Comparing Machine Learning Methods –SVR, XGBoost, LSTM, and MLP- in Forecasting the Moroccan Stock Market

Hassan OUKHOUYA[*], Khalid EL HIMDI[*]

The 1st International Online Conference on Mathematics and Applications
A Celebration of the 10th Anniversary of Mathematics' Impact on Our Wellbeing
01–15 May 2023 | Online
Chaired by Prof. Dr. Francisco Chiclana

[*]Laboratory LMSA, Department of Mathematics, Faculty of Sciences (FSR), Mohammed V University (UM5) in Rabat, Morocco
(March 14, 2023)

Roadmap

Problem Overview

? Why predict the stock market?

- Investment
- Maximize profits
- Predict the economy
- Implement suitable economic policies

Challenges

- Stochastic nature
- Multiple factors
- Market Complexity
- Data Quality

Related Works

Literature reviews

The literature review examined studies on stock market prediction using Machine Learning (ML) models. It concluded that Deep Learning (DL) was the most commonly utilized model for forecasting stock price trends [11, 13].

In the state of the art, the articles provide a comparison of some of them:

- Al-Nefaie et al. [12] used LSTM and MLP models to predict fluctuations in the Saudi stock market. Their research showed that the correlation coefficient between the two models was more significant than 0.995, and the LSTM model was more accurate and had the best fit.
- In [3], the authors evaluated the performance of models such as ARIMA, XGBoost, and LSTM in forecasting stock market trends using various measures such as MSE, MAE, RMSE, and MAPE. Their study concluded that XGBoost outperformed the other models.

Research Purpose

Aim of the study

The main objective of this investigation is to improve the Grid Search (GS) optimization algorithm by optimizing the hyperparameters of ML models.

The paper aims to contribute to the stock market prediction field by improving the performance of ML models through hyperparameter optimization.

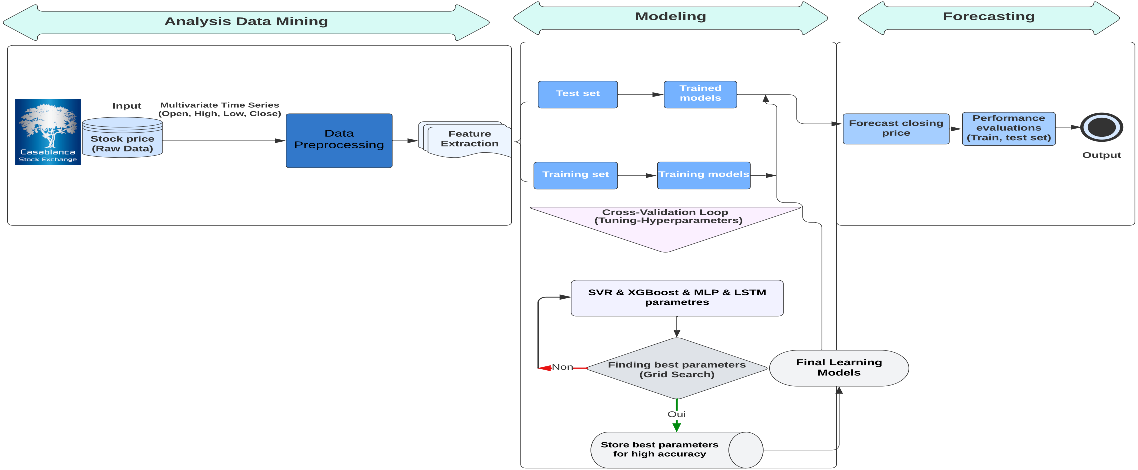
Table 1: Some previous work using ML models for stock market forecasting.

Reference	Location	Year	Benchmark Model	MAPE (%)	RMSE	R^2
[7]	Irlande	2019	LSTM, SVR, MA	1.03	347.46	0.830
[10]	China	2020	MLP, CNN, RNN, LSTM, CNN-RNN	—	39.688	0.965
[3]	American	2022	ARIMA, XGBoost, LSTM	3.8	6.101	0.961
This study	**Morocco**	**2023**	**SVR, XGBoost, MLP, LSTM**	**0.368**	**3.993**	**0.989**

Roadmap

Flowchart of this research

We describe a methodology for modelling and forecasting the closing price of MSI 20 using Python in an empirical study. The main steps of the process are :



Data : MOROCCO STOCK INDEX 20 (MSI 20)

- We chose MSI 20, a new index of the 20 most liquid companies listed on the CSE (Maroc Telecom, Attijariwafa Bank, BCP, LafargeHolcim Maroc, Addoha...etc.)
- The MSI 20 is a price index that takes into account market capitalization and captures 83% of the capitalization of Casablanca and 87% of the total trading volume, it covers 13 sectors of the economy.

