Deep Learning for Financial Time Series: Does LSTM Outperform ARIMA and SVR in International Stock Market Predictions?

Abderrahman Yaakoub[1,*] , Hassan Oukhouya[2,1,*] , Mohamed Elhia[1] , Tarek Zari[1] , Raby Guerbaz[1]

[1] *MAEGE Laboratory, FSJES Aïn Sebaâ, Hassan II University, Casablanca, Morocco*
[2] *LaMSD Laboratory, Team of MSASE, FSJES, Mohammed First University, Oujda, Morocco*

Abstract   Time-series analysis and dynamic modeling are crucial in various fields, including business, economics, and finance. This study is based on the prediction of financial time series, which are known for their volatility, nonlinearity, and sensitivity to macroeconomic and psychological factors. This article examines four international stock market indices, such as MASI, S&P 500, CAC 40, and Nikkei 225, representing Africa, America, Europe, and Asia, respectively, which are challenging to model accurately. This research aims to compare three forecasting models: the classical Autoregressive Integrated Moving Average (ARIMA), the Machine Learning (ML) model Support Vector Regression (SVR) and the Deep Learning (DL) model Long-Short-Term Memory (LSTM). The empirical results reveal that LSTM outperforms both SVR and ARIMA in predicting financial time series; SVR outperforms ARIMA in twos indices: S&P 500, and CAC 40. In contrast, ARIMA outperforms SVR in the MASI and Nikkei 225 index, demonstrating the effectiveness of this traditional method in specific contexts.

Keywords   Financial time series, Forecasting, International stock market, ARIMA, SVR, LSTM.

1. Introduction

Time series analysis and dynamic modeling represent a fundamental and multidisciplinary field of research [1], offering a broad range of applications in diverse fields such as business, economics, finance, and computer science [15, 36]. In particular, stock market forecasting is essential in financial analysis and economic research, given its importance for investors, financial analysts, and economic decision-makers. It helps guide investment strategies, minimize financial risks, and improve decision-making at various levels. However, global financial markets are characterized by great complexity and increased volatility due to multiple factors, such as economic shocks, monetary policies, and geopolitical events. These unpredictable dynamics make the prediction task particularly challenging and require advanced analytical tools capable of adapting to the characteristics of financial time series [25]. In this context, several techniques have been developed to analyze and predict variations in stock prices. Traditional models, such as the Autoregressive Integrated Moving Average (ARIMA) introduced by Box and Jenkins [2], remain benchmark tools due to their robustness and ability to efficiently model linear time series structures [23, 8]. Although effective in many contexts, these methods are limited when capturing the nonlinear relationships and complex dependencies often present in stock market series [13, 29]. To overcome

---

*Correspondence to: Abderrahman Yaakoub (Email: abderrahman.yaakoub-etu@etu.univh2c.ma). Department of Statistics and Applied Mathematics, FSJES Aïn Sebaâ, Hassan II University, Casablanca, Morocco.
Hassan Oukhouya (Email: oukhouya.hassan@ump.ac.ma). Department of Economics, FSJES, Mohammed First University, Oujda, BV Mohammed VI B.P. 724 Oujda 60000 Morocco

these limitations, machine learning (ML) based approaches, such as support vector machines (SVMs), have been introduced, offer greater flexibility by directly learning complex relationships between data, enabling them to better adapt to the nonlinear characteristics of financial markets [19, 26]. Furthermore, the emergence of deep learning (DL) models, notably long-short-term memory (LSTM) networks, has marked a significant advance in time series modeling. These recurrent neural networks (RNN) are particularly effective at capturing long-term temporal dependencies and data nonlinearities, offering superior predictive capability [20, 21].

One significant challenge is selecting the model best suited to the different characteristics of stock market indices. Traditional models often perform well on stationary series, while artificial intelligence (AI)-based approaches better capture nonlinearities [37]. With this diversity of methods, a central question emerges: Which approach is the most effective for predicting the Moroccan All Shares Index (MASI), Standard and Poor's 500 (S&P 500), Nikkei 225, and CAC 40 indices?

Thus, we make the following assumptions: ML models, such as SVR and LSTM, will offer higher predictive accuracy than ARIMA models. However, their performance varies according to the specific features of each stock market index studied. Finally, a multicriteria analysis will reveal the advantages and limitations of each approach. In this investigation, the study evaluates the performance of three models, ARIMA, SVR, and LSTM, in forecasts of stock prices on four international markets: MASI (Morocco), S&P 500 (USA), Nikkei 225 (Japan), and CAC 40 (France). This research aims to identify the model that offers the best performance in terms of accuracy and robustness while considering the specific features of each financial market. This analysis, based on historical data for these stock market indices, aims to identify the strengths and limitations of each approach, offering a global perspective on their applicability and effectiveness in various economic and geographical contexts.

The rest of this article is organized as follows. Section 2 provides an overview of the latest work on stock index forecasting. Section 3 outlines the methodology employed, including models and evaluation criteria. The section 4 compares and analyzes model performance on the various indices studied. Finally, Section 5 summarizes the main results and suggests perspectives for future research.

2. Related works

Numerous studies have examined the forecasting of stock market indices, with particular attention paid to traditional models such as ARIMA and modern ML-based approaches such as SVR and DL, particularly LSTM. This section reviews previous work exploring the use of these models for financial market forecasting. ARIMA models have been widely utilised to forecast economic time series due to their simplicity and efficiency. For example, Ariyo et al. [8] applied ARIMA models to predict the stock prices of Nokia and Zenith Bank, demonstrating that they are effective for short-term forecasts. However, the accuracy decreases over longer horizons. Similarly, the authors of the paper [6] compared different ARIMA configurations to predict Netflix stock prices, concluding that the ARIMA(1,1,33) model offered the best precision with a Mean Absolute Percentage Error (MAPE) of 99.74%. However, they noted that ARIMA models have difficulty capturing sudden price fluctuations, particularly in volatile markets. In [5], a comparative study between ARIMA and LSTM models in forecasting the price of Apex Foods stock. Contrary to expectations, the results showed that the ARIMA model outperformed the LSTM in terms of Root Mean Square Error (RMSE) of 4.336 and Mean Absolute Error (MAE) of 3.4592. Suggests that classic models may still outperform modern methods. In addition, the authors of [23] used ARIMA models to predict Bursa Malaysia's closing prices during COVID-19. The ARIMA(2,1,2) model performed best among the configurations tested, but it also displayed signs of overfitting and an inability to keep up with sudden price changes.

On the other hand, ML models such as the SVR have been used because they can handle nonlinear data and detect complex relationships in financial time series. The study in [7] compares the ARIMA, artificial neural network (ANN), and SVR models in terms of their performance in predicting the Al-Quds stock market index in Palestine. The results proved that SVR gave the best accuracy, with an RMSE of 0.00703, which performs better than the ARIMA and ANN models. Likewise, Henrique et al. [17] utilized SVRs on various markets, such as Brazil, the United States, and China, to forecast stock prices. Their study demonstrated that SVRs with a linear

kernel surpassed ARIMA and random walk models, specifically for small capitalizations. As highlighted by [18], who presented a fine-tuned version of SVR to forecast stock prices in several business sectors. Their model showed a significant improvement in accuracy with reduced computation time over traditional SVRs. However, they pointed out that optimizing hyperparameters remains challenging, particularly for big datasets. This challenge was equally met by [10], who compared SVR with linear regression to predict Amazon stock prices. Although linear regression surpassed SVR in their investigation, the authors outline that SVR can be improved by combining hyperparameter optimisation techniques with other models.

