International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 3, June 2023, pp. 3420~3431
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i3.pp3420-3431

A novel hybrid deep learning model for price prediction

Walid Abdullah[1], Ahmad Salah[1,2]
[1]Department of Computer Science, College of Computers and Informatics, Zagazig University, Zagazig, Egypt
[2]Information Technology Department, College of Computing and Information Sciences, University of Technology and Applied Sciences, Ibri, Sultanate of Oman

Article Info

Article history:

Received Jul 27, 2022
Revised Sep 19, 2022
Accepted Oct 1, 2022

Keywords:

Deep leaning
Machine learning
Price prediction
Statistical models
Time series analysis

ABSTRACT

Price prediction has become a major task due to the explosive increase in the number of investors. The price prediction task has various types such as shares, stocks, foreign exchange instruments, and cryptocurrency. The literature includes several models for price prediction that can be classified based on the utilized methods into three main classes, namely, deep learning, machine learning, and statistical. In this context, we proposed several models' architectures for price prediction. Among them, we proposed a hybrid one that incorporates long short-term memory (LSTM) and Convolution neural network (CNN) architectures, we called it CNN-LSTM. The proposed CNN-LSTM model makes use of the characteristics of the convolution layers for extracting useful features embedded in the time series data and the ability of LSTM architecture to learn long-term dependencies. The proposed architectures are thoroughly evaluated and compared against state-of-the-art methods on three different types of financial product datasets for stocks, foreign exchange instruments, and cryptocurrency. The obtained results show that the proposed CNN-LSTM has the best performance on average for the utilized evaluation metrics. Moreover, the proposed deep learning models were dominant in comparison to the state-of-the-art methods, machine learning models, and statistical models.

Corresponding Author:

Ahmad Salah
Department of Computer Science, College of Computers and Informatics, Zagazig University
El-Zeraa Ssquare, Zagazig, 44519, Egypt
Email: ahmad@zu.edu.eg

1. INTRODUCTION

In recent years, a large number of investors turned to investing in financial instruments such as the stock market, currency, and crypto. One of the most important problems facing them is market fluctuations, which makes it difficult to predict market prices. On the other hand, artificial intelligence (AI) technologies such as machine learning and deep learning (DL) have advanced substantially and rapidly. These technologies have contributed to eliminating many of the difficulties in the different tasks including time series analysis (TSA) and forecasting, which helps us to predict the future values of a data series using its historical values. TSA is a crucial area for research in several fields [1]–[5]. In the financial field, TSA can be used for forecasting instrument prices to help investors and researchers to understand and beat market fluctuations. The accurate forecasting of instrument prices can help investors to minimize risks and obtain higher benefits [6], [7].

Financial time series data forecasting has been a key research field for many years. However, price forecasting is a difficult task and is generally regarded as one of the most challenging problems in time-series forecasting. This is because the price change depends on many factors, and the financial data contains a lot of noise and complexity [8], [9]. For accurate price prediction, the researchers proposed several forecasting models that can be classified based on the utilized methods into four classes, namely, statistical models, DL models, and machine learning models.

The auto-regressive integrated moving average (ARIMA) model is one of the most popular statistical models used in time series analysis tasks [10], [11]. ARIMA has the ability to deal with nonstationary data series which makes it suitable for use in most price forecasting problems. Rangel-González *et al*. [12] applied six classic statistical models, namely, auto-regressive (AR), moving average (MA), auto-regressive integrated (ARI), integrated moving average (IMA), auto regressive moving average (ARMA), and ARIMA models for predicting the financial time series data of the Mexican Stock Exchange. According to the obtained results, the ARIMA model achieved the best results among all these classic models. However, it is critical to evaluate the ARIMA model's properties to get the best performance. ARIMA model was proposed in [13] s well, to forecast the Indian stock market volatility to protect investors' interests. This analysis relied on publicly available time-series data from the Indian stock market and used the Nifty and Sensex indicators to check the model. A comparison of the predicted and actual time series was made, which results in a mean percentage error of about 5% for both the Nifty and the Sensex indicators. For validation purposes, the augmented Dickey–Fuller (ADF) and the L-Jung box tests have been employed in this work.

With the emergence of new AI techniques such as machine learning (ML), researchers in the field of price prediction turned to take advantage of the ability of its predictive algorithms such as linear regression (LR), random forest (RF), and support vector regression (SVR). The study [14] proposed a new approach by combining different kinds of windowing functions with an SVR. The method utilized an SVR model for predicting stock market prices and trends with different kinds of windowing functions as data preprocessing steps to feed the input into the ML algorithm for the sake of pattern recognition. The utilized dataset is collected from the Dhaka Stock Market (DSE). According to the result, the authors found that the SVR model with flattened and rectangular windows is good to predict the stock price 1, 5, and 22 days ahead because the mean absolute percentage error (MAPE) is quite acceptable. In addition, SVM has been used in many other applications such as stock market forecasting [15], and cryptocurrency price perdition [16]. Other works utilized the RF and LR models for gold price prediction tasks [17], [18], and the reported results are acceptable.

DL is a new trend in ML. DL models have a deep nonlinear topology in their particular structure that gives them the ability to extract crucial information from time series data [19]–[21]. Sim *et al.* [22] proposed a convolutional neural network (CNN)-based stock price prediction model to test the applicability of novel learning approaches in stock markets. Technical indicators were transformed into images of the time-series graph. It was found from the results that the proposed CNN model outperformed the models of comparison on the prediction accuracy rates. Moreover, it has the ability to recognize the changes in trends well. Borovkova and Tsiamas [23] introduced a long-short term memory (LSTM) model for intraday stock forecasting (LSTM is a modified version of the recurrent neural network (RNN) algorithm that can learn and memorize long-term dependencies). The proposed LSTM model was designed with a wide range of technical analysis indicators as network inputs, and the model's performance was tested on many large-cap stocks in the US and compared to lasso and ridge logistic regression. The obtained results revealed that the proposed LSTM model performs better than the benchmark models or equally weighted ensembles. A more recent version of the LSTM architecture is known as bidirectional-LSTM (BI-LSTM), it is used for Forex forecasting [24]. The BI-LSTM model has two hidden layers, but in inverse directions. The results show that the BI-LSTM model outperformed the conventional LSTM in Forex forecasting.

