Article

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

Hassan OUKHOUYA ‡*, **Khalid EL HIMDI** ‡

Laboratory LMSA, Department of Mathematics, Faculty of Sciences, Mohammed V University in Rabat, Rabat 10000, Morocco; k.elhimdi@um5r.ac.ma
* Correspondence: hassan.oukhouya@um5r.ac.ma; Tel.: +212-643-804-344 (F.L.)
‡ The 1st International Online Conference on Mathematics and Applications, 01 - 15 May 2023; Available online: https://iocma2023.sciforum.net

Abstract: Forecasting and modeling time series is a crucial aspect of economic research for academics and business practitioners. The ability to predict the direction of stock prices is vital for creating an investment plan or determining the optimal time to make a trade. However, market movements can be complex to predict, non-linear and chaotic, making it difficult to forecast their evolution. In this paper, we investigate modeling and forecasting the daily prices of the new Morocco Stock Index 20 (MSI 20). To this aim, we propose a comparative study between the results obtained from the application of the various Machine Learning (ML) methods: Support Vector Regression (SVR), eXtreme Gradient Boosting (XGBoost), Multilayer Perceptron (MLP), and Long Short-Term Memory (LSTM) models. The results show that using the Grid Search (GS) optimization algorithm, the SVR and MLP models outperform the other models and achieve high accuracy in forecasting daily prices.

Keywords: Time series; Modeling; Forecasting; MSI 20; Stock price; SVR; XGBoost; LSTM; MLP, GS.

Citation: OUKHOUYA, H.; EL HIMDI, K. Title. *Journal Not Specified* **2021**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

1. Introduction

The 2008 recession, the stock market crash of 2015, the Covid 19 Pandemic, and the Russian Invasion of Ukraine are some of the most recent crises with an immense impact on the financial markets and the destruction of wealth worldwide. Modeling and forecasting the stock market is a challenge that many engineers and financial researchers face. 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 [1,2]. Traditional econometric methods might require improved performance in relevant nonlinear time series and may not be appropriate for directly forecasting stock prices because of their volatility [3]. However, for complex nonlinear financial time series, methods such as Support Vector Regression (SVR), eXtreme Gradient Boosting (XGBoost), Multilayer Perceptron (MLP), and Long Short-Term Memory (LSTM) can detect nonlinear relationships in the forecasting stock prices [4], and achieve better fitting results by tuning multiple parameters [5]. Hyperparameter optimization or tuning in ML refers to selecting the most appropriate parameters for a particular learning model [6]. Some studies use Grid Search (GS) optimization [7], while others use Bayesian [8,9] or pigeon-inspired optimization algorithms [10]. In this paper, the GS algorithm is used to optimize the parameters of each model, such as SVR, XGBoost, MLP, and LSTM models. Then we compare them using seven measures Mean Error (ME), Mean Percentage Error (MPE), Mean Square Error (MSE), Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and R^2. In [11] discovered that using LSTM with Moving Averages (MA) yields superior results for predicting stock prices compared to SVR, as measured by different performance criteria such as MAE, MSE, MAPE, RMSE, and R^2 values. In their research [12], Al-Nefaie et al. proposed using LSTM and MLP models to forecast the fluctuations of the Saudi stock market. They found that the correlation coefficient for the two models was higher than 0.995. The LSTM model proved to have the highest accuracy and best model fit. In [13], the authors compared performance measures such as MSE, MAE, RMSE, and MAPE to forecast stock market trends based on Auto-Regressive Integrated Moving Average (ARIMA), XGBoost, and LSTM. Their tests found that XGBoost performed the best. In a different context, ML methods like ANN and MLP [14] were used to forecast solar irradiance. The results indicate that the MLP model with exogenous variables performs better than the other models. Similarly, in a study [15], the predictive performance and stability of eXtreme Gradient Boosting Deep Neural Networks (XGBF-DNN) made it an optimal and reliable model for forecasting hourly global horizontal irradiance using the GS algorithm. The main objective of this investigation is to improve the GS optimization algorithm by optimizing the hyperparameters of ML models. Furthermore, this study compares two ML methods and two DL methods. To achieve this objective, the research includes an in-depth literature review of 20 studies on ML models for stock market prediction. Overall, the paper aims to contribute to the stock market prediction field by improving the performance of ML models through hyperparameter optimization. Our contribution is highlighted in Table 1, a more accurate prediction than other studies.