In contrast, DL models like the LSTM have become valuable tools for capturing long-term dependencies in financial time series. In [3] proposed a simple Google stock price prediction technique using a DL RNN. The findings show an RMSE of 12.68% for the prediction with an accuracy of 87.32%, meaning that the projections are 87.32% similar to the actual Google stock prices. In the study carried out in [15], The authors examined the forecast performance of the ARIMA and LSTM models for several stock market indices, notably the S&P 500 and NASDAQ. The analysis revealed that LSTM minimizes by 84% to 87% compared to ARIMA, proving its ability to model complex time series. Chen et al. [4] pointed out that LSTM models can capture nonlinearity in emerging markets, noting that the forecast of stock market returns in China has been improved from 14.3% to 27.2%. Similarly, the study conducted in [12] adopted ARIMA and LSTM models to predict the financial budgets of a government organization. LSTM was better than ARIMA in terms of accuracy, even though its performance depends on hyperparameters optimization. This dependence has also been studied by [13], who carried out a comparative study between the LSTM, ARIMA, and gated recurrent unit (GRU) models for forecasting the price of stocks in various economic sectors. It shows that LSTM and GRU offer better performance than ARIMA regarding mean squared error (MSE), but this varies depending on the industry. Furthermore, the work in [22] predicts the price trends on the Swedish stock market using LSTM. The results showed the capacity to generate portfolios with higher returns and lower volatility than random portfolios. In [26], H. Oukhouya and K. El Himdi Conducted a comparative study of the ARIMA, SVR, and LSTM models for predicting the MASI index (Morocco), showed that LSTM outperformed ARIMA and SVR in terms of accuracy, but highlighted that optimizing LSTM's hyperparameters was still a challenge. In addition, Kalyan et al. [9] employed ARIMA, LSTM and random forest (RF) models to predict Tata Motors and Infosys stock prices. The findings revealed that LSTMs were better at capturing long-term dependencies but that ARIMAs were more reliable for some indices. In the same way, the study in [11] investigated the ARIMA, SVR, and LSTM models for predicting the S&P 500 and SSE (China) indices. The results confirmed that ARIMA was the best performer for short-term predictions, while LSTM was the best for long-term ones.

Based on existing studies, several limitations have been observed when using stock market forecasting models. Firstly, many studies concentrate on specific markets or indices, which limits the possibility of generalizing results to other markets or regions [23, 22]. Secondly, the performance of models such as LSTM and SVR is highly dependent on hyperparameter optimization and data quality, with persistent challenges related to noise, missing values, and temporal frequency [18, 26]. Third, some studies point to the increased sensitivity of traditional models, such as ARIMA, during high volatility, while DL approaches require massive data volumes and significant computational capabilities [24, 14]. Finally, few studies adopt an international [27, 17, 16], multisector perspective to validate the robustness of the models. The contribution of this paper is to fill these gaps by proposing a comparative analysis of the ARIMA, SVR, and LSTM models applied to four distinct international stock market indices: the MASI (Morocco), the S&P 500 (USA), the CAC 40 (France) and the Nikkei 225 (Japan). By evaluating these models in mature and emerging markets, characterized by varying economic and regulatory dynamics, this study aims to generalize the findings on the predictive effectiveness of the approaches while incorporating hyperparameter optimization and data preprocessing strategies to improve the reliability of the results. This multi-contextual process will enable us to identify the models best adapted to specific markets, giving investors and researchers practical information.

3. Data and Methodology

In this section, we present the methodology employed to compare the performance of stock market forecast models from data with suitable preprocessing, hyperparameter optimization, and evaluation using MAPE, RMSE, mean absolute error (MAE), and coefficient of determination (R^2).

3.1. Data

The data presented in this article represent the historical closing prices of the MASI, S&P 500, CAC 40, and Nikkei 225 stock indices. They were selected because they represent the stock markets of different economic regions.

- MASI (Morocco): Reflects the financial market of an emerging African country and offers an overview of regional economic dynamics and expanding markets.

- S&P 500 (USA): The benchmark index for the US market comprises the 500 largest listed companies and serves as a barometer for the global economy.

- CAC 40 (France): The main index of the Paris Stock Exchange (Euronext, Paris) includes France's 40 largest market capitalizations and serves as a barometer for the French and eurozone economies.

- Nikkei 225 (Japan): Key index for the Asian market, reflecting the performance of major Japanese companies and the impact of economic dynamics in Asia.

The study period runs from January 1, 2023, to July 31, 2024. Data were collected from finance.yahoo.com for the international indices S&P 500 and Nikkei 225, and from investing.com for the MASI and CAC 40 indices.

Table 1. Descriptive statistics for stock market indices

Index	Mean	Median	Std Dev	Min	Max	Skewness
MASI	11946.2	11977.3	1067.9	9717.99	13984.33	-0.128
S&P 500	4613.50	4499.38	492.06	3808.1	5667.2	0.396
CAC 40	7452.75	7382.49	359.44	6594.57	8239.99	0.513
Nikkei 225	33576.24	32851	4351.82	25716.8	42224.02	0.087

Table 1 shows the descriptive statistics of each index, revealing that the Nikkei 225 index is the most volatile, with a wide dispersion reflecting its sensitivity to economic shocks and that the MASI displays a slight negative skew, indicating a slightly left-biased distribution. While the S&P 500 and CAC 40 show a positive skew, suggesting a higher concentration of high values.

3.2. Trends in the stock market indices studied

As shown in Figure 1, the MASI, S&P 500, CAC 40, and Nikkei 225 stock indices show an overall uptrend between January 2023 and July 2024, with marked fluctuations and a notable slowdown or correction in October, reflecting the impact of geopolitical tensions on financial markets.

3.3. Stationarity Tests

Results presented in Table 2 show that the differentiated series is stationary according to Augmented Dickey-Fuller (ADF) and Phillips-Perron (PP) tests, as the null hypotheses of non-stationarity are rejected. The Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test accepts the null hypothesis of stationarity, which is consistent with the results of the other tests.
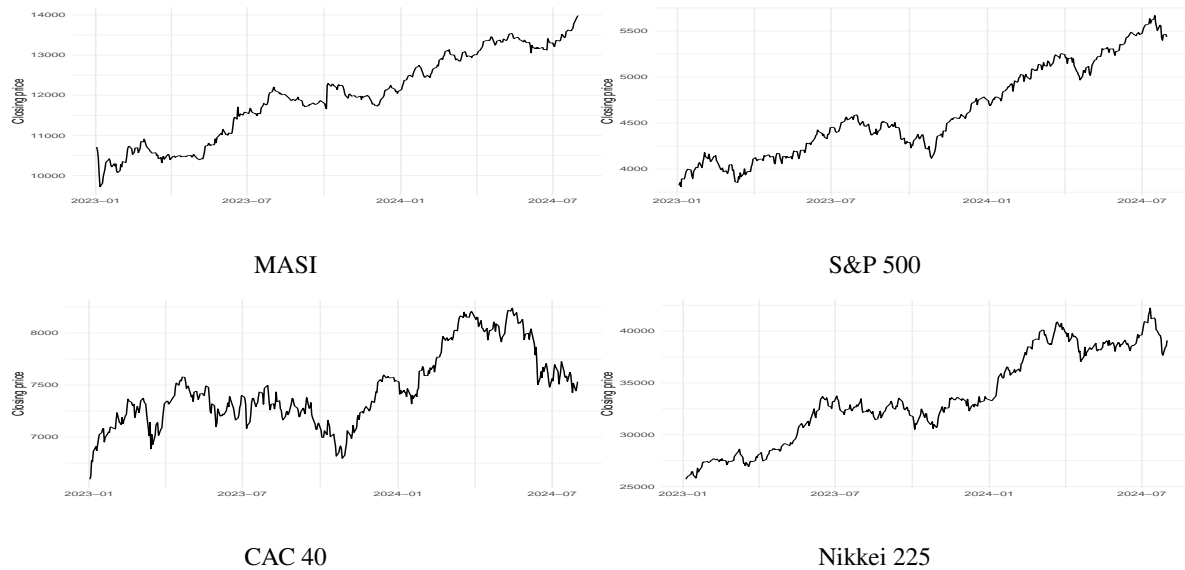
Figure 1. Historical of daily stock market indices studied

Table 2. Stationarity Tests for the first difference of closing price of MASI, S&P 500, CAC 40, and Nikkei 225 Indices