Finally, for getting better results, researchers proposed hybrid models by the integration of two or more models in order to obtain a new model that takes advantage of useful features of each individual model, for example, a hybrid model called the auto-regressive fractional integrated moving average (ARFIMA)-LSTM [25]. This proposed model combined the ARFIMA model, which captures nonlinearity in the residual values, with an LSTM model. This combination helped the proposed hybrid model to overcome the overfitting problem of neural networks and to enhance the prediction accuracy of individual models independently. Another model that combines the ARIMA and LSTM was proposed in [4]; the obtained results admitted that the proposed hybrid model performs better than the other benchmark models taken into account in this study since it attained lower error values.

In literature, there are some studies that have been conducted to compare the accuracy of different models in forecasting financial statements such as conducted in [26], which compared the ARIMA, LSTM, and a set of ML models in price prediction of seasonal items. Makala and Li [27] compared the ARIMA and SVR models' accuracies to predict gold prices. Hua [28] proposed comparing the ARIMA and LSTM performance in bitcoin price prediction. However, it can be noticed that there is no previous research conducted to study the performance gap between statistical, ML, and DL models, and most of the previous studies were conducted with a specific type of problem. In instrument prediction, most of these researchers worked with a financial time series for only a specific type of instruments (i.e., stock market, foreign exchange, cryptocurrency, and gold price). To our knowledge, no work has been tested on different problems with different patterns of data series.

In this work, a novel hybrid DL model named CNN-LSTM was proposed. This model benefits from the advantage of convolution layers' ability to extract useful features embedded in the time series data and LSTM capability for learning order dependence in sequence prediction problems. In addition, a new LSTM model architecture was developed as a DL model. Finally, this work proposed studying the performance gap of the statistical, ML, and DL methods in financial TSA problems. In this context, the results of the ARIMA, LR, the proposed LSTM, and the proposed hybrid CNN-LSTM models are compared. All predictive models were tested with three different types of financial product datasets for three different dataset categories (stocks, foreign exchange instruments, and cryptocurrency). Moreover, the proposed models are compared against the state-of-the-art methods.

The remaining parts of the paper are as follows. Section 2 discusses the methods and approaches used in the paper and presents the proposed methodologies and shows the diagrams of the proposed models. The experimental results are elaborated on in section 3. Finally, section 4 summarizes the conclusion and the findings of our research.

2. RESEARCH METHOD

In this work, we compared a set of different predictive models utilizing three different techniques (i.e., statistical, ML, and DL), in financial time series analysis to predict instrument prices. The used models were trained and tested on three different dataset categories (e.g., currencies forecasting, stock market forecasting, and crypto price forecasting). The predictive models are the ARIMA model as a statistical model, LR as a ML model, and two proposed models namely, the LSTM model as a DL model, and the CNN- LSTM model as a hybrid model.

2.1. ARIMA model

The ARIMA model was developed by Box and Jenkins [10]. It is an improvement of the ARMA model (which combination of autoregressive and moving average models) by adding an "integrated" term that relates to how many times a series must be differenced before the data series becomes stationary. Thus, unlike the ARMA model, the ARIMA model can work with non-stationary data series because it can change it to be stationary using differencing it. It is called "ARIMA (p, d, q)" where the parameter p represents the order of the AR model, d is the degree of differencing, and q represents the order of the MA model.

As shown in Figure 1, the ARIMA model consists of four stages in the first stage, data is collected and prepared. Then in the second stage, Stationarity is checked using tests like the ADF test to check whether the series is stationary or not, The ADF test is a common statistical test for stationarity. If the value of p for a time series is less than 0.05 or the 5% level and the test statistic is greater than the critical values in the ADF test, the series is said to be stationary. If the series is non-stationary, it must be different to become stationary.

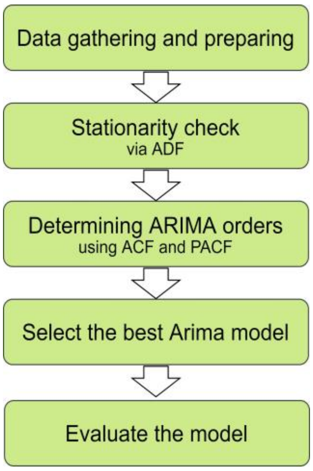


Figure 1. The steps of prediction with ARIMA model

ARIMA parameters (p, d, q) are determined in the third stage by plotting the auto-correlation function (ACF) and partial auto-correlation function (PACF). The parameters denote the relationship between the observations within a time series [29], [30]. ARIMA parameters aid in determining a range of possible values for the model parameters that can produce a higher predictive accuracy. Then, the best ARIMA order is selected by testing all combinations of ARIMA parameters' possible values and evaluating each combination to select the best one. Then, the model, with the best parameters combination, is tested on the selected datasets to forecast future values using accuracy metrics in the last stage.

2.2. Linear regression model

The LR model is one of the most well-known supervised ML models. It is a linear model that is used for finding out the relationship between the input variable x and the output variable y. It is known as simple LR when there is only one input variable x. When there are several input variables, it is called a multiple LR. It works according to (1),

$$Y = \beta_0 + \beta_1 X_1 \tag{1}$$

where β_0 and β_1 represent the coefficients of the linear equation of the LR model.

In this paper, an LR model is used for forecasting financial time series data, where the previous data values (X_1, X_2, …., X_n) are used as input values and fed to the model for forecasting the future value Y. First, the data series is prepared as supervised data (attributes and target). Various numbers of previous data values (lag values) are used as attributes to forecast the next time step value (i.e., target) to test them to choose the suitable number. For the datasets used in this work, multiple numbers of the previous time steps were tested to determine the best number of lag values that lead to the best accuracy. Then, the model was trained using the cross-validation method with five folds to overcome the overfitting problem.

