Multimedia Tools and Applications (2021) 80:17923–17954
https://doi.org/10.1007/s11042-021-10579-8

StockPred: a framework for stock Price prediction

Marwa Sharaf[1] · Ezz El-Din Hemdan[1] · Ayman El-Sayed[1] · Nirmeen A. El-Bahnasawy[1]

Received: 5 June 2020 / Revised: 5 January 2021 / Accepted: 20 January 2021 /
Published online: 12 February 2021

Abstract
Recently, Stock Price prediction becomes a significant practical aspect of the economic arena. The stock price prediction is generally considered as one of the most exciting challenges due to the noise and volatility characteristics of stock market behavior. Therefore, this paper proposes a framework to address these challenges and efficiently predicting stock price using learning models such as Long Short Term Memory (LSTM), Convolutional Neural Network (CNN), Support Vector Machine (SVM), Linear Regression, Logistic Regression, K-Neighbors, Decision Tree, Random Forest, Stacked-LSTM, and Bidirectional-LSTM. Numerous experiments with different scenarios are performed to evaluate the projected framework with the stock price dataset. The results demonstrate that the applied models within the framework such as the CNN model outperformed the other models in stock price prediction at different circumstances based on several evaluation metrics like R-Square (R2), Root Mean Square Error (RMSE), Root Mean Square (RMS), Mean Square Error (MSE), Mean Average Error (MAE) and Mean Average Percentage Error (MAPE).

Keywords Machine learning · Deep learning · LSTM · CNN · SVM · Stock sentiment analysis · Financial data · And prediction

✉ Ezz El-Din Hemdan
ezzvip@yahoo.com

Marwa Sharaf
eng.marwa.sharaf@el-eng.menofia.edu.eg

Ayman El-Sayed
ayman.elsayed@el-eng.menofia.edu.eg

Nirmeen A. El-Bahnasawy
nirmeena.el-bahnasawy@el-eng.menofia.edu.eg

[1] Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University, Menoufia, Egypt

1 Introduction

Presentably, the financial sector is a part of the economy composed of firms and institutions that introduce financial services to commercial and retail customers around the world. This sector includes a wide range of businesses including banks, investment businesses, insurance firms, and real estate organizations [25]. The financial market is a dynamic and composite system where people can transport, buy, and sell currencies, stocks, and derivatives over virtual platforms introduced by agents [40].

Financial stock is a significant factor in the economy. The stock market enables stockholders from owning shares of public companies through the trading process either by exchange or over-the-counter markets. This market has given investors the chance of earning money and having a prospering life through investing small initial amounts of money [40]. The ability to predict stock prices is an important issue regarding the academic area as well as a business [11]. Prediction of stock price behavior is an area of strong effect for both academic researchers and industry practitioners, as it is both a difficult task and could lead to increased profits [45].

The stock market is the area where stocks can be transferred, traded, and distributed. It has been considered an important channel for large companies to raise funds from stockholders. On the one hand, through the issuance of stocks, a large amount of capital flows into the stock market, which enhances the organic configuration of corporate capital by promoting capital Concentration [28]. But, the prediction process of the stock price is generally regarded as a difficult and complex process due to stock properties such as noise and volatility. However, the development mechanism of stock price prediction is quite complicated. As it requires mixing in use between the various factors and the special behavior of individual factors, including political, economic, and market factors as well as technology and investor behavior, it will all lead to changes in stock prices [28].

In the current stock markets, the view of investors whether positive or negative is a remarkable indicator of the future value of that stock [11]. Sentiment analysis for the stockholder's view is a critical process that affects the next movement of this stock. Therefore, this process must be done accurately and efficiently [11]. The successful prediction of a stock's future price will increase stockholder's wins [19]. The contributions of this paper can be concise as follows:

- Provide a review of an existing machine and deep learning models that can be used in financial analysis.
- Recommend an effective way that helps the investor from making the correct decision about the buying or selling stock market as well as helping in protecting the investor from taking a risk about his stock market.
- Develop an efficient analytical framework named StockPred based on machine learning and deep learning models for stock price prediction.
- The applied models in the proposed framework can efficiently classify and predict stock prices. The results show that the CNN model performs stock price prediction proficiently than the others. Therefore, using CNN in the financial segment can make performance enhancement in building real-time stock price prediction applications.

The rest of this paper is prepared as follows Section 2 provides preliminary knowledge regarding stock market analysis, and machine learning techniques while the related work in stock prediction is presented in Section 3. Section 4 presents the proposed stock price prediction framework while the experimental study and results analysis is delivered in Section 5. Finally, the paper conclusion and future work in this inventive subject are introduced in Section 6.

2 Preliminary knowledge

This section provides an overview of stock market analysis, and machine and deep learning techniques.

2.1 Stock market analysis

Lately, there are many applications for the financial sector such as stock market prediction, fraud detection, and prevention, risk analysis, web analytics, etc. [41]. As, there are a huge amount of data generated by the financial sector like client data, logs from their financial products, operations data. That can be integrated with external data, like social media data and data from websites to help in the decision-making process [41]. Such Big Data can be preserved as an excessive source of real-time estimation because of its creation at high frequency and acquisition with little cost [50].

In recent times, the process of stock market prediction has become a very important topic for businesspeople [39]. Sentiment analysis is a useful tool used for stock price prediction. By analyzing the historical data of the stock and/ or online data related to the stock to predict the future price of this stock. Also, it can be the best step for the recommendation system that can recommend the best time for selling and buying the stock [39]. By analyzing the financial news by the sentiment analysis process, the system can determine the direction of the movement of the stock. Sentiment analysis provides automatic extraction of views, emotions from the opinionated contents. It is used to automatically extract views, attitudes, and emotions from the opinionated contents.

2.2 Machine learning techniques

Prediction of a stock price is not a simple process, mainly because of the random movement behavior of a stock time series [19]. Recently, there are many machine learning (ML) algorithms and deep learning (DL) techniques that can be used for stock price prediction. ML algorithms such as Linear Regression, Logistic Regression, Random Forest, K-Neighbors, Support Vector Regression (SVR, and Decision Tree). The explanation of ML techniques as follows [43]:

- ***Linear Regression:*** Linear Regression represents the relationship between scalar variables by finding a linear equation to fit the data. The relationships are defined using linear predictor functions, with undefined model parameters estimated from the data set. Linear regression is used for:

1- Prediction of the dependent variable value where the independent variable value is known [54].

2- Estimation of the variance proportion in the dependent variable that is described by an independent variable.

The operation of linear regression [18] is illustrated in Fig. 1, the linear regression equation is described in equation1.

$$\mathbf{Z} = \mathbf{a} + \mathbf{b}\,\mathbf{X} + \mathbf{e} \tag{1}$$

Where:

Z the predicted value for the dependent variable, it is called the response variable
x a random variable called a predictor variable
a Intercept, the value of Z when X = 0
b it is regression line slope; it is the rate of change in Z with a unit change in X
e it is a random error

- ***Logistic Regression:*** Logistic Regression can handle both nominal and numerical data. It is based on one or more predictor features for calculating binary response probability. It is used when the value of the target variable is categorical. The operation of logistic regression is explained in Eq. (2). The operation of logistic regression is illustrated in Fig. 2 while Fig. 3 illustrates the difference between linear and logistic regression [51].