Test	Index	Test Statistic	p-value	Decision of H_0
ADF	MASI	-7.4505	0.01	Rejected
	S&P 500	-6.9115	0.01	Rejected
	CAC 40	-7.6090	0.01	Rejected
	Nikkei 225	-7.1782	0.01	Rejected
KPSS	MASI	0.08517	0.10	Accepted
	S&P 500	0.05010	0.10	Accepted
	CAC 40	0.16757	0.10	Accepted
	Nikkei 225	0.06550	0.10	Accepted
PP	MASI	-316.49	0.01	Rejected
	S&P 500	-355.08	0.01	Rejected
	CAC 40	-410.42	0.01	Rejected
	Nikkei 225	-405.06	0.01	Rejected

3.4. Preprocessing

Data preprocessing is essential in guaranteeing the relevance and quality of forecasting model results. It converts raw datasets to a format suitable for analysis, ensuring that it is consistent, comparable, and adapted to the algorithms' requirements. This phase includes operations such as normalizing values, splitting into training and test sets, and stationarity of time series where necessary. We normalized the time series to ensure better model convergence and avoid the domination of significant amplitude variables. We used the Min-Max transformation, which brings the values into the interval [0,1], defined by:

$$y_{\text{norm}}(t) = \frac{y(t) - y_{\min}}{y_{\max} - y_{\min}}, \tag{1}$$

where t represents each trading day, and $y(t)$ represents the raw input data, i.e., the daily closing prices. $y_{norm}(t) \in [0,1]$ denotes the normalized data values, while y_{max} and y_{min} respectively represent the Max and Min values of the entire time series. In addition, the time series were split into training and test sets in a ratio of 90%-10% as illustrated in table 3, ensuring that the models were trained on a significant portion of the data while reserving a portion for performance evaluation. However, there are no missing values in our time series.

Table 3. # of observations in the time series

Stock index	Observations		Train Date Range		Total Data	Frequency	#NA
	Train 90%	Test 10%	Start Date	End Date			
MASI	354	39	2023-01-02	2024-06-03	393	Daily	0
S&P 500	356	39	2023-01-03	2024-06-04	395	Daily	0
CAC 40	364	40	2023-01-02	2024-06-05	404	Daily	0
Nikkei 225	349	39	2023-01-04	2024-06-05	388	Daily	0

3.5. ARIMA model

The ARIMA model, the Box-Jenkins methodology (Figure 2), is a classic statistical tool for time series analysis and forecasting. It combines three fundamental components: an autoregressive (AR), an integrated (I) for stationarity, and a moving average (MA). ARIMA is particularly well suited to short-term forecasting [8, 15]; Its effectiveness lies in its ability to model univariate time series with trends or seasonality [6], provided they are made stationary [12]. ARIMA is defined by three parameters (p, d, q):
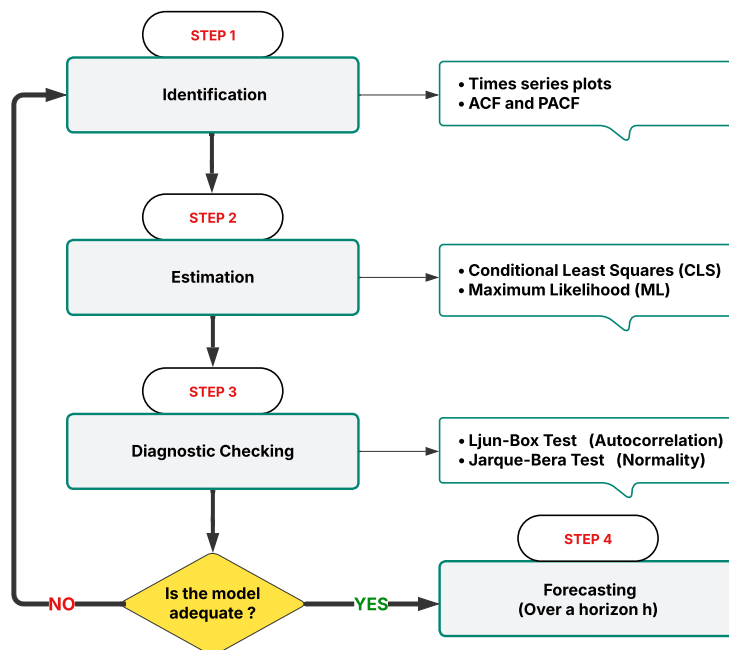

Figure 2. Box-Jenkins methodology

- AR(p): Captures the linear dependency between an observation and p lags. Formally:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t, \tag{2}$$

  where ϕ_i are the coefficients and ε_t the random error.

- I(d): Number of differentiations required to eliminate trends and make the series stationary (i.e., constant mean and variance).

- MA(q): Models past errors as a linear combination:

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}. \tag{3}$$

The general formulation of ARIMA is therefore:

$$\left(1 - \sum_{i=1}^{p} \phi_i B^i\right)(1 - B)^d y_t = \left(1 + \sum_{j=1}^{q} \theta_j B^j\right)\varepsilon_t, \tag{4}$$

where B is the lag operator (i.e., $By_t = y_{t-1}$). The following pseudocode algorithm 1 describes the training procedure for the ARIMA model adopted in our research.

Algorithm 1 Pseudocode for the ARIMA model (Box-Jenkins methodology)

1: **Step 1: Identification**
2: Check the stationarity of the series using the ADF, PP, and KPSS tests.
3: If non-stationary, differentiate the series $(1 - B)^d y_t$.
4: Examine the ACF and PACF plots to estimate the orders of p and q.
5: **Step 2: Estimation**
6: Estimate the parameters of the ARIMA(p, d, q) model by Maximum Likelihood (ML) or Conditional Sum of Squares (CSS).
7: Compare several candidate models using the AIC/BIC criteria.
8: **Step 3 : Diagnostic Checking**
9: Analyze residuals (independence, normality, homoscedasticity).
10: Apply the Ljung-Box test to verify the absence of autocorrelation.
11: Apply the Jarque-Bera/Shapiro-Wilk tests to verify the normality.
12: If the model is inadequate, return to step 1.
13: **Step 4 : Forecasting**
14: Using the validated model to forecast future values of the series.
15: Evaluation of performance using metrics (RMSE, MAE, MAPE, R^2).

3.6. SVR model

The SVR model extends the Support Vector Machines (SVM) adapted to regression problems. It is distinguished by its ability to model nonlinear relationships (Figure 3) through the use of kernels [17] and to minimize prediction error while controlling model complexity [19]. Its application in finance, notably for time series forecasting, is supported by several recent studies [7, 18]. The SVR seeks to find a function $f(x)$ that approximates the target y with a tolerance margin ε_t while penalizing deviations greater than this margin. Formally, the objective is to solve:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{n} (\xi_i + \xi_i^*), \tag{5}$$

under constraints :

$$\begin{cases} y_i - (w \cdot \phi(x_i) + b) \leq \varepsilon + \xi_i, \\ (w \cdot \phi(x_i) + b) - y_i \leq \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geq 0. \end{cases} \tag{6}$$
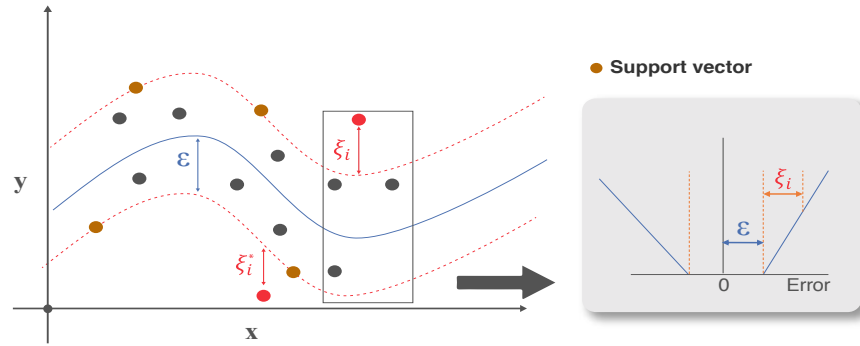
Figure 3. Nonlinear SVR

Where w and b are the model parameters, $\phi(x)$ is a projection function in a high-dimensional space via a kernel (linear, polynomial, radial basis function (RBF)), C is the regularization parameter that controls the trade-off between margin size and error tolerance, and ξ_i, ξ_i^* are the slack variables measuring the deviation beyond the ε-margin. The pseudo-algorithm 2 presents an SVR estimation and forecasting process of our study.