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

Reference	Location	Year	Benchmark Model	MAPE (%)	RMSE	R^2
[11]	Irlande	2019	LSTM, SVR, MA	1.03	347.46	0.83
[16]	China	2020	MLP, CNN, RNN, LSTM, CNN-RNN	—	39.688	0.965
[13]	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**

The abbreviations have the following meaning: RNN = Recurrent Neural Network, CNN = Convolutional Neural Network.

The rest of this paper is organized as follows: Section 2 outlines the suggested methodology for modeling and forecasting financial time series, specifically stock prices. Section 3 details the implementation of the study, results, and discussion. Lastly, Section 4 offers conclusions and suggests future research directions.

2. Materials and Methods

The Materials and Methods section provides an overview of the methodology employed in this research.

2.1. Data Collection

This research focuses on modeling and forecasting a new Moroccan Stock Index 20 (MSI 20) is composed of the most liquid companies listed on the Casablanca Stock Exchange (CSE) from various sectors, including Attijariwafa Bank, Itissalat Al-Maghrib, Banque Populaire, LafargeHolcim Morocco,...etc. The MSI 20 is calculated in real-time during the business hours of the CSE from Monday to Friday, and from 9:00 a.m. to 3:30 p.m. local time (Limited on weekends and holidays). The research was conducted using the Python software. In addition, several used libraries, including Matplotlib, Pandas, NumPy, Sklearn, Tensorflow, and Keras. We use daily MSI 20 data to train each model and predict closing prices. Since the launch of the index of length $N = 541$, we use prices as Input = (Open, High, Low, and Closing prices), see the Table 2 below:

Table 2. Sample data (First five days) and descriptive statistics of MSI 20 index daily prices from 18 December 2020 to 09 February 2023.

Trade date	Open (MAD)	High (MAD)	Low (MAD)	Close (MAD)	Statistics	Open	High	Low	Close
2020-12-18	946.25	950.56	943.85	944.58	**Count**	541	541	541	541
2020-12-21	944.58	944.58	911.51	912.17	**Mean**	988.86	993.28	984.91	988.84
2020-12-22	912.17	928.51	906.77	927.79	**Std**	78.01	77.05	78.35	78.15
2020-12-23	927.79	935.71	927.07	932.43	**Min**	775.38	797.01	775.38	775.38
2020-12-24	932.43	932.81	926.58	927.16	**Max**	1140.69	1142.56	1135.86	1140.69

Note: MAD = Moroccan Dirham.

To train model parameters, we use historical data of length N,

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

where $X^{(i)}$ with $i = 1, 2, 3$ represents the Open, High, and Low prices, respectively. For the output with a single observation sequence, we use only closing price,

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

2.2. Preprocessing

The data set was divided into 90% of the observations used for training and 10% for model evaluation during testing. Data preprocessing is an essential step in ML that helps achieve competitive results and eliminate metric unit effects. In this case, we normalized using a min-max scale, which scales all variables to a range of $[0, 1]$:

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

where X_t is the historical data for each feature variable in the time series (Open, High, and Low prices), the X_t^{max} and X_t^{min} values are the sample's maximum and minimum values.

2.3. SVR Model

The SVR model, a new financial time series prediction method, is used to address the challenges of nonlinear regression. We assume a linear relationship exists between X_t and Y_t as in the left side of the equation 2. To perform nonlinear regression using SVR, the concept consists of creating an $x \longrightarrow \varphi(x)$ transformation that maps the original feature space X, which has N dimensions, onto the new feature space X'. Mathematically, this can be explained by the equation shown below:

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

where w_i is the vector of weights, and b_i is a bias, β_k is coefficient of the Lagrange multipliers, $\mathcal{K}(x_k, x) = \varphi(x_k) \cdot \varphi(x)$ is a kernel function. The most commonly used kernels include linear, gaussian, and polynomial functions [17].