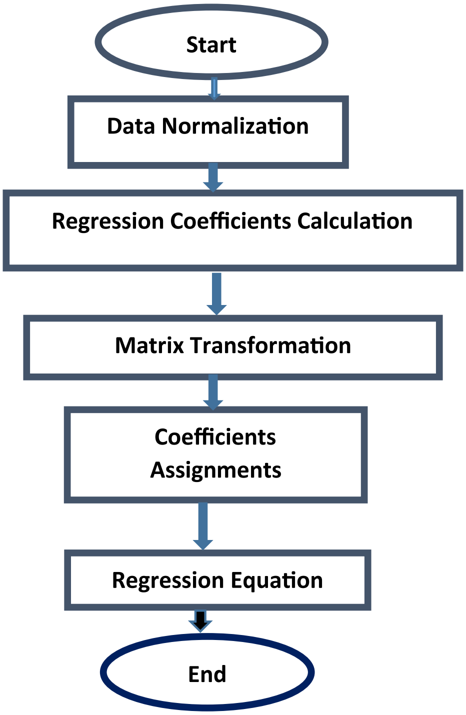
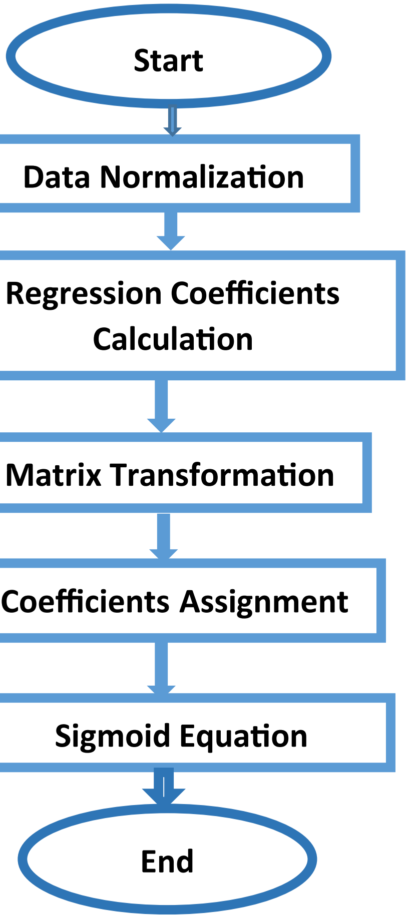


Fig. 1 Linear regression process

Fig. 2 Logistic Regression process



$$W = \frac{\mathbf{1}}{\mathbf{1} + \mathbf{e}^{-(\mathbf{a}+\mathbf{bx})}} \tag{2}$$

Where

W	The Predicted Value for input X
a	Intercept, the value of W when x = 0.
b	it is the regression line slope; it is the rate of change in W with a unit change in X.
x	**a** random variable called a predictor variable.

- ***Decision Tree:*** The decision Tree is a collection of nodes that are connected as the flowchart. These nodes introduce a decision on features connected to certain classes. Every node represents a splitting rule for a feature. New nodes are created until the stopping criterion is met. The sample majority that belongs to a leaf is the basic factor in the determination of class labels. A decision tree is created using either a top-down

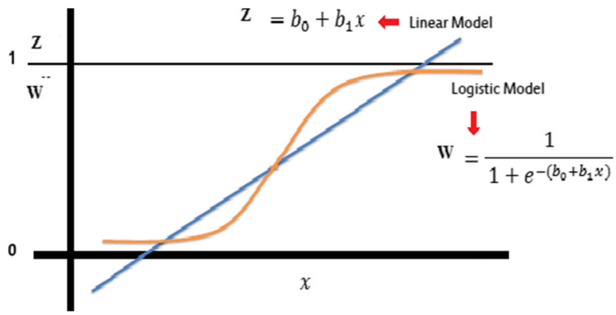


Fig. 3 Linear regression against logistic regression [51]

algorithm or a bottom-up algorithm. The steps of the decision tree algorithm are described in algorithm 1.1 [18, 47]. There are types of decision trees such as Iterative Dichotomiser (ID3) [34, 58], C4.5 (a successor of ID3) [2, 24], Classification and Regression Trees (CART) [61].

- ***Random Forest (RF):*** Random Forest creates an ensemble of random trees. The user sets the number of trees. The resulting model employs voting of all created trees to determine the final classification outcome [43]. A random tree is a tree drawn from a set of possible trees randomly, as randomly means that each tree has the same chance to be a random tree [8]. Random Forest is defined as a classifier consisting of a collection of tree-structured classifiers. The main idea of RF is that operating many relatively uncorrelated models (trees) will outperform any of the individual constituent models**.** There are two ways for combining outputs from multiple used trees such as bagging and boosting. The output class is determined from the voting phase that is based on selecting the output of the tree with major output [30]. The operation of RF is described in Fig. 4.
- ***Support Vector Regression (SVR):*** Because SVR has a nonparametric property [3], it is widely used for classification and regression. SVM is used for the classification of linear and nonlinear data [22]. The operation of SVM is based on the use of kernels [3]. It converts the initial training data to a higher dimension using nonlinear function [14]. Then, for the process of separating records, it searches for the linear most suitable separating hyperplane. Many proposals are based on the improvement of SVR such as [21, 33, 62, 63]. SVR is a type of SVM that is used in regression to minimize the error. The equation of SVR is described in Fig. 5.

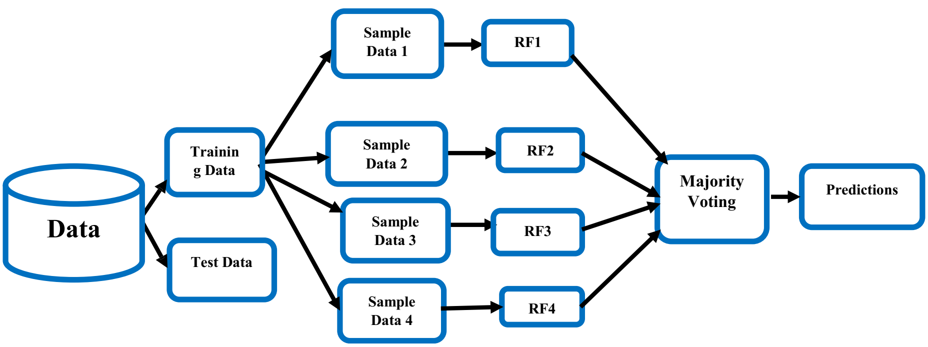


Fig. 4 RF Flowchart

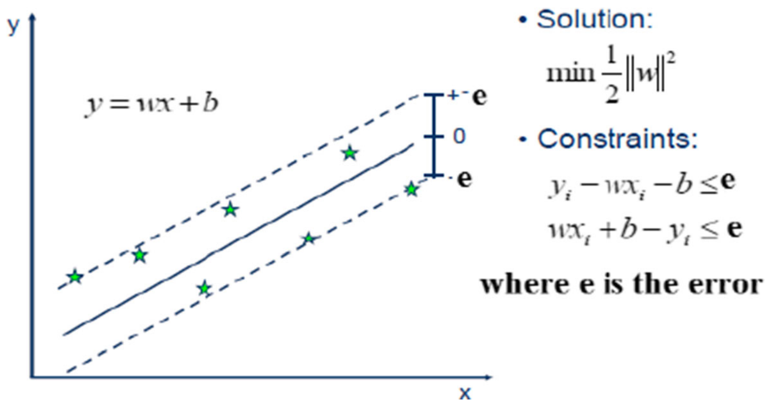


Fig. 5 Operation of SVR [14, 22]