Algorithm 2 SVR estimation and forecasting process

1: Create lagged values L1 of the series.
2: Split the time series into a training set (90%) and a test set (10%).
3: Choosing a kernel function $K(\cdot)$ (linear, polynomial, or RBF).
4: Initialise hyperparameters (C, ε, γ).
5: Solving the SVR optimization problem :

$$\min_{w,b} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*),$$

under constraints :
$$y_i - \langle w, \phi(x_i) \rangle - b \leq \varepsilon + \xi_i, \quad \langle w, \phi(x_i) \rangle + b - y_i \leq \varepsilon + \xi_i^*.$$

6: Adjust (C, ε, γ) by trial-and-error method.
7: Select the model with the lowest RMSE and MAE on the training set.
8: Use the final model to predict values on the test set.
9: Evaluate the model by calculating performance metrics (RMSE, MAE, MAPE, R^2).

3.7. LSTM model

LSTM networks are specialized RNNs conceived to efficiently tackle long-term dependencies in sequential data [13]. They are exceptionally well adapted to tasks involving time-series data, such as stock market forecasts. LSTMs resolve a common problem in traditional RNNs: the vanishing gradient problem [30]. This problem arises when gradients become too small or too large as they are backpropagated in time, resulting in minimal modification of weights and rendering the learning process inefficient. LSTMs can store and refresh information over long sequences by including memory cells and control mechanisms, enabling them to learn and predict patterns in sequential data. The LSTM cell uses an input gate, a forget gate, and an output gate (a simple multilayer perceptron (MLP)). Depending on the data's priority, these gates define whether the data can pass through. The gates also

enable the network to learn what to save, what to forget, what to remember, what to pay attention to, and what to output. The cell and hidden states gather data for processing in the next state. The vanishing gradient can, therefore, be protected [28]. Figure 4 shows the structure of the nodes of the LSTM cell. At every time step t, an LSTM cell
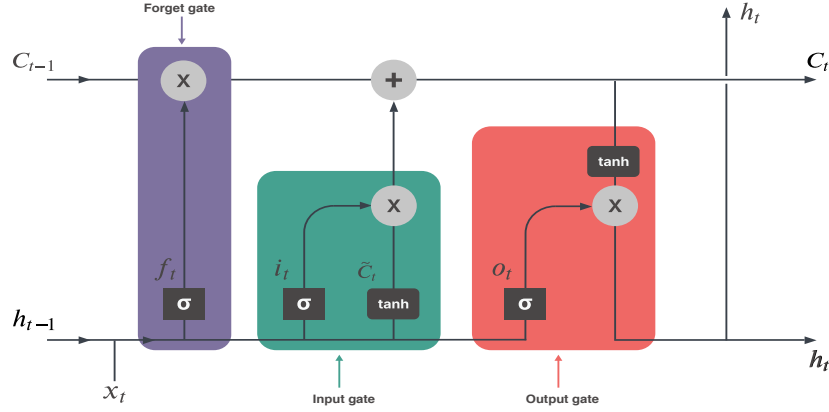

Figure 4. The internal structure of LSTM

has the following elements:

Forget Gate

Determines which information to remove from the cell state. The sigmoid activation function (σ) is applied to a linear combination of the preceding hidden state h_{t-1} and the present input x_t:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \tag{7}$$

where W_f and b_f represent the weight matrix and bias for the forget gate.

Input Gate and Candidate Cell State

The input gate controls which new information to be stored in the cell state. It is composed of the input gate i_t and the candidate cell state $\tilde{C}_t$, which represents new candidate values for the cell state:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{8}$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c), \tag{9}$$

where W_i, W_c, b_i, and b_c represent the weight matrix and bias for the input gate and candidate cell state, respectively.

Cell State Update

The cell state is updated by merging the preceding cell state C_{t-1}, scaled by the forget gate f_t, with the candidate cell state $\tilde{C}_t$, scaled by the input gate i_t:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t. \tag{10}$$

Output Gate and Hidden State

The output gate determines the value of the hidden state h_t. First, the output gate o_t is calculated by applying the sigmoid activation to a combination of the previous hidden state h_{t-1} and the current input x_t. Then, the hidden state h_t is obtained by applying the output gate to the updated cell state C_t:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{11}$$

$$h_t = o_t \cdot \tanh(C_t). \tag{12}$$

Where W_o and b_o represent the weight matrix and bias for the output gate, respectively. Control mechanisms (forget, input, and output gates) enable LSTM to retain or reject information selectively, allowing it to capture short and long-term dependencies in sequential data. The detailed procedure of the LSTM model is outlined in the following algorithm 3:

Algorithm 3 LSTM estimation and forecasting process

 1: Split the time series into a training set (90%) and a test set (10%).
 2: Normalize the data (e.g., Min-Max scaling).
 3: Create sequences using a moving window (X_t, y_t) adapted to the LSTM network.
 4: Define an LSTM architecture including :

- one or more LSTM layers,

- dense fully-connected layers, if necessary,

- a linear output layer.

 5: Initialize hyperparameters: number of neurons, window size, number of epochs, batch size, learning rate.
 6: Compile the model with a loss function (e.g., MSE) and an optimizer Adam.
 7: Train the model on the training set.
 8: Assess performance on a validation set.
 9: Adjust the hyperparameters (neurons, epochs, batch size) to reduce the error.
10: Use the final model to forecast on the test set.
11: Denormalize the predictions to return to the original scale.
12: Calculate key performance metrics (RMSE, MAE, MAPE, R^2).

3.8. Assumptions and implications

After presenting the three methods, namely ARIMA, SVR, and LSTM, it is useful to outline their main assumptions and the implications that arise from them. Table 4 presents the assumptions of each model and their impact in the event of a violation.

The summary in the table 4 shows that each model is built on specific assumptions that directly influence its ability to perform well and adapt to financial data. The ARIMA model is robust for linear relationships and stationary data; the SVR model offers flexibility when dealing with moderate nonlinearities. In contrast, the LSTM model is particularly suited to complex relationships and long-term dependencies.

3.9. Forecast Accuracy Measures

The evaluation of a prediction model is based on its forecast accuracy. To this end, several measures are used, which include the following:

Table 4. Model assumptions and the impact of their violation on forecasting accuracy

Model	Main assumptions	Effects in case of violation
ARIMA	- Stationarity of the series (or made stationary by differentiation) - Linear relationship between values in the series - The residues are uncorrelated and normally distributed	- Inaccurate model specification - Biased forecasts - The forecast intervals may be incorrect
SVR	- The features extracted represent the underlying structure well - The kernel parameters are selected appropriately - The data is free of excessive noise	- Overlearning or underlearning - Poor generalization to new data - Loss of accuracy in highly noisy series
LSTM	- Sufficient data volume for training - Time series include useful long-term dependencies - Hyperparameters are correctly tuned	- Inability to capture complex trends - Overlearning or underlearning - Unstable or slow convergence

- MAE: This calculates the mean error between the predicted and actual values, providing an easy-to-understand measure of model performance.

$$\text{MAE} = \frac{1}{n} \sum_{t=1}^{n} |\hat{y}_t - y_t|, \tag{13}$$

- RMSE: By giving more weight to larger errors, RMSE emphasizes substantial deviations, making it a useful metric when large errors are undesirable.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (\hat{y}_t - y_t)^2}, \tag{14}$$

- MAPE: This expresses forecast accuracy as a percentage, making it easier to interpret across different scales.

$$\text{MAPE} = \frac{100}{n} \sum_{t=1}^{n} \left| \frac{\hat{y}_t - y_t}{y_t} \right|. \tag{15}$$

Where $\hat{y}_t$ is the forecasted value, y_t is the actual value, and n is the number of observations. The authors of [31] stated that the model exhibiting the smallest value across all criteria is considered the most suitable for forecasting.