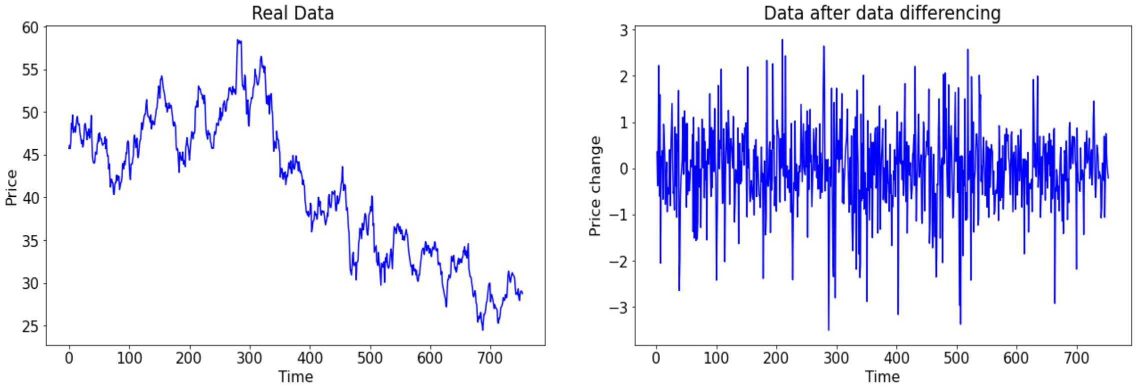
2.3. The proposed LSTM models

LSTM is a modified version of RNN [31]. RNNs have been effectively employed to forecast data series. It can remember previous observations by keeping track of a cell state that is updated at each time step. However, long-term dependencies cannot be simulated efficiently because the error signals from previous observations have to be less and less during training, this is known as the "vanishing gradient problem". Unlike RNN, LSTM was proposed in [31] to deal with the vanishing gradient problem by storing states in a cell state. In addition, the LSTM has a forget gate that determines if prior state information is important or not. If the forget gate output is 1, the information is saved, and if the output is 0, the information is ignored. This design helps LSTM's cells to store only the important information.

In this work, an LSTM model has been proposed as a DL model. The proposed model has been developed through four phases as followed. In the first phase, data is collected and processed. The model is proposed to work on different data values. Thus, instead of using the actual data, the model uses a differenced data value, which is a series of changes in data that was created using (2).

$$X_{(t)} = X_{(t)} - X_{(t-1)} \tag{2}$$

Figure 2 shows the effect of data differencing on the data series, the real data is shown in Figure 2(a) and the differenced data is shown in Figure 2(b). In the second phase, we proposed an LSTM architecture and hyperparameters tuning of the proposed model. The proposed LSTM model architecture is depicted in Figure 3.



(a) (b)

Figure 2. The effect of data differencing on the data series (a) the real data and (b) the data after differencing

As shown in Figure 3, the proposed LSTM model contains three LSTM layers, each layer contains 40 LSTM cells. A dropout layer was used after each LSTM layer to avoid overfitting [32]. A dropout layer is a regularization approach that allows training the neural networks with different architectures in parallel by randomly removing some of the output features of the layer during training, it is considered a powerful approach to prevent overfitting. The last layer is a fully connected layer, which is the output layer of the model and contains one neuron. The model uses the Adam optimizer to fit the data. Adam optimizer is an adaptive optimization technique that has been proven to be effective in tackling practical DL challenges [33], where the mean absolute error (MAE) function is used as the loss function. The third phase includes model training and validation. Then, the model prediction is ready to be conducted. In the last phase, the actual predicted data will be reversed from the differenced data to retrieve the original data values.

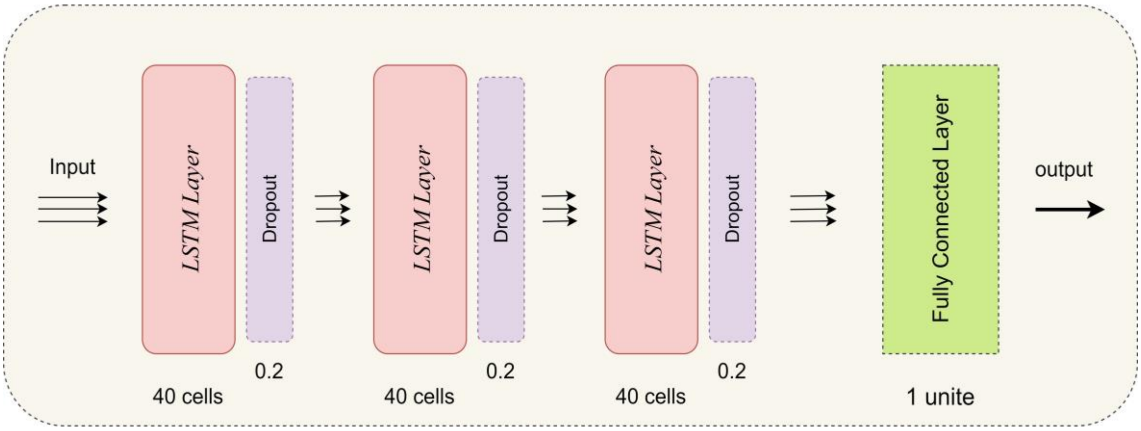


Figure 3. The proposed LSTM model architecture

2.4. The proposed hybrid CNN-LSTM model

A CNN is a type of neural network designed to work with two-dimensional image input [34]. It has the ability to extract and learn useful features from univariate time series data and other one-dimensional sequence data. Therefore, it is considered one of the most important models of DL and is used for many other purposes as in [35], [36]. The main goal of developing a hybrid model is to combine the CNN and LSTM layers to make use of the respective qualities of these layer types, which produces an effective model for forecasting instrument prices accurately. In this section, a hybrid model called CNN-LSTM model is proposed to make use of the characteristics of convolution layers like their ability to extract useful features embedded in the time series data, and CNN layers help in filtering out the noise of the input data. In addition, the proposed hybrid model benefits from the ability of LSTM layers for identifying short-term and long-term dependencies.

The proposed CNN-LSTM architecture is depicted in Figure 4. The first layer in the model is a 1D convolutional layer that reads through the subsequence, it requires several filters to extract features or interpretations of the input sequence. Then, a rectified linear unit (ReLU) activation function is used to improve the model's ability to learn complex structures, The ReLU activation function is resistant to gradient vanishing problems to improve the trainability of the network. After a convolutional layer, a pooling layer is frequently added to mitigate the invariance of the resulting feature map. In the proposed hybrid model, a max-pooling layer is placed after the convolution layer to reduce the feature maps by a constant factor, and as a result, it helps in highlighting the most important features. To prevent overfitting, a dropout layer is incorporated into the network, which comprises a random selection of neurons that are deactivated.