- ***K- Nearest Neighbors (KNN):*** In the nearest neighbor technique, the classification of an unknown data record is performed by analyzing the nearest neighbor's labels [27]. KNN algorithm follows the nearest neighbor's principle. But, the KNN algorithm allows a fixed number of nearest neighbors to vote in the process of classification of an unknown data record which is identified by k, where k is a positive integer. If k = 1, then the unknown data record is classified as the class of the closest training data tuple to it [27]. To find the nearest data points to unknown records, several distance measures are used such as Euclidean distance, Minkowski distance, Manhattan distance as described in Eqs. (3), (4), and (5) [31]. The operation of KNN is described in Fig. 6.

1. Euclidean distance: the distance between two data point is calculated as described in Eq. (3):

$$dist((x,y),(a,b)) = \sqrt{(x-a)^2 + (y-b)^2} \tag{3}$$

2. Manhattan distance, the distance between two data point is calculated as described in Eq. (4):

$$dist((x,y),(a,b)) = |x-a| + |y-b| \tag{4}$$

3. Minkowski distance, the distance between two data point is calculated as described in Eq. (5), is a generalization of the Euclidean and Manhattan distances

$$\mathbf{dist}((\mathbf{x},\mathbf{y}),(\mathbf{a},\mathbf{b})) = \mathbf{n}\sqrt{|\mathbf{x}-\mathbf{a}|^{\mathbf{n}} + |\mathbf{y}-\mathbf{b}|^{\mathbf{n}}} \tag{5}$$

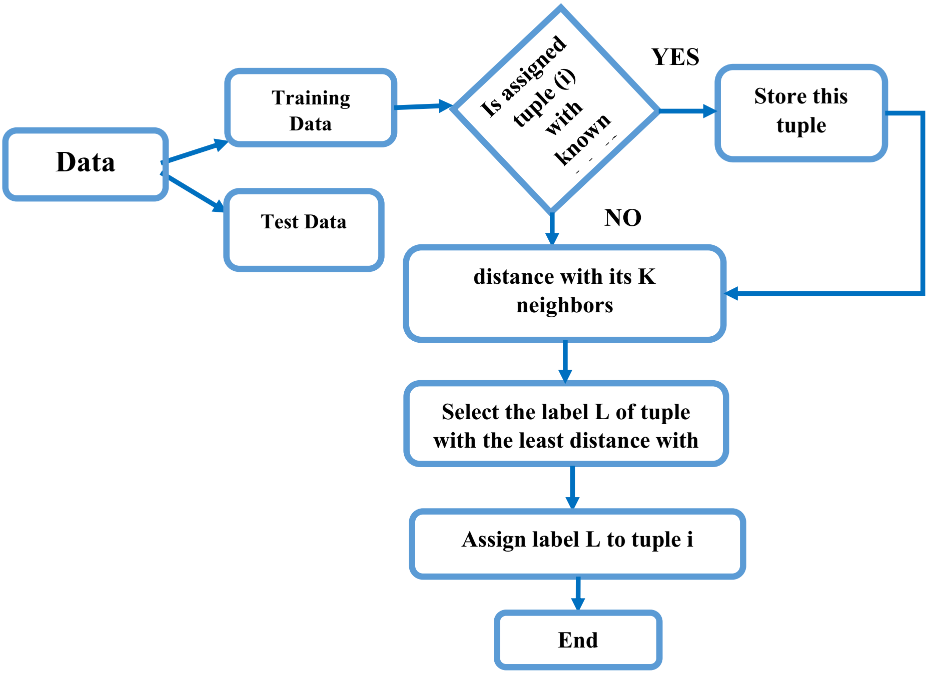


Fig. 6 KNN Operation

2.3 Deep learning techniques

Deep Learning is an effective research topic that belongs to Artificial Intelligence (AI). Deep Learning refers to machine learning techniques based on supervised and unsupervised methods to automatically learn hierarchical representations in deep architectures [52]. It tries to simulate the human brain, which is capable of processing complicated input data and learning different knowledge intellectually to solve complex problems. Deep Learning comes from the human brain concept which has multiple types of representation with simpler features at the lower and higher-level abstractions built on top of that.

Humans arrange the ideas hierarchically as first, humans learn simple basics then shape them to represent more abstract ones. The human brain is composed of several neurons likes processing elements that are used in a deep neural network. These neurons act as feature detectors, detecting more abstract features as the levels go up. The basic aim of Deep Learning is to extract more basic features in the higher levels of the representation using Artificial Neural Networks (ANN). The ANN structure is described in Fig. 7 [32].

In a deep network, each layer enters in pre-training phase with unsupervised learning techniques, resulting in a nonlinear transformation of its input or the output of the previous layer and extract more essential features from its input. Finally, the deep architecture is adjusted regarding a supervised training principle at the last training phase.

There are three major classes for deep networks architectures as follows [9, 10, 20]:

1. **Deep networks for unsupervised learning:** This network is designed to extract the high-order association of the observed data for pattern analysis with unknown target labels. By

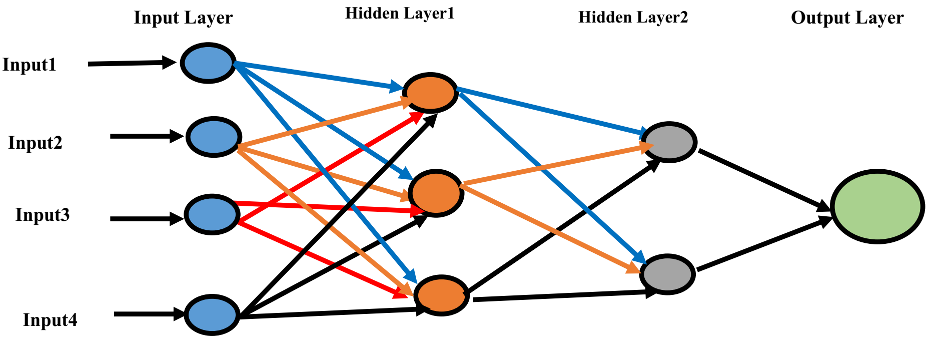


Fig. 7 ANN Architecture

using unsupervised learning in generative case, it may also be developed to characterize joint statistical distributions of the visible data and their associated labels.

2. **Deep networks for supervised learning:** This type of learning also known as discriminative deep networks proposed to support discriminative power for pattern classification directly. This occurs often by characterizing the back distributions of classes conditioned on the visible data. Target label data are always defined in direct or indirect forms for such supervised learning.
3. **Hybrid deep networks:** Where the goal is segregation that is supported, often in a significant way, with the results of unsupervised deep networks. This can be designed via better optimization of the deep networks for supervised learning. The target can also be achieved when discriminative criteria for supervised learning are used to estimate the parameters in any of the unsupervised deep networks.