This research uses a dataset with 4 variables (Open, Low, High, and Close prices) and 541 daily observations, spanning from 18 December 2020 to 09 February 2023. The focus is on using the stock prices of the MSI 20 index for modelling and prediction, as it reflects all daily activities of the index.

Date	Open	High	Low	Close
2020-12-18	946.25	950.56	943.85	944.58
⋮	⋮	⋮	⋮	⋮
2023-02-09	860.11	869.08	858.06	865.69

Statistics	Open	High	Low	Close
Count	541	541	541	541
Mean	988.86	993.28	984.91	988.84
Std	78.01	77.05	78.35	78.15
Min	775.38	797.01	775.38	775.38
Max	1140.69	1142.56	1135.86	1140.69

Preprocessing

To train model parameters, we use prices as Input = (Open, High, Low, and Closing prices) of length N,

$$X = \{X_t^{(1)}, X_t^{(2)}, X_t^{(3)}, \quad t = 1, 2, \ldots, N\},$$

For the output with a single observation sequence, we use only closing price,

$$Y = Y_t, \quad t = 1, 2, \ldots, N$$

The data set was divided into 90% of the observations used for training and 10% for model evaluation during testing. In this case, we normalized using a min-max scale, which scales all variables to a range of $[0, 1]$:

$$\widetilde{X}_t = \frac{X_t - X_t^{min}}{X_t^{max} - X_t^{min}} \tag{1}$$

SVR model for financial time series

The SVR model, a new financial time series prediction method, is used to address the challenges of nonlinear regression. Mathematically, this can be explained by the equation shown below:

$$Y_t = WX_t + b = \sum_{i=1}^{N} w_i x_{it} + b_i, \quad \xhookrightarrow[mapping]{Non-linear} \quad Y_t = \sum_k \beta_k y_k \mathcal{K}(x_k, x) + b \tag{2}$$

The most commonly used kernels include linear, gaussian, and polynomial functions [14].

XGBoost model for financial time series

XGBoost is an ML model used for stock market time series forecasting that uses a set of decision trees [1]. A gradient descent algorithm guides the process of preparing subsequent trees to minimize the loss function of the last tree [20].

$$\mathcal{L}_T(F(x_i)) = \sum_{i=1}^{N} \chi(y_i, F_T(x_i)) + \sum_{t=1}^{T} \Pi(f_t), \tag{3}$$

where $\chi(\cdot)$ is a loss function specified that quantifies the deviations of the predicted and actual target values, $\Pi(\cdot)$ represents the term of regularization, which penalizes the model complexity [18].

MLP model for financial time series

The MLP is a frequently used ANN consisting of three layers of neurons: The inputs (x_i) are multiplied by their weights (w_i), and the resulting products are combined. This sum and a bias term (b) are fed into an activation function to produce the neuron's output (Y_t) [5]. Equation (2) can be used to express this process in mathematical words:

$$\zeta_t = \sum_{i=1}^{N} w_i x_{it} + b \Longrightarrow Y_t = \sigma(\zeta_t), \tag{4}$$

$$\sigma(\zeta_t) = \frac{1}{1 + e^{-\zeta_t}} \tag{5}$$

where $\sigma(\cdot)$, the activation function, is frequently employed as a function, either continuous or discontinuous, that maps real numbers to a specific interval. Alternatively, the sigmoidal activation function can also be utilized [12].

LSTM model for financial time series

LSTM models are RNNs that excel at learning and retaining long-term dependencies, making them successful in various applications such as financial time series forecasting. The principle of an LSTM cell consists of the following four equations [17]:

- **Forget gate:**

$$f_t = \delta(W_{ef}(X_t, h_{t-1}) + W_{cf}C_{t-1} + b_f), \tag{6}$$

- **Input gate:**

$$i_t = \delta(W_{ki}(X_t, h_{t-1}, C_{t-1}) + b_i), \quad k = x, h, c \tag{7}$$

- **Memory cell:**

$$C_t = f_t C_{t-1} + i_t \tanh(W_{lc}(X_t, h_{t-1}) + b_c), \tag{8}$$

- **Output gate:**

$$o_t = \delta(W_{ji}(X_t, h_{t-1}, C_{t-1}) + b_o), \quad j = x, h, c \tag{9}$$

The GS hyper-parameter sets for the ML models

ML Models	Parameters	Type	Search Space
	Epsilon in the SVR loss function (epsilon)	Continuous	[0.0001,**0.001**,0.01]
	Regularization parameter (C)	Discrete	[100,**1000**,1100]
SVR	The kernel type (kernel)	Categorical	['**linear**', 'poly', 'rbf', 'sigmoid']
	Kernel coefficient (gamma)	Continuous	[**1e-5**,1e-4,1e-3]
	Tolerance for stopping criterion (tol)	Continuous	default=1e-3
	# of regression trees (n_estimators)	Discrete	[100,**200**,1000]
	Maximum regression tree depth (max_depth)	Discrete	[5,10,**15**,20]
	Boosting the rate of learning (learning_rate)	Continuous	[0.01, **0.06**, 0.09]
XGBoost	Minimum reduction of loss (gamma)	Continuous	[0.001,**0.01**,0.1]
	Regularization term L1 on weights (reg_alpha)	Continuous	[0.001,**0.01**,0.1]
	Regularization term L2 on weights (reg_lambda)	Continuous	[0.001,**0.01**,0.1]
	Objective learning function (objective)	Categorical	['**reg:squarederror**','reg:linear']
	# of neurons in the ith hidden layer (hidden_layer_sizes)	Discrete	[(50,50,50), (50,100,50), **(100,)**]
	Activation function for hidden layer (activation)	Categorical	['**relu**','tanh','logistic','identity']
MLP	The solver for weight optimization (solver)	Categorical	['sgd','adam','**lbfgs**']
	Strength of the regularization term L2 (alpha)	Continuous	[0.001, **0.01**, 0.1]
	Learning rate for weight update program (learning_rate)	Categorical	['constant','**invscaling**','adaptive']
	# of epochs to train the model (epochs)	Discrete	[20,**90**,100]
	# of hidden layer	Discrete	[1,**2**,3,4]
	The function that tries to optimize (optimizer)	Categorical	['**adam**','rmsprop','sgd']
LSTM	Learning rate (learning_rate)	Continuous	[0.1, **0.01**, 0.001]
	Activation functions (activation)	Categorical	['tanh','sigmoid','**relu**']
	# of hidden layer neurons (neurons_1, neurons_2)	Discrete	[150, **250**, 350], [200, **200**, 300],
	Loss functions to be minimized during model training (loss)	Categorical	['mae','**mse**','mape']

Roadmap

Formulas for model performance measures.

We report and compare the performance of the SVR, XGBoost, MLP, and LSTM models for forecasting the MSI 20 stock market. Various evaluation measures are obtained to assess the models' accuracy, such as ME, MPE, MSE, MAE, RMSE, MAPE, and R^2 scores.

Metrics	**ME**	**MAE**	**MSE**	**MPE (%)**	**MAPE (%)**	**RMSE**	**R^2 (%)**
Formulas	$\frac{1}{n}\sum_{t=1}^{n}\varepsilon_{it}$	$\frac{1}{n}\sum_{t=1}^{n}\lvert\varepsilon_{it}\rvert$	$\frac{1}{n}\sum_{t=1}^{n}\varepsilon_{it}^2$	$\frac{100}{n}\sum_{t=1}^{n}\frac{\varepsilon_{it}}{y_t}$	$\frac{100}{n}\sum_{t=1}^{n}\left\lvert\frac{\varepsilon_{it}}{y_t}\right\rvert$	$\sqrt{\text{MSE}}$	$1-\frac{{}_{n}\text{MSE}}{\sum_{t=1}^{n}(y_t-\overline{y}_t)}$

Let $\hat{y}_{it}$ be the forecast of model i in time t, y_t is the real value in time t, $\overline{y}_t$ is the mean value, and n is the length of the set time series (i.e., training & test sets). The error for model i at time t is defined as $\varepsilon_{it} = y_t - \hat{y}_{it}$.

Forecast of MSI 20 stock price

Figure 1: MSI 20 stock price prediction: results of LSTM, SVR, MLP, and XGBoost models.

The performance analysis metrics of each model

Models	SVR		XGBoost		MLP		LSTM	
Error measure	Training	Test	Training	Test	Training	Test	Training	Test
ME	0.0899	0.674	2.277	0.558	5.406e-6	0.883	2.339	1.4
MAE	2.058	3.092	2.460	9.165	2.059	3.101	3.554	5.065
RMSE	2.646	3.993	3.141	13.515	2.642	4.018	4.496	6.322
MPE (%)	9.134e-3	0.082	0.226	0.116	7.603e-4	0.107	0.164	0.234
MAPE (%)	0.207	0.368	0.244	1.095	0.207	0.370	0.357	0.603
MSE	7.003	15.941	9.864	182.651	6.978	16.286	20.215	39.97
$\mathbf{R}^2$	0.998	0.989	0.997	0.882	0.998	0.989	0.995	0.974

Comparison and discussion

Based on the results presented in the table above and Figure 1, it can be concluded that the SVR model followed by MLP models had the best performance for forecasting MSI 20 compared to other models.