Finally, using a flatten layer, these layers are converted to a single 1D vector, which is used as a single input time step to the LSTM layers to evaluate the input sequence read by the CNN model and make the prediction. The LSTM part of the model consists of three LSTM layers, each layer with 100 units. Each LSTM layer is followed by a dropout layer. The output layer of the model is a dense layer with one unit. In this proposed hybrid model, the Adam optimizer with the mean absolute error loss function is utilized as the objective function. Table 1 lists the model's different layers configuration while Table 2 lists the hybrid model's hyperparameter settings.

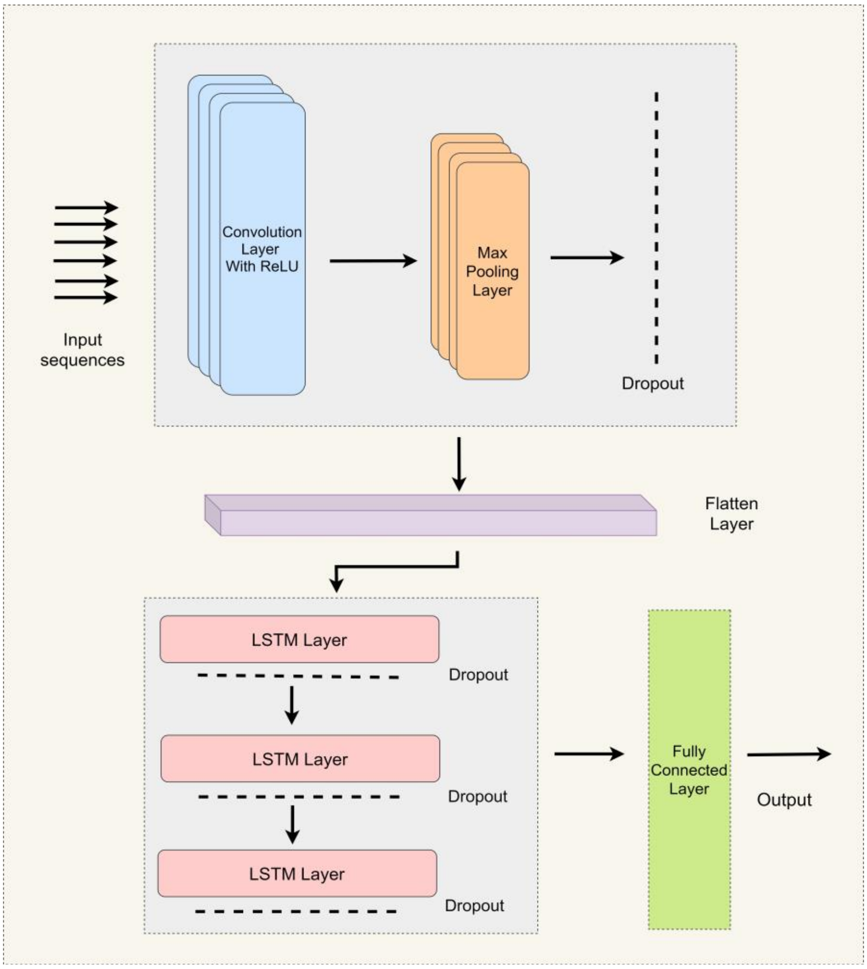


Figure 4. The proposed CNN-LSTM framework architecture

Table 1. The proposed CNN-LSTM model's different layers configuration

Layer	Parameters	Configuration
Conv1D	Kernel size	1
	No of Filters	64
	Activation	ReLU
MaxPooling	Pool size	2
Dropout1	-	0.2
Flatten	-	-
LSTM1	No of units	100
	Return sequence	True
Dropout2	-	0.2
LSTM2	No of units	100
	Return sequence	True
Dropout3	-	0.2
LSTM3	No of units	100
Dropout4	-	0.2
Fully connected	No of Units	1

Table 2. The proposed model's hyperparameter settings

Parameter	Setting
Optimizer	Adam
Initial Learning rate	0.0001
Decay rate	0.6
Lose function	MAE
Batch size	64
Epochs	100
Early Stopping	Monitor=loss, patience=10

3. RESULTS AND DISCUSSION

This section has been divided into four subsections that provide a detailed discussion of the case study and the final results. It presents the experimental setup and the utilized tools for the models' implementation and discusses the utilized datasets and evaluation metrics for testing and evaluating the models. Additionally, it presents the models' hyperparameters tuning process and its results. Finally, it displays the final results and outcomes of the work.

3.1. Experimental setup

The experiments were conducted on a PC with a 64-bit Windows 10 OS with an Intel 8-core processor running at 2.3 GHz and 8 GB RAM. The Python Scripting language version 3.8.3 was used to develop the predictive models. For implementing LSTM and CNN-LSTM models, Keras [37] with TensorFlow [38] was employed. Additionally, Scikit-learn [39] and Statsmodels libraries were used to implement the LR and ARIMA models, respectively.

3.2. Dataset

All the utilized datasets in the experiments were obtained from Yahoo Finance [40] for three years before the Coronavirus pandemic, from the 1st of December 2016 to the 1st of December 2020 using a daily time frame. As mentioned before, the experiments were performed on three different dataset categories, namely, stock market, foreign exchange instruments, and cryptocurrency. To forecast currency rates, three datasets have been utilized (i.e., EURUSD, USDTRY, and EURGBP). To predict stock market prices, three datasets have been used (i.e., AAPL, AAL, and CBMB). Finally, BTC-USD, THETA-USD, and VET-USD datasets have been used to predict cryptocurrencies' prices.

3.3. Evaluation metrics

To determine the correctness and accuracy of the predictive models, two assessment metrics are used, namely, mean absolute error (MAE) and mean squared error (MSE). MAE calculates the average difference between the original and predicted values. As a result, we can estimate how close the predictions are to the real data. MAE is represented mathematically as (3).

$$MAE = \frac{1}{N} \sum_{i=1}^{N} \mid p_i - \mathrm{a}_i \mid \tag{3}$$

where p_i stands for predicted values, a_i stands for the actual values, and N stands for the number of samples.