Recently, there are several DL techniques that can be used for predicting stock prices such as Convolutional Neural Network (CNN), Long-Short Term Memory (LSTM), Stacked-LSTM, and Bidirectional LSTM. The explanation of these algorithms as follow:

- **Convolutional Neural Network (CNN)**

CNN is a class of traditional ANNs in that they are composed of neurons that self-optimize through learning. Each neuron will still receive input and operate [38]. CNN consists of an input layer, an output layer, as well as multiple stages of hidden layers as shown in Fig. 8. CNN's hidden layers are composed of a set of convolutional layers followed by a set of pooling layers followed by fully connected layers and normalization layers (RELU). CNN is developed to learn spatial hierarchies of features automatically through backpropagation by using its basic blocks (convolution layers, pooling layers, and fully connected layers) [59]. Examples of a typical CNN can be described [26, 38, 59] as follows: Input Layer, Convolutional Layer (Convolution+ RELU), Fully connected layers, Pooling Layer.

- **Long-Short Term Memory (LSTM)**

As CNN is a feedforward network so, there is no feedback from output to input. This can be done by backpropagation that results in expanding or vanishing of parameters. Therefore, CNN faces the problem of overfitting/ underfitting when using the backpropagation. This problem is solved by applying recurrent neural networks (RNN) that have at least one loop

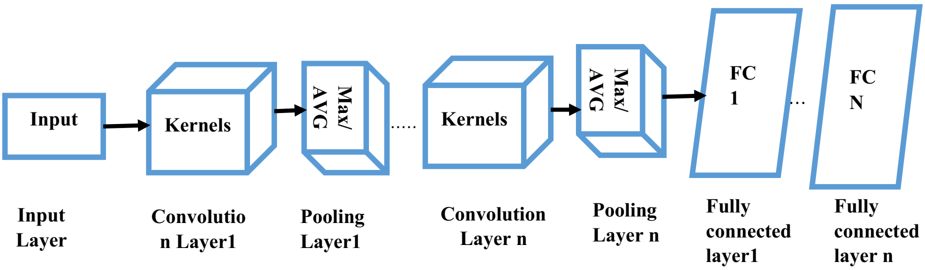


Fig. 8 CNN Architecture

from output to input. This operation also can be described by Eqs. (6) and (7) as following [60]. Figure 9 describes the operation of RNN [48].

$$h_t = H(w_{xh}X_t + w_{hh}h_{t-1} + b_h) \tag{6}$$

$$Y_t = (w_{yh}h_t + b_o) \tag{7}$$

Where:

H	hidden layer function (sigmoid function)
w_{xh}	weight of input hidden layer
w_{yh}	weight of hidden layer in the back direction
b_h	bias of hidden
h_{t-1}	previous state of the output
h_{t-1}	current state of output in forwarding direction
X	input vector
H	hidden vector
Y	output vector

But, RNN faces a problem in storing the input states for a long period. So, there is a need for LSTM that can store input for a long using a special unit called memory/ state unit [48]. LSTM store the previous states of input in its memory cell, so it can work efficiently with long term models and solves the long-term problem of RNN. The architecture of LSTM is

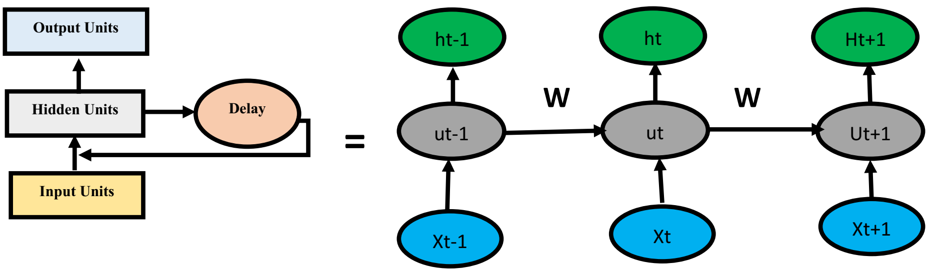


Fig. 9 RNN Operation

illustrated in Fig. 10 [16, 57]. The LSTM can delete or add information from or to the cell state, efficiently performed by structures called gates.

LSTM-gates are a technique to optionally pass information through. They consist of a sigmoid neural network layer and a pointwise multiplication operation. An LSTM is composed of three gates, to monitor and protect the cell state. Steps of LSTM operation are as follows [16]:

The first step LSTM is to determine what information will leave the memory cell. A sigmoid layer called the "forget gate layer" is responsible for this step. Operation of this step is described in Eq. (8) as follows:

$$\mathbf{f(t)} = \sigma(\mathbf{Wf}.[\mathbf{h(t{-}1)}, \mathbf{X(t)}]) + \mathbf{bf} \tag{8}$$

Where:

σSigmoid net layer

Wfforget layer weight

h(t-1)output of the previous stage

X(t)current input

bfbias of forget layer

The second step is to decide on what new information will be stored in the memory cell. There are two layers, the first one is the sigmoid layer called the input gate layer in and a second layer is a tanh layer that creates a vector of new candidate values, which are used to control the second step. Equations (9)–(10) describe the operation of input and tanh layer as follows:

$$\mathbf{i(t)} = \sigma(\mathbf{Wi}.[\mathbf{h(t{-}1)}, \mathbf{X(t)}]) + \mathbf{bi} \tag{9}$$

$$\mathbf{T(t)} = \mathbf{tanh}(\mathbf{Wc}.[\mathbf{h(t{-}1)}, \mathbf{X(t)}]) + \mathbf{bc} \tag{10}$$

Where:

σSigmoid net layer

Wiinput layer weight

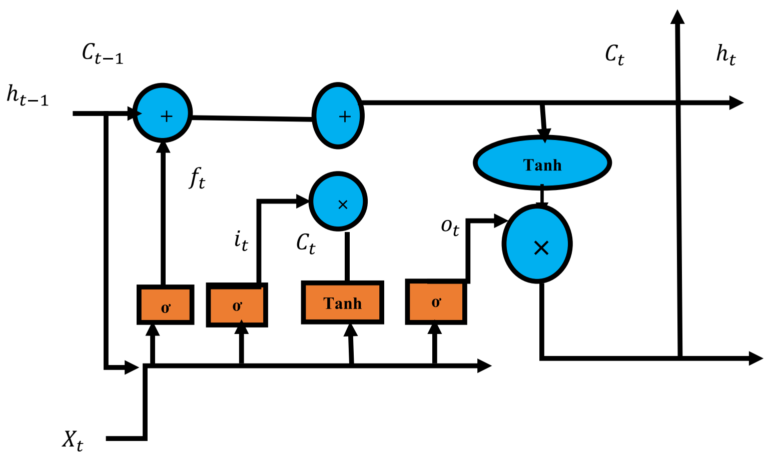


Fig. 10 LSTM Architecture

Wctanh layer weight
h(*t*-1)*output of previous stage*
X(*t*)*current input*
bibias of input layer,
bcbias of tanh layer

So, the summation of the output of each layer will decide the current state of the memory cell as in Eqs. 11, 12, and 13:

$$\mathbf{C(t)=ft*C(t-1)+i(t)*T(t)} \tag{11}$$

Where: C(t-1) = previous state of memory cell

The final step in LSTM operation is to decide what the output is. The fourth sigmoid net layer performs this operation as follows:

$$\mathbf{o(t)} = \sigma(\mathbf{Wo.[h(t-1), X(t)]}) + \mathbf{bo} \tag{12}$$

$$\mathbf{ht} = \mathbf{o(t)\ x\ tanh(Ct)} \tag{13}$$

Where: ht. = the final output of the current stage.