2.4. XGBoost Model

XGBoost is an ML model used for stock market time series forecasting that uses a set of decision trees [18]. A gradient descent algorithm guides the process of preparing subsequent trees to minimize the loss function of the last tree [6].

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

where $\chi(\cdot)$ is a loss function specified that quantifies the deviations of the predicted and actual target values, $F_T(x_i) = \sum_{t=1}^{T} f_t(x_i)$ denotes the forecast on the i-th sample at the T-th boost and $\Pi(f) = \alpha K + 0.5 \times \kappa \|\varpi\|^2$, where K is the number of leaves. For the regularization term, α is the parameter of complexity. $\|\varpi\|^2$ is the L2 norm of weight regularization, κ is a constant coefficient, and $\Pi(\cdot)$ represents the term of regularization, which penalizes the model complexity [8].

2.5. MLP Model

The MLP is a frequently used ANN consisting of three layers of neurons: an input layer, one or more hidden layers, and an output layer. 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) [19]. 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].

2.6. LSTM Model

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 [20]:

$$\begin{aligned}
&\text{Forget gate:} && f_t = \delta(W_{ef} \cdot (X_t, h_{t-1}) + W_{cf} C_{t-1} + b_f), && \\
&\text{Input gate:} && i_t = \delta(W_{ki} \cdot (X_t, h_{t-1}, C_{t-1}) + b_i), && k = x, h, c \\
&\text{Memory cell:} && C_t = f_t \cdot C_{t-1} + i_t \cdot \tanh(W_{lc}(X_t, h_{t-1}) + b_c), && l = x, h \\
&\text{Output gate:} && o_t = \delta(W_{ji} \cdot (X_t, h_{t-1}, C_{t-1}) + b_o), && j = x, h, c
\end{aligned}$$

where W_{e*}, W_{c*}, W_{k*}, W_{l*}, W_{j*}, and b_f, b_i, b_c, b_o are the weight and bias of each layer respectively, and $h_t = o_t \cdot \tanh(C_t)$ represents the hidden layer output. The forget gate (f_t) determines if the information should be kept or discarded, while the input gate (i_t) integrates new data into the cell (C_t). Eventually, the output gate (o_t) governs the selection of relevant data to transmit to the succeeding cell (C_{t-1}).

2.7. Grid Search

In ML, GS is commonly used to fine-tune parameters such as regularization strength, learning rate, several hidden layers...etc. By using the GS algorithm, we can identify the optimal set of hyperparameters for the models, resulting in better predictions and improved performance. In our case, we first defined the hyperparameters and their search space, as shown in the Table 3. The optimal hyperparameters for each model are shown below in bold blue. The design of the research study is illustrated in Figure 1.

Table 3. The GS hyper-parameter sets for the SVR, XGBoost, MLP, and LSTM models.

ML Models	Parameters	Type	Search Space
SVR	Epsilon in the SVR loss function (`epsilon`)	Continuous	[0.0001,**0.001**,0.01]
	Regularization parameter (`C`)	Discrete	[100,**1000**,1100]
	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
XGBoost	# 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]
	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`']
MLP	# 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`']
	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`']
LSTM	# 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`']
	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`']