Roadmap

Conclusions, Limitations

Conclusion

The results showed that SVR outperformed the other models with lower errors and higher accuracy 98.9%.

Limitations

- The XGBoost model are not robust.
- The time-consuming GS algorithm.

Future Research

- Focus on improving the performance of XGBoost model.
- Exploring the potential of other models, such as CNN-LSTM, for stock market forecasting.

Roadmap

References I

[1] Arsalan Dezhkam and Mohammad Taghi Manzuri. "Forecasting stock market for an efficient portfolio by combining XGBoost and Hilbert–Huang transform". In: *Engineering Applications of Artificial Intelligence* 118 (2023), p. 105626.

[2] Hamid Ettayyebi and Khalid El Himdi. "Artificial neural network for forecasting one day ahead of global solar irradiance". In: *Smart Application and Data Analysis for Smart Cities (SADASC'18)* (2018).

[3] Gaurav Goverdhan, Sarthak Khare, and R Manoov. "Time Series Prediction: Comparative Study of ML Models in the Stock Market". In: (2022).

[4] Taewook Kim and Ha Young Kim. "Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data". In: *PloS one* 14.2 (2019), e0212320.

[5] Paraskevas Koukaras, Christina Nousi, and Christos Tjortjis. "Stock Market Prediction Using Microblogging Sentiment Analysis and Machine Learning". In: *Telecom*. Vol. 3. 2. MDPI. 2022, pp. 358–378.

References II

[6] Pratima Kumari and Durga Toshniwal. "Extreme gradient boosting and deep neural network based ensemble learning approach to forecast hourly solar irradiance". In: *Journal of Cleaner Production* 279 (2021), p. 123285.

[7] Sai Krishna Lakshminarayanan and John P McCrae. "A Comparative Study of SVM and LSTM Deep Learning Algorithms for Stock Market Prediction.". In: *AICS*. 2019, pp. 446–457.

[8] Ranran Li, Teng Han, and Xiao Song. "Stock price index forecasting using a multiscale modelling strategy based on frequency components analysis and intelligent optimization". In: *Applied Soft Computing* 124 (2022), p. 109089.

[9] Tian Liwei et al. "Forecast of LSTM-XGBoost in Stock Price Based on Bayesian Optimization.". In: *Intelligent Automation & Soft Computing* 29.3 (2021).

[10] Wenjie Lu et al. "A CNN-LSTM-based model to forecast stock prices". In: *Complexity* 2020 (2020), pp. 1–10.

References III

[11] Latrisha N Mintarya et al. "Machine learning approaches in stock market prediction: A systematic literature review". In: *Procedia Computer Science* 216 (2023), pp. 96–102.

[12] Abdullah H Al-Nefaie and Theyazn HH Aldhyani. "Predicting Close Price in Emerging Saudi Stock Exchange: Time Series Models". In: *Electronics* 11.21 (2022), p. 3443.

[13] Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer. "Deep learning for financial applications: A survey". In: *Applied Soft Computing* 93 (2020), p. 106384.

[14] Vesna Ranković et al. "Development of support vector regression identification model for prediction of dam structural behaviour". In: *Structural Safety* 48 (2014), pp. 33–39.

[15] Lihki Rubio and Keyla Alba. "Forecasting Selected Colombian Shares Using a Hybrid ARIMA-SVR Model". In: *Mathematics* 10.13 (2022), p. 2181.

[16] Niam Abdulmunim Al-Thanoon, Zakariya Yahya Algamal, and Omar Saber Qasim. "Hyper parameters Optimization of Support Vector Regression based on a Chaotic Pigeon-Inspired Optimization Algorithm". In: *Mathematical Statistician and Engineering Applications* 71.4 (2022), pp. 4997–5008.

References IV

[17] Jimmy Ming-Tai Wu et al. "A graph-based CNN-LSTM stock price prediction algorithm with leading indicators". In: *Multimedia Systems* (Feb. 2021). DOI: 10.1007/s00530-021-00758-w.

[18] Yufei Xia et al. "A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring". In: *Expert systems with applications* 78 (2017), pp. 225–241.

[19] Shuojiang Xu et al. "Comparison of Different Approaches of Machine Learning Methods with Conventional Approaches on Container Throughput Forecasting". In: *Applied Sciences* 12.19 (2022), p. 9730.

[20] Li Yang and Abdallah Shami. "On hyperparameter optimization of machine learning algorithms: Theory and practice". In: *Neurocomputing* 415 (2020), pp. 295–316.