- **Stacked-LSTM**

LSTMs can process sequence data [29] and are applied in many practical applications such as speech recognition [17], image captioning [56], music composition [13], and human trajectory prediction [7]. The deep LSTM is networking with several stacked LSTM hidden layers, in which the output of an LSTM hidden layer will be fed back as the input into the subsequent LSTM hidden layer as described in Fig. 11 [7]. This network is known as bidirectional LSTM. LSTM is composed of multiple layers of stacked hidden layers but, output goes through a forwarding direction (Stacked-LSTM) [49, 64]. This stacked-layer mechanism can enhance the power of neural networks by applying deep learning on the data to provide more accurate predictions. Passing the output from each layer as the input of the next layer allows the model from deeply feature extraction process.

- **Bidirectional LSTM**

It is composed of two separate hidden layers; to generate the output ***Yt***. First, it computes the hidden vector $\overrightarrow{h_t}$ in the forward direction as described in Eq. (14). Then, it computes it in a backward direction $h_t^{\leftarrow}$ as described in Eq. (15). Finally, it combines these two vectors as described in Eq. (16) [5, 64]. Figure 12 describes the structure of a bidirectional LSTM [1].

$$\overrightarrow{\mathbf{h}}_{\mathbf{t}}=\mathbf{H}\left(\mathbf{w}_{\mathbf{x}\overrightarrow{\mathbf{h}}}\mathbf{x}_{\mathbf{t}}+\mathbf{w}_{\overrightarrow{\mathbf{h}}\overrightarrow{\mathbf{h}}}\overrightarrow{\mathbf{h}}_{\mathbf{t-1}}+\mathbf{b}_{\overrightarrow{\mathbf{h}}}\right) \tag{14}$$

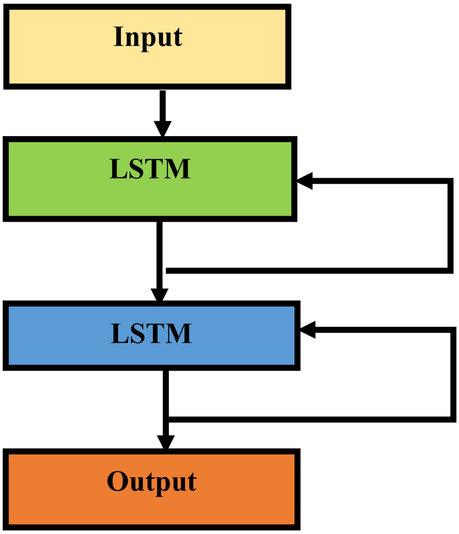


Fig. 11 Stacked LSTM

$$\mathbf{h_t}^{\leftarrow} = \mathbf{H}\left(\mathbf{w}_{\mathbf{xh}^{\leftarrow}} \quad \mathbf{x_t} + \mathbf{w}_{\mathbf{h}^{\leftarrow}\mathbf{h}^{\leftarrow}} \mathbf{h}^{\leftarrow}{}_{\mathbf{t-1}} + \mathbf{b}_{\mathbf{h}^{\leftarrow}}\right) \tag{15}$$

$$\mathbf{Y_t} = \left(\mathbf{w}_{\mathbf{y}\overrightarrow{\mathbf{h}}} \ \overrightarrow{\mathbf{h_t}} + \mathbf{w}_{\mathbf{h}^{\leftarrow}\mathbf{h}^{\leftarrow}} \ \mathbf{h_t}^{\leftarrow} + \mathbf{b_y} \right) \tag{16}$$

3 Related work

This section provides a discussion on prior relevant literature on stock market prediction. In 2020, the researchers [4] presented a model based on Stacked LSTM and Multilayer

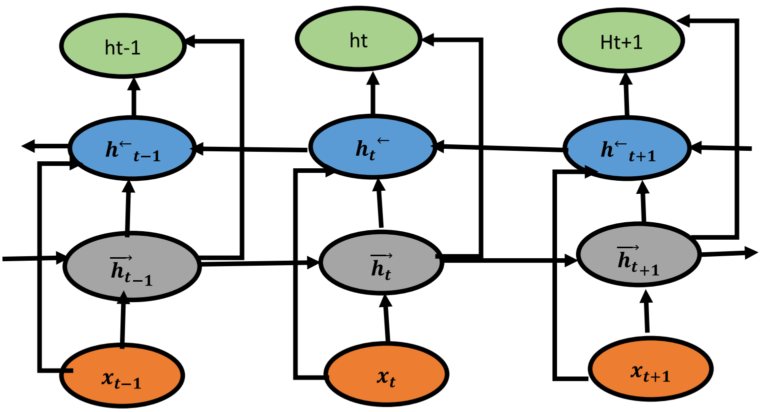


Fig. 12 Bidirectional LSTM

perceptron (MLP) to enhance the accuracy of stock prediction. The authors [42] proposed a stock prediction model that is based on LSTM, an attention technique and uses the wavelet transform to clear noise from historical stock data. In 2019, the authors [12] proposed an associated deep recurrent neural network model with several inputs and several outputs based on LSTM. The presented model can enhance the accuracy of prediction by up to 95%. In [28], the authors proposed an empirical modal decomposition (EMD), which is based on the improvement of LSTM to improve time delay and prediction accuracy.

Also, in [37], the researchers introduced a new outline, called a deep transfer with related stock information (DTRSI), that is characterized by a DNN and transfer learning characteristic. It is based on extracting the relationship between different stocks using LSTM that results in performance improvement. In [11], the authors proposed a stock prediction model and test it on two different datasets based on sentiment analysis of customer opinion. In [23], the authors proposed a stock market prediction model that is based on the CNN framework. This model improved the performance of the stock prediction process. In [53], a new model of stock movement prediction was proposed that was based on two methods namely SVR and nonnegative matrix factorization. The prediction accuracy was enhanced to 60.16%.

Also, the authors [15] proposed Convolutional Recurrent Neural Network (CRNN)-based architecture, in which long and short-term memory is used in RNN (Long-Short). -term memory, LSTM) architecture. LSTM solves the long-term problem of traditional RNN and effectively enhances the accuracy and stability of prediction. In [44], the authors made integration between sentiment analysis and machine learning method based SVR. This model improved prediction accuracy SSE 50 Index movement up to 89.93%. In [35], the authors proposed a novel machine learning model to predict stock movement by analyzing the financial news. This model proved that it can perform its task even if there is no news available on the target. This reference can be used as our background in our future work and can be improved in the future to predict stock close price by applying an effective sentiment analysis on finance news and customer opinion from social media.