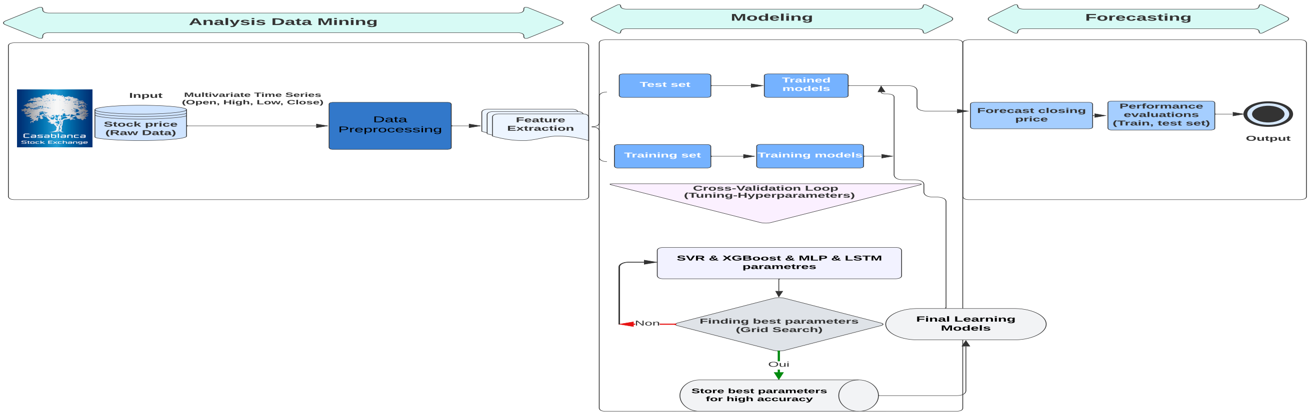

Figure 1. The proposed research design architecture for analyzing, modeling, and predicting MSI 20.

3. Results and Discussion

In the section on results and discussion, we report and compare the performance of the SVR, XGBoost, MLP, and LSTM models for forecasting the MSI 20 stock market. Various evaluation measures such as ME, MPE, MSE, MAE, RMSE, MAPE, and R^2 scores are obtained to assess the models' accuracy. 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 $\varepsilon_{it} = y_t - \hat{y}_{it}$. The general formula for evaluation measures is presented in Table 4 below:

Table 4. Formulas for model performance measures.

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

Table 5 shows the evaluation values of various forecasting models for the MSI 20 price. XGBoost had the highest error among the four models. LSTM had the second-worst performance. SVR and MLP models had lower performance metrics than XGBoost and LSTM models. However, the SVR model had the best evaluation values for forecasting MSI 20 prices, achieving optimal ME, MAE, RMSE, MPE, MAPE, and MSE values of 0.674, 3.092, 3.993, 0.082%, 0.368%, and 15.941, respectively. The MLP model also showed excellent performance metrics. However, the predicted results of the XGBoost model in Figure 2 on the right showed a specific error between actual. They predicted values of MSI 20, especially during the fall period (06/01/2023 - 10/01/2023), indicating that the model needed more data to improve its performance. Based on the results presented in Table 5 and Figure 2, it can be concluded that the SVR model followed by MLP models had the best performance for forecasting MSI 20 compared to other models.

4. Conclusions and future work

This study applied four machine learning models, including SVR, XGBoost, MLP, and LSTM, to model and forecast MSI 20 prices using multivariate time series data. The results showed that SVR outperformed the other models with lower errors and higher accuracy 98.9%. Future research could focus on improving the performance of XGBoost and exploring the potential of other models, such as CNN-LSTM, for stock market forecasting.

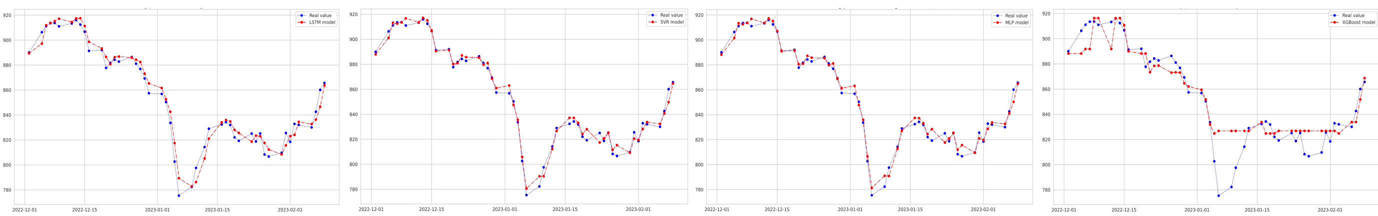

Figure 2. MSI 20 stock price prediction: results of LSTM, SVR, MLP, and XGBoost models.

Table 5. The performance analysis measures 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
R^2	0.998	0.989	0.997	0.882	0.998	0.989	0.995	0.974

Funding: This research received no external funding.

Data Availability Statement: The data is available on the CSE website: https://www.casablanca-bourse.com.