3.10. Criteria for Selecting the Optimal Model

When choosing the most appropriate model from several options, various criteria can be considered to select the best model [33]. The Bayesian Information Criterion (BIC) and the Akaike Information Criterion (AIC) are two essential criteria.

$$\text{BIC} = -2 \ln(l) + k \ln(n), \tag{16}$$

$$AIC = -2\ln(l) + 2k. \tag{17}$$

Where l is the model likelihood, k is the order for the moving average q plus the order for the autoregressive p (i.e., $k = q + p$), and n is the number of observations. Thus, the better model that gives the least value to both the BIC and AIC [33].

4. Results and discussions

All experiments were executed using a PC equipped with an Intel Core i5 processor running at 2.3 GHz, 8 GB of RAM, and a Windows 10 operating system. Training ARIMA models is particularly fast and inexpensive in terms of resources. In contrast, the effectiveness of SVR decreases as the number of features or observations increases, resulting in high training time and memory requirements [35]. As for LSTM model training, it requires more computing time due to its complexity and the numerous parameters to be optimized. This difference highlights the classic trade-off between forecast accuracy and computational cost. The study was conducted using the **RStudio** environment (version 4.4.1) by exploiting several specialized libraries such as `forecast` for the ARIMA model, `caret` and `e1071` for the SVR model, and `keras` with `tensorflow` for LSTM networks. The integration of these tools helped ensure the repeatability and efficiency of the experimental process.

4.1. Results of the ARIMA model

Estimation of the parameters of the ARIMA models for the MASI, S&P 500, CAC 40, and Nikkei 225 indices was carried out using the maximum likelihood estimation (MLE) method, which is widely used due to its efficacy in estimating parameters. The conditional sum of squares (CSS) method is also used, but the MLE method offers more accurate estimates, especially for large samples. Table 5 shows the results of the ARIMA(2,1,2) model fitted to the MASI index and the ARIMA(1,1,1) models fitted to the S&P 500, CAC 40, and Nikkei 225 indices. The estimated AR and MA coefficients, associated standard deviations, and p-values are presented. Furthermore, AIC, BIC, and maximum log-likelihood values are reported to assess the goodness of fit of each model. Analysis of the estimated coefficients shows varying dynamics between markets. The MASI shows strong momentum with partially corrective effects, reflecting a market memory that is marked but sensitive to shocks. The S&P 500 and Nikkei 225 have negative AR coefficients, which suggests a quick corrective reaction to past variations, consistent with the relative efficiency of these markets. Finally, the CAC 40 is characterized by high persistence of past dynamics, offset by a quasi-symmetrical correction mechanism in response to new information.

Table 5. Estimated ARIMA parameters for MASI, S&P 500, CAC 40, and Nikkei 225

Index	Model	Parameter	Estimate	SE	p-value	AIC	BIC
MASI	ARIMA(2,1,2)	ϕ_1	1.378	0.054	$< 0.001^{***}$	4063.93	4077.41
		ϕ_2	-0.794	0.047	$< 0.001^{***}$		
		θ_1	-1.351	0.062	$< 0.001^{***}$		
		θ_2	0.793	0.065	$< 0.001^{***}$		
S&P 500	ARIMA(1,1,1)	ϕ_1	-0.706	0.221	0.0014^{**}	3539.32	3545.07
		θ_1	0.753	0.198	$< 0.001^{***}$		
CAC 40	ARIMA(1,1,1)	ϕ_1	0.946	0.038	$< 0.001^{***}$	3991.02	3996.82
		θ_1	-0.964	0.033	$< 0.001^{***}$		
Nikkei 225	ARIMA(1,1,1)	ϕ_1	-0.934	0.064	$< 0.001^{***}$	5064.52	5070.23
		θ_1	0.903	0.073	$< 0.001^{***}$		

Note: * $p < 5\%$, ** $p < 1\%$, *** $p < 0.1\%$.

The models can therefore be described as in the following table 6, using estimated parameters:

Table 6. Estimated ARIMA models for MASI, S&P 500, CAC 40, and Nikkei 225

Index	Model	ARIMA Model Equation
MASI	ARIMA(2,1,2)	$\hat{y'}_t = 1.378y'_{t-1} - 0.794y'_{t-2} + \varepsilon_t - 1.351\varepsilon_{t-1} + 0.793\varepsilon_{t-2}$
S&P 500	ARIMA(1,1,1)	$\hat{y'}_t = -0.706y'_{t-1} + \varepsilon_t + 0.753\varepsilon_{t-1}$
CAC 40	ARIMA(1,1,1)	$\hat{y'}_t = 0.946y'_{t-1} + \varepsilon_t - 0.964\varepsilon_{t-1}$
Nikkei 225	ARIMA(1,1,1)	$\hat{y'}_t = -0.934y'_{t-1} + \varepsilon_t + 0.903\varepsilon_{t-1}$

Where $y'_t = y_t - y_{t-1}$ represents the first-order difference of the time series, and ε_t denotes the error term, the models describe the relationship between the present index values and their past values.

4.2. Results of the SVR model

This research employed several SVR models in the training set to predict the closing prices of the MASI, S&P 500, CAC 40, and Nikkei 225 indices. The kernel types tested included linear, polynomial, and RBF. After evaluating the performance of each model, the following kernels were selected for each index based on the data results. The linear kernel was selected for the MASI (Figure 5) and S&P 500 (Figure 6) indices because of their effectiveness in modeling linear relationships. In contrast, the polynomial kernel was selected for the CAC 40 (Figure 7) and Nikkei 225 (Figure 8) indices to capture complex nonlinear patterns.
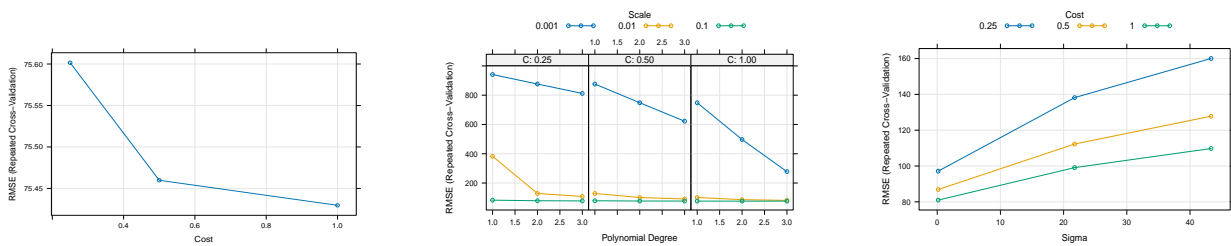

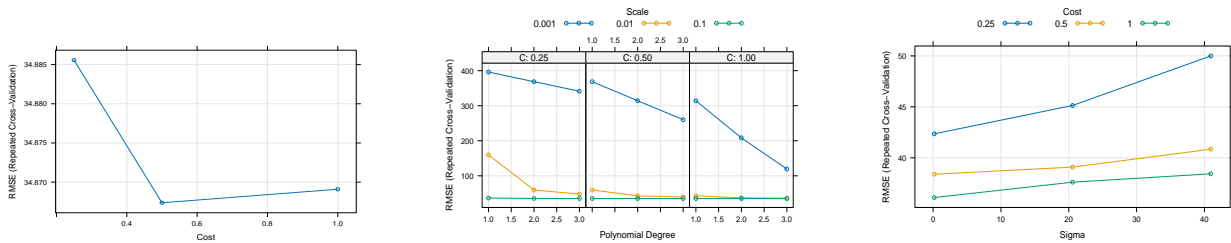Figure 5. MASI : SVR with different kernels fitting parameters

Figure 6. S&P 500 : SVR with different kernels fitting parameters

The main features of the selected models for each index are summarized in the table 7.