In 2018, the authors [36] designed an online learning technique that uses a kind of recurrent neural network (RNN) called Long-Short Term Memory (LSTM), that enhances the prediction accuracy. In [46], the authors applied sentiment analysis of stock news and customer opinions to improve stock prediction accuracy by up to 70.59%. Also, the authors [50] introduced a strong comparative study between the ML models such as logistic regression and DL models such as CNN and LSTM that prove that DL Models have better performance than ML models. In [55], a stock price prediction model was proposed using information from both numerical analysis and textual analysis. The numerical analysis was accomplished using the LSTM model with a sliding window. Then textual analysis was performed on the news articles.

In addition to [6], a deep learning method based on CNN was proposed to predict the stock price movement of the Chinese stock market. In 2013, the authors [19] provided a stock prediction model based on the integration of Particle swarm optimization (PSO) and least square support vector machine (LS-SVM).

It's noticed that several authors introduced much research on stock analysis, most of the work applying machine learning models only for the prediction of stock price and little dealing with deep learning. Considering the limitation in related work for stock market price prediction techniques, the main contribution of this paper is to represent a proficient framework called StockPred to explore and utilizing different deep learning and machine learning techniques to select and provide the best and effective model suitable for stock price prediction in an efficient manner.

Due to the noise of stock data and random movement of stock close price, it is difficult to analyze stock data carefully with traditional ML techniques. As the main problem for the investor is to understand how the stock market changes to be able to decide which stock is to buy or to sell. As stock future price is based on some items such as the company's profile, the historical price, and the finance news about this stock. In our framework, stock price prediction is based on an analysis of historical prices to predict a more accurate future price. Finally, in this framework, we introduce some ML and DL models to allow investors from predicting the future price of the stock market and make the best decision about the buying or selling process. As our work aims to present an efficient framework for stock price prediction models. This paper aims to help investors in deciding on buying or selling the stock market. This paper aims to prevent the investor from taking losing his money if he buys the stock at a decreased price.

4 Proposed StockPred framework

The stock prediction process in the StockPred framework involves several stages such as data collection (stock dataset), data preprocessing, setting up stock prediction model, model evaluation finally the presentation of predictions to the user as described in Fig. 13. Likewise, Fig. 14 describes the architecture of the proposed stock price prediction framework.

As shown in Fig. 13, the stock prediction process in the proposed framework can be summarized as follows:

- **Phase 1 (data collection):** It is where stock data are collected and fed to the preprocessing phase.
- **Phase 2 (data preprocessing):** This phase is which transforms data into a form suitable for analysis.
- **Phase 3 (Modeling):** This involves the development of a stock prediction model and fed the transformed data to it. In this step, the data analysis is performed to predict stock close price accurately.

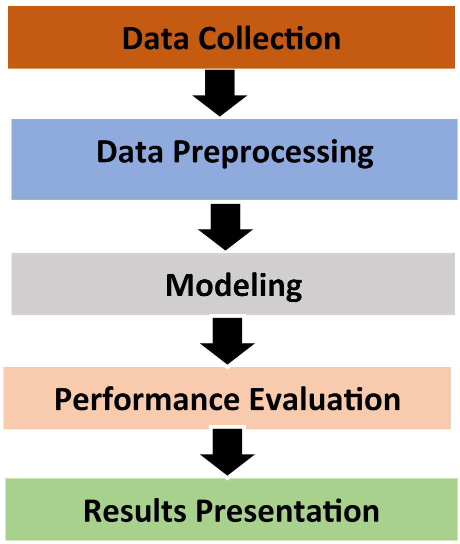


Fig. 13 Phases of Stock Price Prediction Model

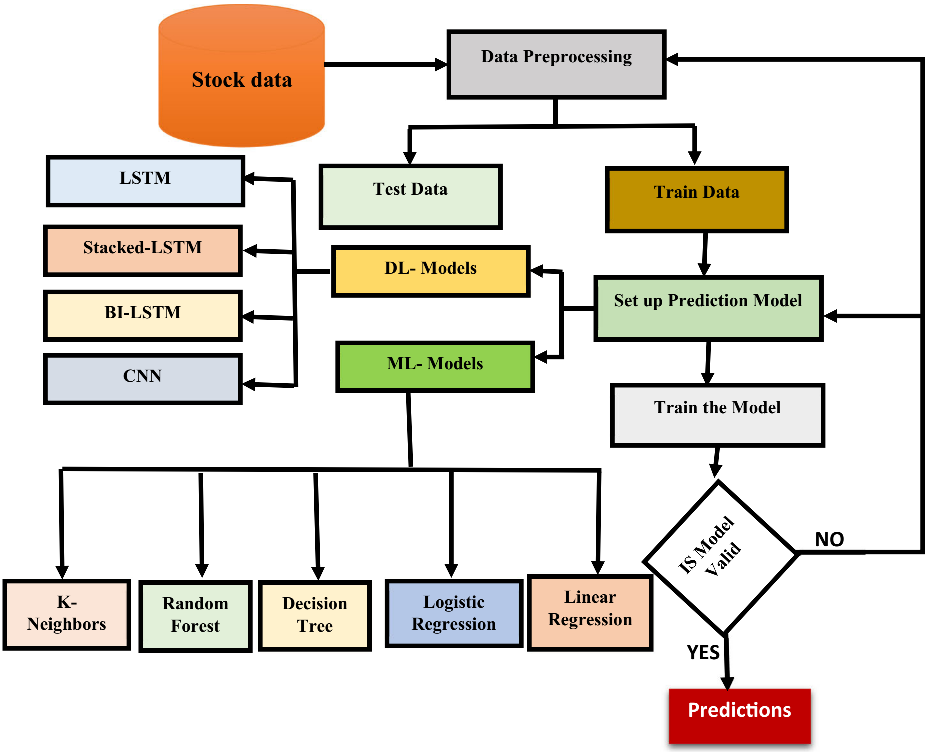


Fig. 14 Proposed StockPred Framework

- **Phase 4 (Performance Evaluation):** it is where the output of the model is tested against real output then check if this model is valid or not. If this model does not provide the proposed output, then there is a need for model enhancement or data preprocessing enhancement. Else, communicate the results to the user.
- **Phase 5 (Results Presentation):** It is the last phase, where the model is visualized and the results and prediction are presented to the user.

The steps of the proposed prediction framework can be as the following:

- **Step 1:** Where stock data are collected from the historical stock dataset that is requested and processed in a real-time process. So from the benefits of our framework, there are no storage requirements for historical stock data
- **Step 2:** The stock dataset is fed to the preprocessing step where some preprocessing techniques are applied such as data normalization, feature selection, and data reduction. These steps can be considered as important as it reproduces the stock dataset in the form suitable for processing and analysis.

a. Data Normalization: the stock dataset attributes contain some attributes such as the open, high, close, and volume attributes. From the study of the state of each attribute, we find that the values of the first four attributes change in the range of small values such as 15.1 and 200, where the values of volume attribute change in the range of large values such as

92,303,500 to 90,648,055. From this study, there is a large difference in data scaling. So data normalization technique is used to scale the stock data in the range from 0 to 1.

b. Data Reduction: this sub-process is used to reduce the dimensionality of data by creating our used framework with determined attributes and a determined range of data records.