Acknowledgments: We appreciate all the reviewers who helped us improve the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mintarya, L.N.; Halim, J.N.; Angie, C.; Achmad, S.; Kurniawan, A. Machine learning approaches in stock market prediction: A systematic literature review. *Procedia Computer Science* **2023**, *216*, 96–102.
2. Ozbayoglu, A.M.; Gudelek, M.U.; Sezer, O.B. Deep learning for financial applications: A survey. *Applied Soft Computing* **2020**, *93*, 106384.
3. Li, R.; Han, T.; Song, X. Stock price index forecasting using a multiscale modeling strategy based on frequency components analysis and intelligent optimization. *Applied Soft Computing* **2022**, *124*, 109089.
4. Kim, T.; Kim, H.Y. Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data. *PloS one* **2019**, *14*, e0212320.
5. Xu, S.; Zou, S.; Huang, J.; Yang, W.; Zeng, F. Comparison of Different Approaches of Machine Learning Methods with Conventional Approaches on Container Throughput Forecasting. *Applied Sciences* **2022**, *12*, 9730.
6. Yang, L.; Shami, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing* **2020**, *415*, 295–316.
7. Rubio, L.; Alba, K. Forecasting Selected Colombian Shares Using a Hybrid ARIMA-SVR Model. *Mathematics* **2022**, *10*, 2181.
8. Xia, Y.; Liu, C.; Li, Y.; Liu, N. A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. *Expert systems with applications* **2017**, *78*, 225–241.
9. Liwei, T.; Li, F.; Yu, S.; Yuankai, G. Forecast of LSTM-XGBoost in Stock Price Based on Bayesian Optimization. *Intelligent Automation & Soft Computing* **2021**, *29*.
10. Al-Thanoon, N.A.; Algamal, Z.Y.; Qasim, O.S. Hyper parameters Optimization of Support Vector Regression based on a Chaotic Pigeon-Inspired Optimization Algorithm. *Mathematical Statistician and Engineering Applications* **2022**, *71*, 4997–5008.
11. Lakshminarayanan, S.K.; McCrae, J.P. A Comparative Study of SVM and LSTM Deep Learning Algorithms for Stock Market Prediction. AICS, 2019, pp. 446–457.
12. Al-Nefaie, A.H.; Aldhyani, T.H. Predicting Close Price in Emerging Saudi Stock Exchange: Time Series Models. *Electronics* **2022**, *11*, 3443.
13. Goverdhan, G.; Khare, S.; Manoov, R. Time Series Prediction: Comparative Study of ML Models in the Stock Market **2022**.
14. Ettayyebi, H.; El Himdi, K. Artificial neural network for forecasting one day ahead of global solar irradiance. *Smart Application and Data Analysis for Smart Cities (SADASC'18)* **2018**.
15. Kumari, P.; Toshniwal, D. Extreme gradient boosting and deep neural network based ensemble learning approach to forecasting hourly solar irradiance. *Journal of Cleaner Production* **2021**, *279*, 123285.
16. Lu, W.; Li, J.; Li, Y.; Sun, A.; Wang, J. A CNN-LSTM-based model to forecast stock prices. *Complexity* **2020**, *2020*, 1–10.
17. Ranković, V.; Grujović, N.; Divac, D.; Milivojević, N. Development of support vector regression identification model for prediction of dam structural behavior. *Structural Safety* **2014**, *48*, 33–39.
18. Dezhkam, A.; Manzuri, M.T. Forecasting stock market for an efficient portfolio by combining XGBoost and Hilbert–Huang transform. *Engineering Applications of Artificial Intelligence* **2023**, *118*, 105626.
19. Koukaras, P.; Nousi, C.; Tjortjis, C. Stock Market Prediction Using Microblogging Sentiment Analysis and Machine Learning. Telecom. MDPI, 2022, Vol. 3, pp. 358–378.
20. Wu, J.M.T.; Li, Z.; Herencsar, N.; Vo, B.; Lin, C.W. A graph-based CNN-LSTM stock price prediction algorithm with leading indicators. *Multimedia Systems* **2021**. doi:10.1007/s00530-021-00758-w.