The parameters C, the degree of the polynomial kernel, and the scale were adjusted after several trials to ensure a suitable model generalization.
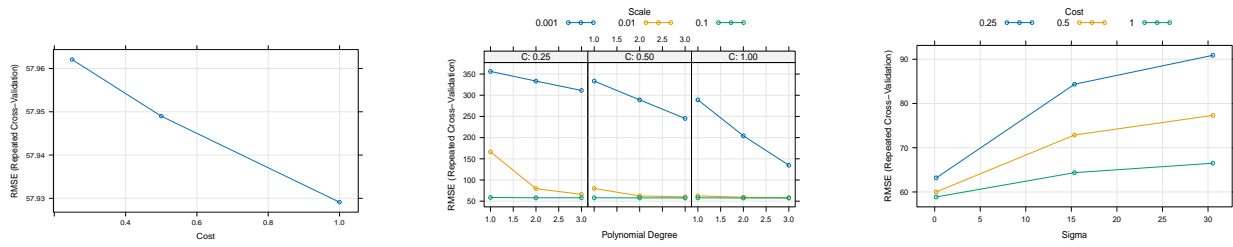
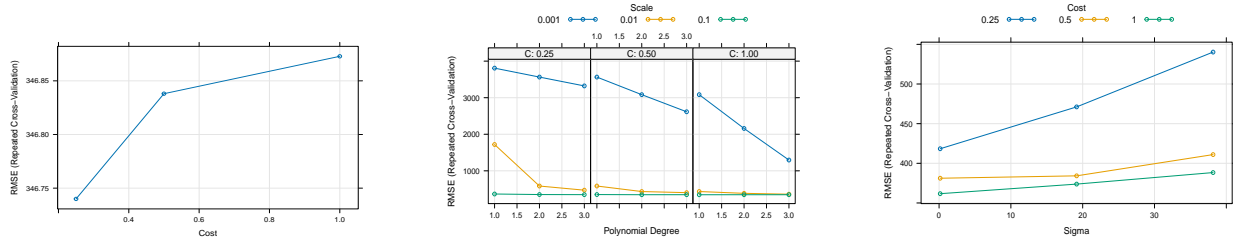Figure 7. CAC 40 : SVR with different kernels fitting parameters

Figure 8. Nikkei 225 : SVR with different kernels fitting parameters

Table 7. Performance of SVR model with different kernel types

Index	Kernel	Parameters	Cross-Validation	RMSE	MAE	R^2
MASI	Linear	C (Cost)	10-fold CV, repeated 10 times	75.43	53.11	0.9944
	Polynomial	C, Degree, Scale	10-fold CV, repeated 13 times	76.32	53.83	0.9943
	RBF with Sigma	Sigma, C	10-fold CV, repeated 13 times	80.86	56.11	0.9937
S&P 500	Linear	C (Cost)	10-fold CV, repeated 10 times	34.87	27.63	0.9934
	Polynomial	C, Degree, Scale	10-fold CV, repeated 10 times	34.96	27.65	0.9934
	RBF with Sigma	Sigma, C	10-fold CV, repeated 15 times	36.10	28.75	0.9929
CAC 40	Linear	C (Cost)	13-fold CV, repeated 10 times	57.93	44.32	0.9764
	Polynomial	C, Degree, Scale	13-fold CV, repeated 5 times	57.52	44.04	0.9765
	RBF with Sigma	Sigma, C	13-fold CV, repeated 10 times	58.87	44.44	0.9759
Nikkei 225	Linear	C (Cost)	10-fold CV, repeated 12 times	346.74	267.66	0.9930
	Polynomial	C, Degree, Scale	10-fold CV, repeated 12 times	346.03	267.17	0.9930
	RBF with Sigma	Sigma, C	10-fold CV, repeated 12 times	361.70	286.97	0.9923

4.3. Results of the LSTM model

We have defined and adjusted the hyperparameters for each analyzed index to model the financial indices (MASI, S&P 500, CAC 40, and Nikkei 225) using LSTM. We started by setting the architecture's initialization standards, and then we adjusted the hyperparameters using an empirical trial-and-error method. In order to reduce validation error, this method has involved testing multiple parameter combinations. The primary hyperparameters include the batch size, set to 1 in all cases, and the number of units in the LSTM layer, which varies by index. The models are constructed with a sequential architecture consisting of LSTM, Dense, and Dropout layers. We utilized the *tanh* activation function for both the LSTM and Dense layers. The tanh function is defined as $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$. The models are compiled with the Adam optimizer [34] and a loss function of Mean Squared Error. They are trained for 50 epochs, with some data reserved for validation. The table 8 summarizes the configurations for each index:

Table 8. Summary of LSTM model configurations for each index

Index	Units	Total Params	Epochs	Validation Split	Activation Function
MASI	5	146	50	0.1	tanh
S&P 500	3	64	50	0.1	tanh
CAC 40	5	146	50	0.1	tanh
N225	5	146	50	0.1	tanh

4.4. Comparative analysis

In this comparative analysis of the ARIMA, SVR, and LSTM models applied to the forecasting of the MASI, S&P 500, CAC 40, and Nikkei 225 stock market indices, the results obtained show a clear superiority of the LSTM model over all the indices studied. Indeed, according to the evaluation metrics used (MAE, RMSE, MAPE and R^2), LSTM offers the most accurate performance in forecasting, better capturing trends, and the nonlinear dynamics of time series. Furthermore, the SVR model outperforms ARIMA in three indices (S&P 500, CAC 40, and Nikkei 225), confirming its ability to model complex relations between previous and future observations. However, for the MASI index, ARIMA outperformed SVR, suggesting that, in some contexts, statistical models can still offer competitive results over machine learning approaches.

Table 9. Comparison of performance metrics of models on train and test sets for different indices

Models	MAE		RMSE		MAPE (%)		R^2	
	Train	Test	Train	Test	Train	Test	Train	Test
MASI								
ARIMA	47.64	56.74	70.99	82.28	0.411	0.423	0.9949	0.8617
SVR	53.28	58.65	78.03	84.11	0.465	0.437	0.9939	0.8555
LSTM	46.65	54.88	70.64	78.76	0.406	0.410	0.9950	0.8686
S&P 500								
ARIMA	27.23	28.05	34.85	37.01	0.610	0.510	0.9930	0.8139
SVR	27.57	27.71	34.91	37.76	0.619	0.504	0.9930	0.8153
LSTM	18.22	19.54	23.31	27.50	0.408	0.356	0.9968	0.9112
CAC 40								
ARIMA	44.58	68.15	58.48	79.24	0.605	0.894	0.9751	0.6545
SVR	43.79	68.71	58.06	78.14	0.594	0.901	0.9755	0.6640
LSTM	42.51	64.11	55.57	72.58	0.577	0.840	0.9882	0.7715
Nikkei 225								
ARIMA	268.10	333.57	346.38	457.76	0.809	0.842	0.9926	0.8395
SVR	265.53	350.90	346.36	458.56	0.800	0.886	0.9927	0.8389
LSTM	221.68	284.08	287.51	385.73	0.668	0.718	0.9949	0.8860

The detailed results are presented in Table 9, which groups the values of the various evaluation metrics for each model and each index. Furthermore, the predictive performance of the models is illustrated in Figures 9, 10, 11, and 12, where the prediction curves obtained are superimposed on the actual values for each index, allowing clear visualization of the differences between models. These observations confirm the growing interest in approaches based on RNN, notably LSTM, for the analysis and prediction of financial time series.
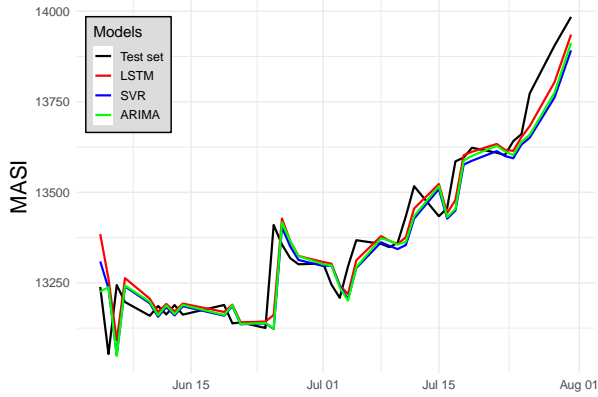
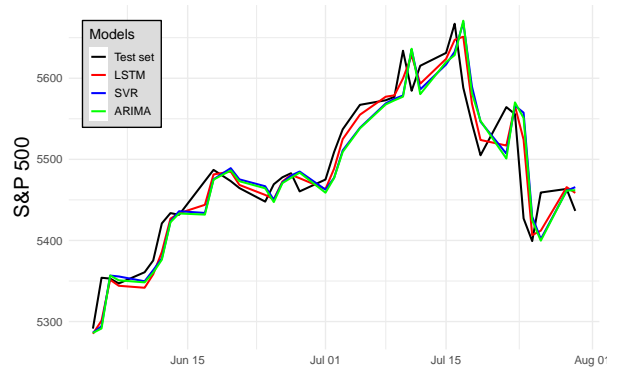Figure 9. Comparison of all models for MASI prediction