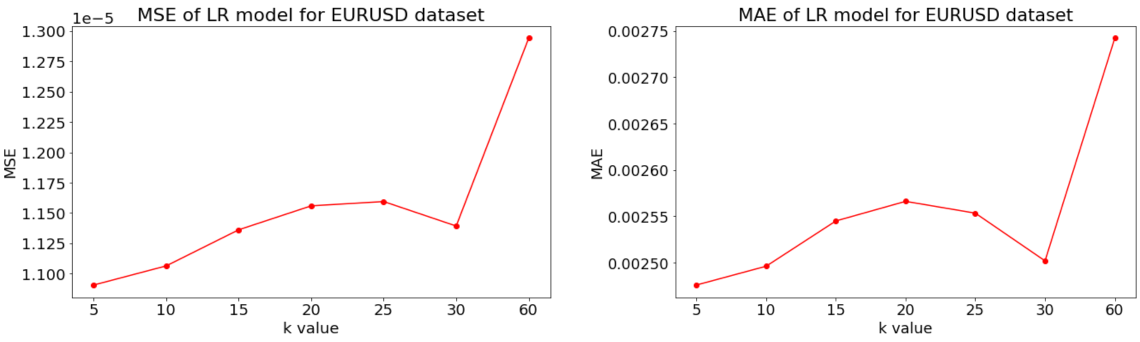
On the other hand, MSE measures the average of the squares of the errors. MSE represents the average squared difference between the real and the predicted values. It is used to evaluate the accuracy of regression problems. It can be mathematically represented as (4).

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (p_i - \mathrm{a}_i)^2 \tag{4}$$

3.4. Hyperparameters tuning

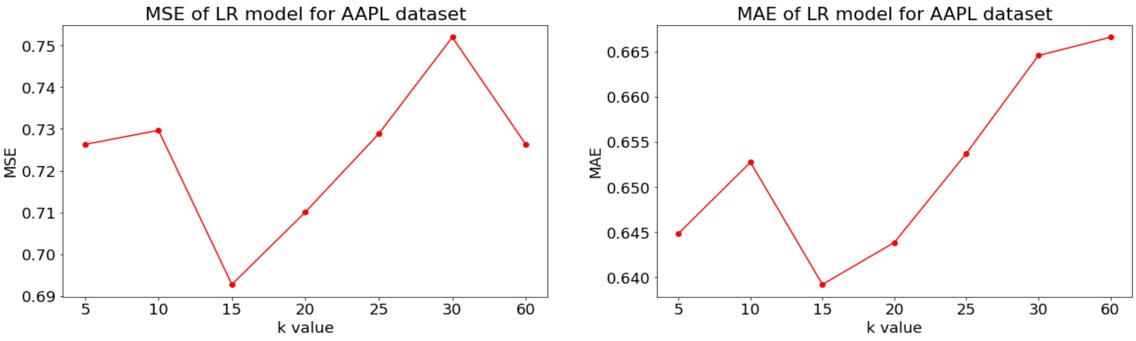
Hyperparameter tuning is the most important process. There is no specific way that tells us about the best combination of parameters' values for a model that results in improving the model's accuracy. In this paper, there are several models have been utilized; each model has a lot of parameters that are used in the models, which have been built and need to be optimized. We proposed performing a grid search and trial and error method to search for the best combinations, as used in the ARIMA model. ACF and PACF were used to determine a range of candidate values for ARIMA orders then test all combinations of these possible values and evaluate each model to select the best. Similarly, grid search was applied to the rest of the models.

ML and DL models' parameters are a time-consuming step because there are a huge number of hyperparameters to be tuned. One of the most critical parameters in any ML or DL model is the number of earlier observations which are known as lag features that are used in the training to help the model to predict future values. The used models need to be trained using a number of lag features (i.e., K), that is an important hyperparameter and it needs to be well-optimized. Multiple values have been tested to determine the best K value that gives the highest possible accuracy prediction rate in terms of MAE and MSE evaluation. Figures 5 to 7 depict the performance of the ML model (LR) with various values of K (i.e., K =5, 10, 15, 20, 30, and 60) for EURUSD, APPL, and BTC-USD datasets, respectively. Figures 5(a), 6(a), and 7(a) show the MSE and Figures 5(b), 6(b), and 7(b) show the MAE of the models. The results of one dataset of each dataset category are reported, as the rest of the datasets in the same category have similar behavior.



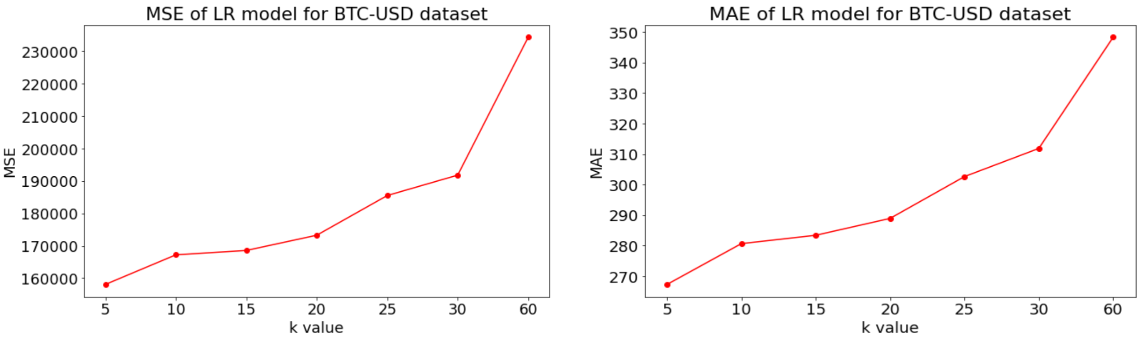
(a) (b)

Figure 5. LR model evaluation with different number of lag features for the EURUSD dataset (a) MSE and (b) MAE



(a) (b)

Figure 6. LR model evaluation with a different number of lag features for the APPL dataset (a) MSE and (b) MAE



(a) (b)

Figure 7. LR model evaluation different number of lag features for the BTC-USD dataset (a) MSE and (b) MAE

As depicted in Figures 5 to 7, the LR model performance is related to the value of K for all of the datasets. In other words, when the value of K increases, the performance of the LR model decreases. That is normal for ML models, since K is the number of attributes that are used to predict the target, and the higher number of attributes results in a higher prediction complexity. Thus, 5 and 10 are considered the best values for the K variable for the LR model; this is one shortcoming of this model because a high value of K helps to understand the data series better which gives better results. On the other hand, using high K values with LSTM and CNN-LSTM models does not represent any problem because of the presence of LSTM layers that can "remember" previous observations and have the ability to learn and identify long-term dependencies. Thus, unlike LR models, LSTM and CNN-LSTM models can take advantage of high values of K to better understand the data series to increase their performance. In this work, K with a value of 60 is used in the experiments for DL models.