All the applied sub-processes in the preprocessing phase are applied to enhance the performance of its next-hop (processing step) and to reduce resource consumption.

- **Step 3:** Where the preprocessed dataset is split to train and test datasets, as the training dataset is used for the training phase of the analysis process to train the different machine learning models with the nature of stock data variations. Then test dataset is used for the verification and validation phase to check how the ML model can predict the target value (close price). In this phase, we build our framework with a satisfied number of ML and DL models to clear for the user the benefits and drawbacks in each of them proved with practical results.
- **Step 4:** Where some evaluation metrics are used to measure the accuracy using the R-Square metric and error using the Root Mean Square error metric (RMS) of ML models. Where in this framework stock historical data is time-series data, so some continuous metrics such as accuracy, F1-score, precision, are not permitted in our evaluation process.
- **Step 5:** In this step, the results are visualized and presented to clear the benefits and drawbacks of each ML model.

5 Experimental study and results analysis

This section describes the used datasets, performance evaluation metrics, and the experimental study with results analysis for the proposed framework.

5.1 Dataset

In this work, we use the stock price dataset from quandl. The description of this dataset is shown in Fig. 15 which composed of several attributes such as open, close, adjusted-close, adjusted -low, and adjusted-volume as open attribute presents the open price of a stock for the next day, where the closed attribute presents the stock price on this day. The adjusted-attributes present the direction of the stock movement as adjusted-close presents a stock's closing price to accurately reflect that stock's value after accounting for occurred commercial operations. In our experiment, we request the stock data in the range from 31/12/2000 until 31/12/2019. This dataset contains data about amazon stock that describe the stock price movement. The statistical description of the used dataset is shown in Fig. 16.

5.2 Performance evaluation metrics

To evaluate the proposed system, several metrics can be used as the following:

- **Confidence/R-Square**: This metric determines how well the model work; its mathematical formula is described in Eq. (17) [28].
- **Root Mean Square Error (RMSE):** It is the root of squared measured error (measure difference between Actual and predicted values, it is described in Eq. (18) [28].

quandl Stock Dataset

	Open	High	...	Adj. Close	Adj. Volume
Date			...		
2001-01-02	15.81	16.00	...	13.88	9203500.0
2001-01-03	13.63	17.88	...	17.56	14680400.0
2001-01-04	17.00	17.56	...	15.50	10620500.0
2001-01-05	15.50	15.88	...	14.56	8798800.0
2001-01-08	14.44	15.56	...	14.94	10444700.0
2001-01-09	14.56	16.63	...	16.38	23455200.0
2001-01-10	15.84	17.00	...	16.50	11002500.0
2001-01-11	15.75	17.31	...	17.00	12756600.0
2001-01-12	17.34	18.50	...	17.69	8434200.0
2001-01-16	18.31	18.38	...	18.06	6017000.0
2001-01-17	18.88	19.56	...	18.38	9166000.0
2001-01-18	18.38	19.50	...	19.50	4115800.0
2001-01-19	20.38	20.38	...	19.94	6177500.0
2001-01-22	19.50	19.94	...	18.50	4398600.0
2001-01-23	18.38	19.31	...	18.95	4744300.0
2001-01-24	19.19	22.38	...	21.88	13848100.0
2001-01-25	21.69	21.75	...	19.00	7066400.0
2001-01-26	18.94	20.50	...	19.50	5094600.0

Fig. 15 The used dataset description

- **Root Mean Square (RMS):** Average of difference between actual and predicted values, its operation is described in Eq. (19) [28].
- **Mean Square Error (MSE):** It is the average of the squared error used as a loss function, it is described in Eq. (20) [28].
- **Mean Absolute Error (MAE):** It is the mean of the absolute values of the individual prediction errors on all samples in the test data as described in eq. (21) [28].
- **Mean Absolute Percentage Error (MAPE):** It is a statistical measure of how accurate a forecast system is. It measures this accuracy as a percentage and can be calculated as the average absolute percent error for each time minus actual values divided by actual values, as described in Eq. (22) [28].
- **Processing Time**: It is the time taken by the system for the prediction process. That is calculated by calculating the period between loading data and providing the predictions as described in eq. (23). The current time can be measured using time. Time () function that is supported by the python programming language. This function returns the time in seconds in epochs. This function is called twice. First at the beginning of the processing to measure the start time, then it is called at the end of processing to measure the end time. Then by calculating the difference between the end and start time, we can measure the processing time.

Attribute_name	open	low	high	Close	Adj. Close
Mean	236.26579607662143	233.53564440341583	238.78401947842121	236.30614491114673	236.30614491114673
Median	91.38	90.04	92.57	91.28	91.28
Count	4333	4333	4333	4333	4333
Std.	298.52234851945724	295.60889995039366	300.87679673104986	298.4190386884232	298.4190386884232
Max	1615.96	1590.89	1617.54	1598.39	1598.39
Min	5.91	5.51	6.1	5.97	5.97
25%	39.4	38.8	39.96	39.41	39.41
75%	306.3	302.91	310.23	306.45	306.45

Fig. 16 The statistical description of the used dataset

$$R^2 = 1 - \frac{\left(\sum_{i=1}^{n}\left(y_i - \widetilde{y}_i\right)^2\right)/n}{\left(\sum_{i=1}^{n}\left(\overline{y}_i - \widetilde{y}_i\right)^2\right)/n} \tag{17}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widetilde{y}_i\right)^2} \tag{18}$$

$$RMS = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widetilde{y}_i\right) \tag{19}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widetilde{y}_i\right)^2 \tag{20}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \widetilde{y}_i\right| \tag{21}$$

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \widetilde{y}_i}{y_i}\right| \tag{22}$$

$$\textbf{Processing time} = end\_time - start\_time \tag{23}$$

5.3 Results analysis

In this part, we provide results analysis of applying the machine and deep learning models used in the proposed framework and applied to the stock dataset through two experiments with different three cases to evaluate and validate the models' performance for stock price prediction.

In the first experiment, we discuss the stock price prediction process using normal ML models that are independent of neural networks such as SVR, linear Regression, Logistic Regression, DT, Random Forest, and KNN. In this experiment, we analyze our models against different cases for the test process not for the train where we forecast or model to predict the future values for three different cases. For more clearance, we can clear the forecasting process refers to what? generally, the means of descriptive and predictive analytics do price forecasting.

- **Descriptive analytics.** It relies on statistical methods that include data collection, data analysis, interpretation, and presentation of outputs. It allows for the transformation of raw observations into knowledge one can understand and share. In short, this analytics type helps to answer the question of *what happened?*

- **Predictive analytics.** Predictive analytics refers to analyzing current and historical data to forecast the probability of future events, outcomes, or values in the context of price predictions. Predictive analytics requires numerous statistical techniques, such as data mining (identification of patterns in data) and machine learning.

In our first experiment, we use predictive analytics for the forecasting process that deletes the data records determined by forecasting rate from the analysis process, then use it in the evaluation process where the comparison between the actual and forecasted predicted values is done.