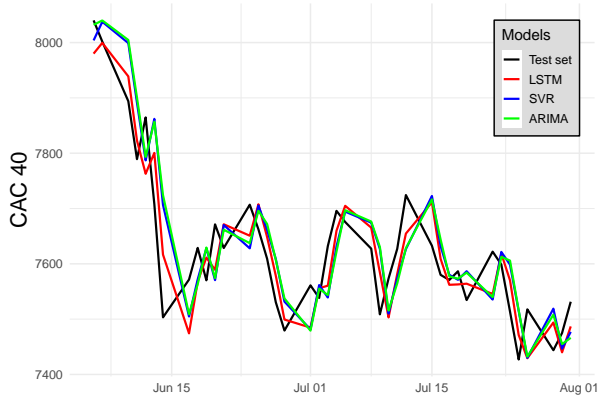Figure 10. Comparison of all models for S&P 500 prediction

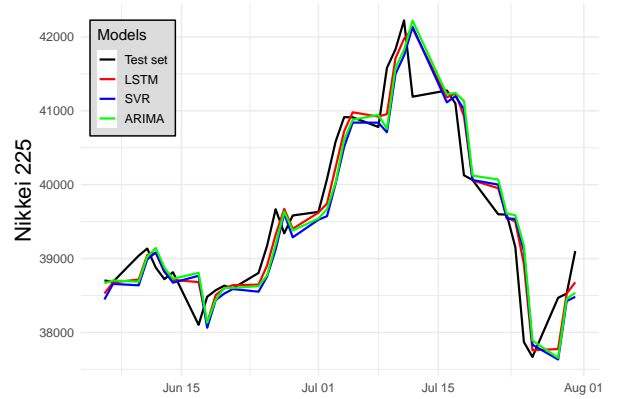Figure 11. Comparison of all models for CAC 40 prediction

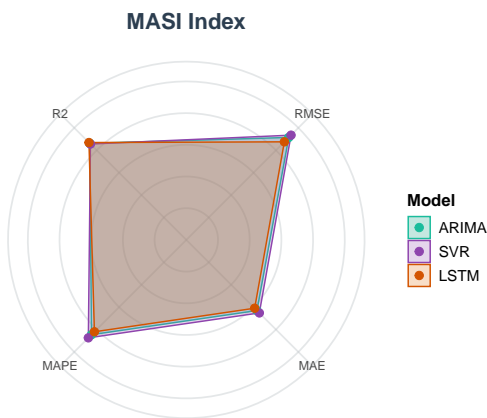Figure 12. Comparison of all models for Nikkei 225 prediction
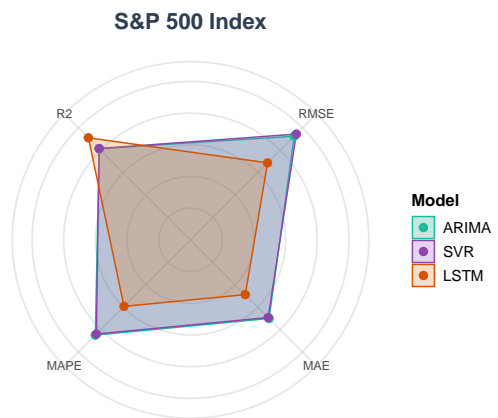

Figure 13. MASI benchmark radar chart

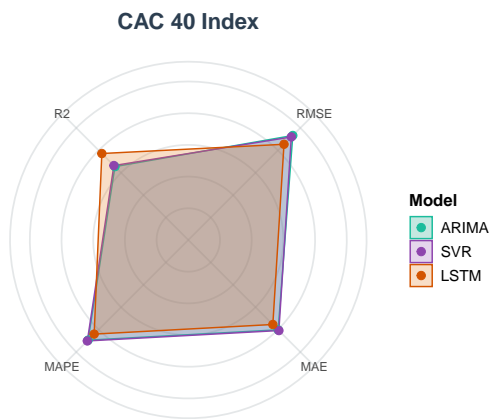Figure 14. S&P 500 benchmark radar chart
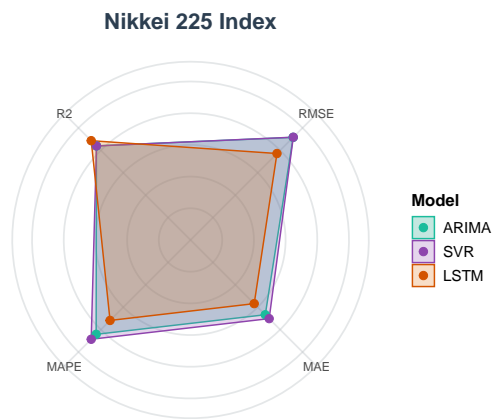
Figure 15. CAC 40 benchmark radar chart

Figure 16. Nikkei 225 benchmark radar chart

According to the radar charts in Figures 13, 14, 15, and 16, LSTM consistently outperforms the four indices, confirming its superiority in financial forecasting. However, the relative performance of the ARIMA and SVR models differs depending on the market: SVR performs better on the CAC 40 and S&P 500, while ARIMA remains competitive on the MASI and Nikkei 225, revealing the influence of market-specific characteristics for test sets.

5. Conclusion

Forecasting financial time series, particularly stock market indices, is challenging due to their volatility, nonlinearity, and sensitivity to macroeconomic and psychological factors. This study compared the effectiveness of three forecasting models, ARIMA, SVR, and LSTM, applied to the MASI, S&P 500, CAC 40, and Nikkei 225 stock indices from January 2023 to July 2024. The main objective was determining which model offers the best ability to forecast market fluctuations by applying the Box-Jenkins methodology to construct the ARIMA models. The results of the performance measures MAE, RMSE, MAPE, and R^2 showed that the LSTM DL model outperformed the other two models, particularly its ability to capture complex patterns and long-term dependencies. However, a significant limitation was observed for the CAC 40 index, where LSTM did not deliver the expected results. This is probably due to using a trial-and-error approach to hyperparameter selection, which is only sometimes optimal. A more systematic approach like grid search could improve LSTM's performance on series like the CAC 40. In addition, the classic ARIMA model outperformed the SVR ML model for MASI and Nikkei 225 stock indices, demonstrating the effectiveness of this traditional method in certain specific contexts. Although generally presented in the literature, SVR outperforms ARIMA in similar studies. This result suggests that the particular characteristics of each time series can strongly influence the performance of the forecasting models. Even if the results obtained show that the LSTM, SVR and ARIMA models are effective in predicting stock market indices, it is important to note that any prediction error can have a significant impact in a real-life context, particularly for investment or portfolio management decisions [32]. In future research, we envisage exploring the multivariate approach by incorporating additional factors influencing stock index fluctuations, including macroeconomic indicators and sentiment analysis. We will also explore advanced hybrid models and deep architectures, such as BiLSTM (Bidirectional LSTM) and Stacked LSTM, to improve prediction accuracy and better capture the interactions between different variables.