3.5. Results

In this section, the performance of each implemented model was measured with all datasets used in this study. As mentioned above, the models need to be evaluated with different types of time series patterns. Therefore, the datasets have been selected from three different categories (i.e., currencies rate and the stock market and cryptocurrencies). Since in real life, each dataset category has a different pattern of time series and behaves specially along with the series, and different products in each dataset within the same category also have a different pattern, the prediction model must deal with all these variations of patterns. Then, each model was tested on all datasets in all categories. After that, the proposed methods are thoroughly evaluated and compared against state-of-the-art methods in [41], [42]. Tables 3 to 5 list the evaluation metrics for currencies and the stock market and cryptocurrencies prediction respectively for all used models.

Table 3. Various models' evaluation metrics for currencies forecasting

Model	EURUSD		USDTRY		EURGBP	
	MAE	MSE	MAE	MSE	MAE	MSE
ARIMA	0.002458	1.06E-05	0.033323	0.001972	0.002673	1.35E-05
LR	0.002476	1.09E-05	0.034532	0.002086	0.002707	1.39E-05
LSTM	0.002404	1.05E-05	0.032246	0.001945	0.002635	1.33E-05
CNN-LSTM	0.002471	1.06E-05	0.032244	0.001918	0.002618	1.31E-05
[41]	0.037755	0.001457	0.634695	0.406883	0.011370	0.000232
[42]	0.006885	7.77E-05	0.120082	0.021759	0.005311	4.77E-05

Table 4. Various models' evaluation metrics for stock market forecasting

Model	AAPL		AAL		CBMB	
	MAE	MSE	MAE	MSE	MAE	MSE
ARIMA	0.633166	0.707329	0.498855	0.459775	0.041411	0.004188
LR	0.644838	0.726291	0.505314	0.467957	0.039097	0.003713
LSTM	0.640490	0.718705	0.495852	0.458594	0.044154	0.004104
CNN-LSTM	0.637772	0.705883	0.493771	0.451721	0.041239	0.004003
[41]	2.067835	8.068114	4.944883	7.229334	0.174802	0.031923
[42]	1.885983	6.0204133	1.537432	3.154869	0.220959	0.062321

Table 5. Various models' evaluation metrics for cryptocurrencies forecasting

Model	BTC-USD		THETA-USD		VET-USD	
	MAE	MSE	MAE	MSE	MAE	MSE
ARIMA	233.4722	115548.639	0.003602	2.66E-05	0.0002214	9.95E-08
LR	267.2719	158070.390	0.003597	2.62E-05	0.0002225	9.96E-08
LSTM	264.7609	155688.131	0.003634	2.61E-05	0.0002175	9.70E-08
CNN-LSTM	232.7548	117411.911	0.003633	2.61E-05	0.0002173	9.85E-08
[41]	509.3929	886403.031	0.015145	0.000249	0.0009576	1.27E-06
[42]	364.3295	485759.658	0.005415	5.17E-05	0.0007218	8.51E-07

The visual illustration of the ARIMA, LR, LSTM, and CNN-LSTM models’ performance is depicted in Figures 8 to 11. Using the AAPL dataset as an example, the figures show the comparison of the predicted values and the real values for the first 30 days in testing the dataset for the proposed models. The models' efficiency can be determined by noting the difference between the real value and the predicted values, When the difference is smaller it means that the model performs better.

The discussion of the obtained results can be summarized as follows: i) the results show that all the developed models in this paper outperform the state-of-the-art models [41], [42] with all datasets; ii) the comparison between statistical, ML, and DL models reveal that the DL models like LSTM and CNN-LSTM have better performance for most of the data series patterns compared to ARIMA and linear regression models; iii) the results proved that the proposed CNN-LSTM model outperforms the LSTM model for most of the utilized datasets in this work, which proves that, merging the convolution layer’s ability to extract useful features embedded in the time series data with the ability of LSTM for identifying long-term dependencies feature led to higher prediction accuracy rates; iv) experiments showed that ML models like linear regression may deal well in some cases with TSA if the values of the hyperparameters are carefully set. As listed in the results, linear regression models produce a good prediction on some datasets such as CBMB and THETA-USD; and v) finally, it was observed that no predictive model outperforms all of the other models neither in all used datasets nor in all datasets within any category.

