., Ferdouse L., & Ajmery S. (2024). Comparative Analysis of ARIMA and LSTM Models for Stock Price Prediction. In *2024 IEEE/ACIS 27th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*. IEEE.
- Asha S., Kumar C.S., Narayana G.S., Koppera V., & Manish H. (2024). Time series forecasting of stock market using ARIMA, LSTM and FB prophet. *MATEC Web of Conferences, 392*, 01163.
- Box, G.E.P., & Jenkins, G.M. (1970). *Time Series Analysis: Forecasting and Control*. San Francisco, CA: Holden-Day.
- Elmasdotter, A., & Nyströmer, C. (2018). A comparative study between LSTM and ARIMA for sales forecasting in retail [*Bachelor's thesis, KTH Royal Institute of Technology*]. DiVA portal. http://www.diva-portal.org/smash/get/diva2:1214307/FULLTEXT01.pdf
- Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research, 270*(2), 654-669. https://doi.org/10.1016/j.ejor.2017.11.054
- García, F., Guijarro, F., Oliver, J., & Tamošiūnienė, R. (2023). Foreign Exchange Forecasting Models: ARIMA and LSTM Comparison. *Eng. Proc. 2023, 39*(1), 81. https://doi.org/10.3390/engproc2023039081
- García, F., Guijarro, F., Oliver, J., & Tamošiūnienė, R. (2024). Foreign Exchange Forecasting Models: LSTM and BiLSTM Comparison. *Engineering Proceedings, 68*(1), 19. https://doi.org/10.3390/engproc2024068019
- Goncalves J.V.M. & Alexandre M. & Gilberto T.L., 2023. ARIMA and LSTM: A Comparative Analysis of Financial Time Series Forecasting. *Working Papers, Department of Economics* 2023_13, University of São Paulo (FEA-USP).

- 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.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
- Holm, A., & Åkesson, A. (2024). *A comparison of LSTM and ARIMA forecasting of the stock market*. Degree project, School of Electrical Engineering and Computer Science.
- Hua, Y. (2020). Bitcoin price prediction using ARIMA and LSTM. *E3S Web of Conferences, 218*, 01050. https://doi.org/10.1051/e3sconf/202021801050
- Khan, S.M. (2022). Application of Deep Learning LSTM and ARIMA Models in Time Series Forecasting: A Methods Case Study analyzing Canadian and Swedish Indoor Air Pollution Data. *Austin Journal of Medical Oncology*, 9(1), 1073.
- Kirelli, Y. (2024). Comparative Analysis of LSTM and ARIMA Models in Stock Price Prediction: A Technology Company Example. *Black Sea Journal of Engineering and Science*. DOI:10.34248/bsengineering.1445997
- Lahboub, K., & Benali, M. (2024). Assessing the Predictive Power of Transformers, ARIMA, and LSTM in Forecasting Stock Prices of Moroccan Credit Companies. *Journal of Risk and Financial Management, 17*(7), 293. https://doi.org/10.3390/jrfm17070293
- Latif, N., Selvam, J. D., Kapse, M., Sharma, V. & Mahajan, V., (2023). Comparative Performance of LSTM and ARIMA for the Short-Term Prediction of Bitcoin Prices. *Australasian Accounting, Business and Finance Journal* 17(1), 256-276. doi: https://doi.org/10.14453/aabfj.v17i1.15
- Li, Z., Liao, Y., Hu, B., Ni, L., & Lu, Y. (2022). A financial deep learning framework: Predicting the values of financial time series with ARIMA and LSTM. *International Journal of Web Services Research, 19*(1), 1–15. https://doi.org/10.4018/IJWSR.302640
- Ma, Q. (2020). Comparison of ARIMA, ANN and LSTM for Stock Price Prediction. *E3S Web of Conferences, 218*, 01026. https://doi.org/10.1051/e3sconf/202021801026
- Mochurad, L., & Dereviannyi, A. (2024). An ensemble approach integrating LSTM and ARIMA models for enhanced financial market predictions. *Royal Society Open Science, 11*(11). https://doi.org/10.1098/rsos.240699
- Nasser A., Etem, T. (2021). Hourly energy consumption forecasting by LSTM and ARIMA methods. *Journal of Computer & Electrical and Electronics Engineering Sciences,* 1(1), 1-13.
- Pilla, P., & Mekonen, R. (2025). *Forecasting S&P 500 Using LSTM Models*. arXiv. DOI:10.48550/arXiv.2501.17366
- Putri, K. S., & Halim, S. (2020). Currency movement forecasting using time series analysis and long short-term memory. *International Journal of Industrial Optimization*, 1(2), 71-80.
- Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2018). A comparison of ARIMA and LSTM in forecasting time series. *Presented at the International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil*. https://par.nsf.gov/servlets/purl/10186768
- Siami-Namini, S., Tavakoli, N., & Siami Namin, A. S. (2019). *A comparative analysis of forecasting financial time series using ARIMA, LSTM, and BiLSTM* (arXiv preprint arXiv:1911.09512). arXiv.
- Trisya, C. P., Azani, N. W., Sari, L. M., Handayani, H., & Alhamid, M. R. M. (2024). Performance comparison of ARIMA, LSTM and SVM models for electric energy

consumption analysis. Public Research Journal of Engineering, Data Technology and Computer Science, 1(2), 85–94. https://doi.org/10.57152/predatecs.v1i2.869

- Wu, H., Chen, S., & Ding, Y. (2023). Comparison of ARIMA and LSTM for Stock Price Prediction. *Financial Engineering and Risk Management, 6*(1), 1–7. https://doi.org/10.23977/ferm.2023.060101
- Xiao, R., Feng, Y., Yan, L., & Ma, Y. (2022). Predict stock prices with ARIMA and LSTM. *arXiv preprint arXiv:2209.02407*.
- Yamak, P. T., Yujian, L., & Gadosey, P. K. (2019). A comparison between ARIMA, LSTM, and GRU for time series forecasting. In *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence* (pp. 49–55). Association for Computing Machinery. https://doi.org/10.1145/3377713.3377722
- Yanuar, N.H., Gibson, N.G.D., Sabilla, S.H. & Kartiasih, F. (2024). Comparison of ARIMA and LSTM Methods in Predicting Jakarta Sea Level. *Jurnal Ilmu dan Teknologi Kelautan Tropis, 16*(2), 163–178. https://doi.org/10.29244/jitkt.v16i2.52818
- Yavasani, R., & Wang, F. (2023). Comparative Analysis of LSTM, GRU, and ARIMA Models for Stock Market Price Prediction. *Journal of Student Research, 12*(4). https://doi.org/10.47611/jsrhs.v12i4.5888
- Zhang, Z. (2024). Comparison of LSTM and ARIMA in Price Forecasting: Evidence from Five Indexes. In F. Balli et al. (Eds.), *Proceedings of the 2023 2nd International Conference on Economics, Smart Finance and Contemporary Trade (ESFCT 2023), Advances in Economics, Business and Management Research 261*. https://doi.org/10.2991/978-94-6463-268-2_6
- Zheng, N., Huang, C., & Dufour, J.C. (2025). LSTM networks and ARIMA models for financial time series prediction. In S. Shiaeles & B. R. Auwal (Eds.), *Proceedings of the 5th International Conference on Signal Processing and Machine Learning* (pp. 56–63). https://doi.org/10.54254/2755-2721/134/2025.22270