Availability of data and materials

The dataset used and/or analysed during the current research are available on the CSE and Yahoo Finance website: https://www.casablanca-bourse.com/en, https://finance.yahoo.com/

Competing interests

The authors have no conflicts of interest or personal relationships that could have biased the research.

REFERENCES

1. R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications*, 3rd ed., Springer, 2000.
2. G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*, John Wiley & Sons, 2015.
3. K. Ullah, and M. Qasim, *Google stock prices prediction using deep learning*, 2020 IEEE 10th International Conference on System Engineering and Technology (ICSET), pp. 108–113, 2020, IEEE.
4. K. Chen, Y. Zhou, and F. Dai, *A LSTM-based method for stock returns prediction: A case study of China stock market*, in *2015 IEEE International Conference on Big Data (Big Data)*, pp. 2823–2824, 2015.
5. I. Ahammad, W. A. Sarkar, F. A. Meem, J. Ferdus, M. K. Ahmed, M. R. Rahman, R. Sultana, and M. S. Islam, *Advancing stock market predictions with time series analysis including LSTM and ARIMA*, *Cloud Computing and Data Science*, pp. 226–241, 2024.
6. S. Khan and H. Alghulaiakh, *ARIMA model for accurate time series stocks forecasting*, *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 7, 2020.
7. M. K. Okasha, *Using support vector machines in financial time series forecasting*, *International Journal of Statistics and Applications*, vol. 4, no. 1, pp. 28–39, 2014.
8. A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, *Stock price prediction using the ARIMA model*, in *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, pp. 106–112, 2014.
9. B. Kalyan, S. P. Reddy, K. K. Kumari, and M. Jain, *Comparative analysis of stock price prediction accuracy: A machine learning approach with ARIMA, LSTM, and random forest models*, *International Research Journal on Advanced Engineering Hub (IRJAEH)*, vol. 2, no. 05, pp. 1141–1151, 2024.
10. B. Panwar, G. Dhuriya, P. Johri, S. S. Yadav, and N. Gaur, *Stock market prediction using linear regression and SVM*, in *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pp. 629–631, 2021.
11. S. Jin, *A comparative analysis of traditional and machine learning methods in forecasting the stock markets of China and the US*, *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 4, 2024.
12. M. Rhanoui, S. Yousfi, M. Mikram, and H. Merizak, *Forecasting financial budget time series: ARIMA random walk vs LSTM neural network*, IAES International Journal of Artificial Intelligence, vol. 8, no. 4, pp. 317, 2019.
13. R. Yavasani, and H. Wang, *Comparative Analysis of LSTM, GRU, and ARIMA Models for Stock Market Price Prediction*, Journal of Student Research, vol. 12, no. 4, 2023.
14. K. Sako, B. N. Mpinda, and P. C. Rodrigues, *Neural networks for financial time series forecasting*, Entropy, vol. 24, no. 5, p. 657, 2022, MDPI.
15. S. Siami-Namini, N. Tavakoli, and A. S. Namin, *A comparison of ARIMA and LSTM in forecasting time series*, Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1394–1401, 2018, IEEE.
16. S. Siami-Namini, N. Tavakoli, and A. S. Namin, *A comparative analysis of forecasting financial time series using ARIMA, LSTM, and BiLSTM*, arXiv preprint arXiv:1911.09512, 2019.
17. B. M. Henrique, V. A. Sobreiro, and H. Kimura, *Stock price prediction using support vector regression on daily and up to the minute prices*, The Journal of Finance and Data Science, vol. 4, no. 3, pp. 183–201, 2018, Elsevier.
18. R. K. Dash, T. N. Nguyen, K. Cengiz, and A. Sharma, *Fine-tuned support vector regression model for stock predictions*, Neural Computing and Applications, vol. 35, no. 32, pp. 23295–23309, 2023, Springer.

19. A. Kurani, P. Doshi, A. Vakharia, and M. Shah, *A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting*, Annals of Data Science, vol. 10, no. 1, pp. 183–208, 2023, Springer.
20. S. Mehtab, J. Sen, and A. Dutta, *Stock price prediction using machine learning and LSTM-based deep learning models*, Proceedings of the Second Symposium on Machine Learning and Metaheuristics Algorithms, SoMMA 2020, pp. 88–106, 2021, Springer.
21. A. Yenireddy, M. S. Narayana, K. V. B. Ganesh, G. P. Kumar, and M. Venkateswarlu, *Stock market index prediction based on market trend using LSTM*, Indonesian Journal of Electrical Engineering and Computer Science, vol. 35, no. 3, pp. 1601–1609, 2024.
22. C. Fjellström, *Long short-term memory neural network for financial time series*, Proceedings of the 2022 IEEE International Conference on Big Data (Big Data), pp. 3496–3504, 2022, IEEE.
23. M. K. Ho, H. Darman, and S. Musa, *Stock price prediction using ARIMA, neural network and LSTM models*, Journal of Physics: Conference Series, vol. 1988, no. 1, p. 012041, 2021, IOP Publishing.
24. A. Chatterjee, H. Bhowmick, and J. Sen, *Stock price prediction using time series, econometric, machine learning, and deep learning models*, Proceedings of the 2021 IEEE Mysore Sub Section International Conference (MysuruCon), pp. 289–296, 2021, IEEE.
25. S. Mehtab, J. Sen, and S. Dasgupta, *Robust analysis of stock price time series using CNN and LSTM-based deep learning models*, Proceedings of the 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 1481–1486, 2020, IEEE.
26. H. Oukhouya, and K. E. Himdi, *A comparative study of ARIMA, SVMs, and LSTM models in forecasting the Moroccan stock market*, International Journal of Simulation and Process Modelling, vol. 20, no. 2, pp. 125–143, 2023, Inderscience Publishers (IEL).
27. H. Oukhouya, and K E. Himdi. *Machine Learning Models-Based Forecasting Moroccan Stock Market*, INTERNATIONAL CONFERENCE ON LOGISTICS OPERATIONS MANAGEMENT. Cham: Springer Nature Switzerland, 2024.
28. P. T. Yamak, Y. Li, and P. K. Gadosey, *A comparison between ARIMA, LSTM, and GRU for time series forecasting*, Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence, pp. 49–55, 2019.
29. H. Oukhouya, A. Lmakri, M. El Yahyaoui, R. Guerbaz, S. El Melhaoui, M. Faizi, and K. El Himdi, *Predictive modeling for the Moroccan financial market: a nonlinear time series and deep learning approach*, Future Business Journal, vol. 11, no. 1, 2025.
30. S. Hochreiter, *Long Short-Term Memory*, Neural Computation, MIT Press, 1997.
31. T.-H. Chan, L. C. Teck, and C.-W. Hooy, *Forecasting Malaysian ringgit: before and after the global crisis*, AAMJAF, vol. 9, no. 2, pp. 157–175, 2013.
32. A. El Rhiouane, H. Oukhouya, R. Guerbaz, K. Belkhoutout, A. Lmakri, M. Fihri, and A. El Afia, *Integrating ESG criteria in portfolio optimization: A Moroccan case study using Markowitz's theory and correlation network analysis*, Physica A: Statistical Mechanics and its Applications, vol. 667, 2025.
33. Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2015). *Introduction to time series analysis and forecasting*. John Wiley & Sons.
34. Kingma D. P., Ba J., and others, *A method for stochastic optimization*, arXiv preprint arXiv:1412.6980, vol. 1412, no. 6, 2014.
35. L. Mochurad, and A. Dereviannyi, *An ensemble approach integrating LSTM and ARIMA models for enhanced financial market predictions*, Royal Society Open Science, vol. 11, no. 9, p. 240699, 2024.
36. C. Bekkaye, H. Oukhouya, T. Zari, R. Guerbaz, and H. El Bouanani, *Advanced Strategies for Predicting and Managing Auto Insurance Claims Using Machine Learning Models*, Statistics, Optimization & Information Computing, vol. 14, no. 3, 2025.
37. A. Hade and M. Elhia, *Predicting mortgage credit defaults in Morocco using machine learning approaches*, Discover Artificial Intelligence, vol. 5, no. 1, 2025.