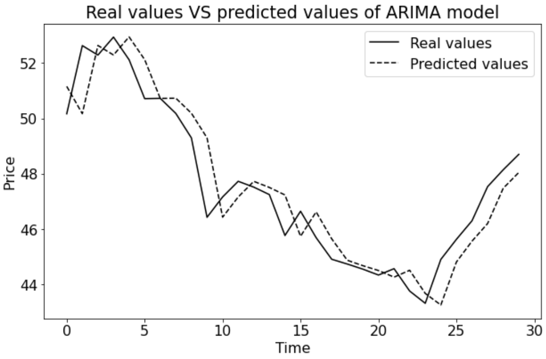


Figure 8. ARIMA predicted values against actual values for AAPL dataset

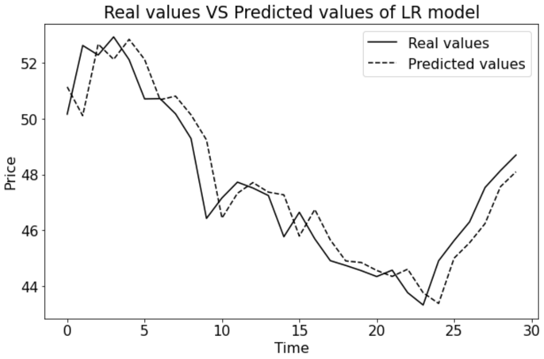


Figure 9. LR predicted values against actual values for AAPL dataset

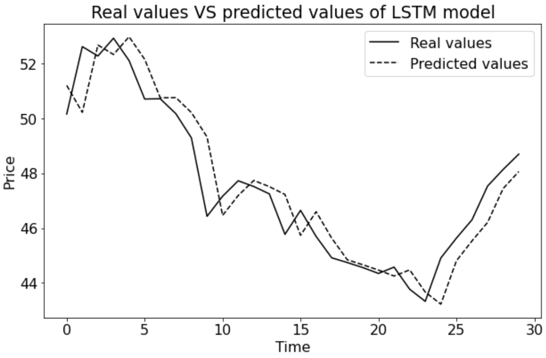


Figure 10. LSTM predicted values against actual values for AAPL dataset

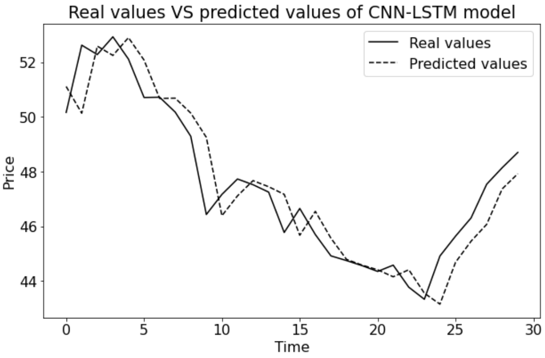


Figure 11. CNN-LSTM predicted values against actual values for AAPL dataset

4. CONCLUSION

In this work, we proposed a hybrid DL model named CNN-LSTM model for price prediction. The proposed model combines layers from two different architectures, i.e., CNN and LSTM. As different financial types have different price patterns, we proposed collecting three datasets for stocks, foreign exchange instruments, and cryptocurrency. Then, we proposed evaluating the performance variance of the proposed model as a response to different dataset patterns. In addition, the proposed hybrid model was compared against representative models from the major price prediction techniques, namely, statistical, ML, and DL models. The performance of the proposed CNN-LSTM model was compared against the state-of-the-art models using two different evaluation metrics, MSE and MAE. The obtained results showed that the proposed DL models are better than the state-of-the-art methods, ML models, and statistical models on average. Moreover, the results reported that the proposed CNN-LSTM model outperforms the LSTM model for most of the utilized datasets in this work, which prove that the merging of the convolution layer's ability to extract useful features embedded in the data and the ability of LSTM layers of identifying long-term dependencies feature led to higher prediction accuracy rates. Future directions include exploring the performance of a DL model that combines the architectures of CNN, LSTM, and Transformers. In addition, considering combining the financial sentiment analysis techniques with the proposed model will be explored in future works.

REFERENCES

[1] E. Yuniarti, S. Nurmaini, and B. Y. Suprapto, "Indonesian load prediction estimation using long short term memory," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 11, no. 3, pp. 1026–1032, Sep. 2022, doi: 10.11591/ijai.v11.i3.pp1026-1032.

[2] S. Bhanja and A. Das, “A hybrid deep learning model for air quality time series prediction,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 22, no. 3, pp. 1611–1618, Jun. 2021, doi: 10.11591/ijeecs.v22.i3.pp1611-1618.

[3] N. F. Aurna *et al.*, “Time series analysis of electric energy consumption using autoregressive integrated moving average model and holt winters model,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 3, pp. 991–1000, Jun. 2021, doi: 10.12928/telkomnika.v19i3.15303.

[4] O. Yakubu and N. B. C., “Electricity consumption forecasting using DFT decomposition based hybrid ARIMA-DLSTM model,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 24, no. 2, pp. 1107–1120, Nov. 2021, doi: 10.11591/ijeecs.v24.i2.pp1107-1120.

[5] H. AL-Khazraji, A. Nasser, and S. Khlil, “An intelligent demand forecasting model using a hybrid of metaheuristic optimization and deep learning algorithm for predicting concrete block production,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 11, no. 2, pp. 649–657, Jun. 2022, doi: 10.11591/ijai.v11.i2.pp649-657.

[6] E. F. Fama, “Market efficiency, long-term returns, and behavioral finance,” *Journal of Financial Economics*, vol. 49, no. 3, pp. 283–306, Sep. 1998, doi: 10.1016/S0304-405X(98)00026-9.

[7] K. Pawar, R. S. Jalem, and V. Tiwari, “Stock market price prediction using LSTM RNN,” in *Emerging Trends in Expert Applications and Security*, 2019, pp. 493–503. doi: 10.1007/978-981-13-2285-3_58.

[8] A. Cowles, “Can stock market forecasters forecast?,” *Econometrica*, vol. 1, no. 3, pp. 309–324, Jul. 1933, doi: 10.2307/1907042.

[9] R. A. Schwartz, “Efficient capital markets: A review of theory and empirical work: discussion,” *The Journal of Finance*, vol. 25, no. 2, pp. 419–421, May 1970, doi: 10.2307/2325488.

[10] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: Forecasting and control*. John Wiley & Sons, 2015.

[11] J. Contreras, R. Espinola, F. J. Nogales, and A. J. Conejo, “ARIMA models to predict next-day electricity prices,” *IEEE Transactions on Power Systems*, vol. 18, no. 3, pp. 1014–1020, Aug. 2003, doi: 10.1109/TPWRS.2002.804943.

[12] J. A. Rangel-González, J. Frausto-Solis, J. Javier González-Barbosa, R. A. Pazos-Rangel, and H. J. Fraire-Huacuja, “Comparative study of ARIMA methods for forecasting time series of the mexican stock exchange,” in *Fuzzy Logic Augmentation of Neural and Optimization Algorithms: Theoretical Aspects and Real Applications*, 2018, pp. 475–485. doi: 10.1007/978-3-319-71008-2_34.

[13] S. M. Idrees, M. A. Alam, and P. Agarwal, “A prediction approach for stock market volatility based on time series data,” *IEEE Access*, vol. 7, pp. 17287–17298, 2019, doi: 10.1109/ACCESS.2019.2895252.

[14] P. Meesad and R. I. Rasel, “Predicting stock market price using support vector regression,” in *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, May 2013, pp. 1–6. doi: 10.1109/ICIEV.2013.6572570.

[15] J. Stanković, I. Marković, and M. Stojanović, “Investment strategy optimization using technical analysis and predictive modeling in emerging markets,” *Procedia Economics and Finance*, vol. 19, pp. 51–62, 2015, doi: 10.1016/S2212-5671(15)00007-6.

[16] R. Ślepaczuk and M. Zenkova, “Robustness of support vector machines in algorithmic trading on cryptocurrency market,” *Central European Economic Journal*, vol. 5, no. 52, pp. 186–205, Aug. 2019, doi: 10.1515/ceej-2018-0022.

[17] C. Pierdzioch and M. Risse, “Forecasting precious metal returns with multivariate random forests,” *Empirical Economics*, vol. 58, no. 3, pp. 1167–1184, Mar. 2020, doi: 10.1007/s00181-018-1558-9.

[18] K. R. Sekar, M. Srinivasan, K. S. Ravidiandran, and J. Sethuraman, “Gold price estimation using a multi variable model,” in *2017 International Conference on Networks & Advances in Computational Technologies (NetACT)*, Jul. 2017, pp. 364–369. doi: 10.1109/NETACT.2017.8076797.

[19] R. C. Cavalcante, R. C. Brasileiro, V. L. F. Souza, J. P. Nobrega, and A. L. I. Oliveira, “Computational intelligence and financial markets: A survey and future directions,” *Expert Systems with Applications*, vol. 55, pp. 194–211, Aug. 2016, doi: 10.1016/j.eswa.2016.02.006.

[20] A. Fathalla, A. Salah, K. Li, K. Li, and P. Francesco, “Deep end-to-end learning for price prediction of second-hand items,” *Knowledge and Information Systems*, vol. 62, no. 12, pp. 4541–4568, Dec. 2020, doi: 10.1007/s10115-020-01495-8.

[21] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, “Stock price prediction using LSTM, RNN and CNN-sliding window model,” in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Sep. 2017, pp. 1643–1647. doi: 10.1109/ICACCI.2017.8126078.

[22] H. S. Sim, H. I. Kim, and J. J. Ahn, “Is deep learning for image recognition applicable to stock market prediction?,” *Complexity*, vol. 2019, pp. 1–10, Feb. 2019, doi: 10.1155/2019/4324878.

[23] S. Borovkova and I. Tsiamas, “An ensemble of LSTM neural networks for high-frequency stock market classification,” *Journal of Forecasting*, vol. 38, no. 6, pp. 600–619, Sep. 2019, doi: 10.1002/for.2585.

[24] S. Hansun, F. P. Putri, A. Q. M. Khaliq, and H. Hugeng, “On searching the best mode for forex forecasting: bidirectional long short-term memory default mode is not enough,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 11, no. 4, pp. 1596–1606, Dec. 2022, doi: 10.11591/ijai.v11.i4.pp1596-1606.

[25] A. H. Bukhari, M. A. Z. Raja, M. Sulaiman, S. Islam, M. Shoaib, and P. Kumam, “Fractional neuro-sequential ARFIMA-LSTM for financial market forecasting,” *IEEE Access*, vol. 8, pp. 71326–71338, 2020, doi: 10.1109/ACCESS.2020.2985763.

[26] M. A. Mohamed, I. M. El-Henawy, and A. Salah, “Price prediction of seasonal items using machine learning and statistical methods,” *Computers, Materials & Continua*, vol. 70, no. 2, pp. 3473–3489, 2022, doi: 10.32604/cmc.2022.020782.

[27] D. Makala and Z. Li, “Prediction of gold price with ARIMA and SVM,” *Journal of Physics: Conference Series*, vol. 1767, no. 1, Feb. 2021, doi: 10.1088/1742-6596/1767/1/012022.

[28] Y. Hua, “Bitcoin price prediction using ARIMA and LSTM,” *E3S Web of Conferences*, vol. 218, Dec. 2020, doi: 10.1051/e3sconf/202021801050.

[29] G. Jain and B. Mallick, “A study of time series models ARIMA and ETS,” *SSRN Electronic Journal*, 2017, doi: 10.2139/ssrn.2898968.

[30] W. Wang, K. Chau, D. Xu, and X.-Y. Chen, “Improving forecasting accuracy of annual runoff time series using ARIMA based on EEMD decomposition,” *Water Resources Management*, vol. 29, no. 8, pp. 2655–2675, Jun. 2015, doi: 10.1007/s11269-015-0962-6.

[31] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[32] S. B. Fonseca, R. C. L. de Oliveira, and C. M. Affonso, “Short-term wind speed forecasting using machine learning algorithms,” in *2021 IEEE Madrid PowerTech*, Jun. 2021, pp. 1–6. doi: 10.1109/PowerTech46648.2021.9494848.

[33] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, Dec. 2014.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[35] C. K. Chin, D. A. binti Awang Mat, and A. Y. Saleh, “Hybrid of convolutional neural network algorithm and autoregressive integrated moving average model for skin cancer classification among Malaysian,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 10, no. 3, pp. 707–716, Sep. 2021, doi: 10.11591/ijai.v10.i3.pp707-716.

[36] A. Issam, A. K. Mounir, E. M. Saida, and E. M. Fatna, “Financial sentiment analysis of tweets based on deep learning approach,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 25, no. 3, pp. 1759–1770, Mar. 2022, doi: 10.11591/ijeecs.v25.i3.pp1759-1770.

[37] F. Chollet, *Deep Learning mit python und keras: Das praxis-handbuch vom entwickler der keras-bibliothek (mitp Professional)*. German: mitp; 2018th edition, 2018.

[38] M. Abadi *et al.*, “TensorFlow: a system for large-scale machine learning,” in *OSDI’16: Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*, 2016, pp. 265–283.

[39] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in python,” *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.

[40] Yahoo, “Yahoo Finance - Stock market live, quotes, business & finance news,” *Yahoo Finance*, 2022. https://finance.yahoo.com/ (accessed Apr. 01, 2021).

[41] H. Vaheb, “Asset price forecasting using recurrent neural networks,” *Prepr. arXiv2010.06417*, Oct. 2020.

[42] H. K. Choi, “Stock price correlation coefficient prediction with ARIMA-LSTM hybrid model,” *Prepr. arXiv1808.01560*, Aug. 2018.

BIOGRAPHIES OF AUTHORS

Walid Abdullah holds a bachelor's degree in computers and information from Zagazig University, Egypt, 2018. He is currently working as a teaching assistant in the Department of Computer Science at the Faculty of Computers and Information, Zagazig University, Egypt. His research interests include artificial intelligence, machine learning, and time series analysis using artificial intelligence techniques. He can be contacted at walidaim2@gmail.com.

Ahmad Salah received a Ph.D. degree in computer science from Hunan University, China, in 2014. He received a master’s degree in CS from Ain-Shams University, Cairo, Egypt. He is currently an associate professor of Computer Science at Zagazig University, Egypt. He has published more than 30 papers in international peer-reviewed journals, such as the IEEE Transactions on Parallel and Distributed Systems and IEEE-ACM Transactions on Computational Biology Bioinformatics, and ACM Transactions on Parallel Computing. His current research interests are parallel computing, computational biology, and machine learning. He can be contacted at ahmad@zu.edu.eg.