- **Experiment 1:** In this experiment, we train and test ML Models for stock prediction on the predefined dataset with a period from 1/1/2000 to 30/12/2019. The prediction process forecasted out for multiple periods such as 30 days, 100 days, and 1000 days as follows. In each case, each ML Model is trained then tested.
- **Case 1:** 30 day

In this case, the results for 30-day forecast- out prediction are shown in Table 1. ML models are trained and tested for this case discussed evaluation metrics. The representation of ML prediction results against evaluation metrics is shown in Figs. 17 and 18.

Case 2: 100 day.

In this case, the results for 100 -day forecast- out prediction are shown in Table 2. ML models are trained and tested for this case with measured predefined metrics. The representation of ML models prediction results against evaluation metrics is shown in Figs. 19 and 20.

Case 3: 1000 day.

In this case, the results for 1000 -day forecast- out prediction are shown in Table 3. ML models are trained and tested for this case with measured predefined metrics. The representation of ML prediction results against evaluation metrics is shown in Fig.s 21 and 22.

- **Experiment 2:** In this experiment, we train and test DL Models for stock prediction on the predefined dataset with a period from 1/1/2000 to 30/12/2019. The prediction process is trained and tested for different values of epoch and batch size. DL model operation is based on two hyperparameters known as epoch and batch-size. The number of epoch reflects the times that the learning technique will work through the entire training dataset while the batch size reflects the number of data samples to work with

Table 1 ML models for Case 1

Learning Algorithm	Confidence/R2 Square %	RMS	RMSE	MSE	MAE	MAPE%
SVR	93.4	72.81	72.81	5301.63	21.18	12.78
Linear Regression	98.9	29.39	29.39	863.9	16.6	17.644
Logistic Regression	98.9	29.39	29.39	863.9	16.6	17.644
Decision Tree Regression	97.9	40.38	40.38	1630.59	28.216	61.39
Random Forest Regression	98.9	28.715	28.72	824.6	15.94	14.389
KNN	**99.1**	**26.377**	**26.38**	**695.75**	**14.979**	**14.389**

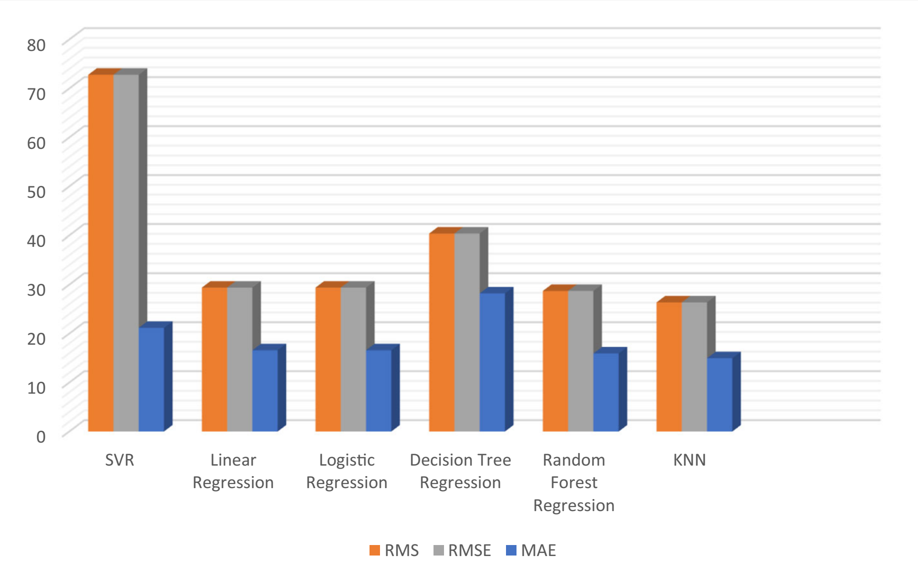


Fig. 17 Absolute ML Stock Prediction Results per 30 days

before parameter updating for the internal model parameters is done. There are three cases for different values of epoch with batch size such as:

- **Case 1:** epoch =100 and batch size = 1.

In this case, the results for DL models such as LSTM, Stacked LSTM, Bidirectional LSTM, and CNN are shown in Table 4. The representation of these results is shown in Fig. 23.

Case 2: epoch =10 and batch size = 10

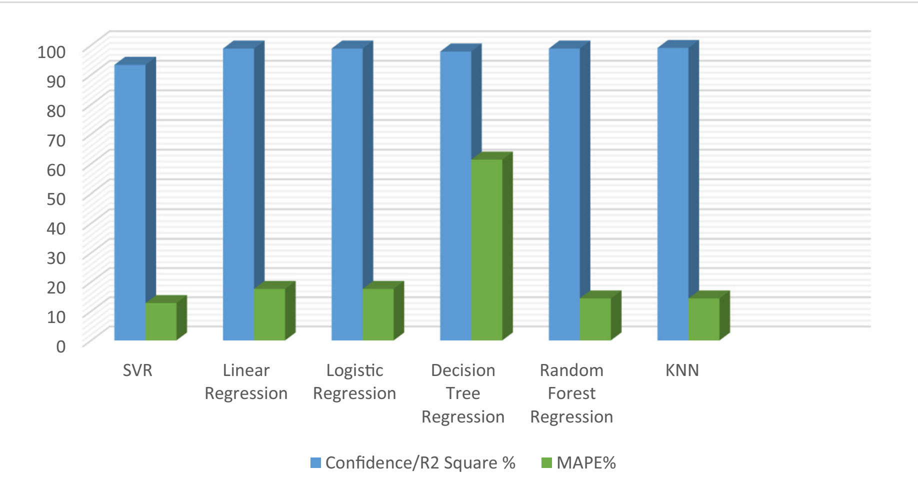


Fig. 18 Percentage ML Stock Prediction Results per 30 days

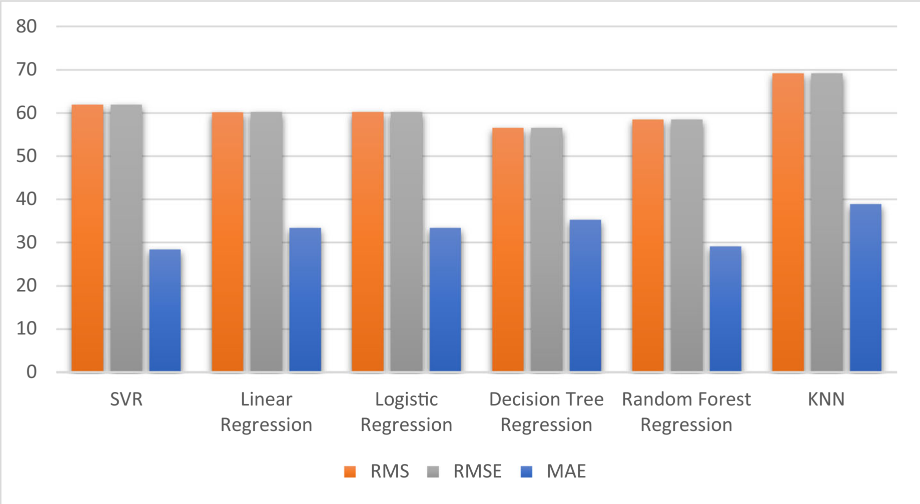


Fig. 19 Absolute ML Stock Prediction Results per 100 days

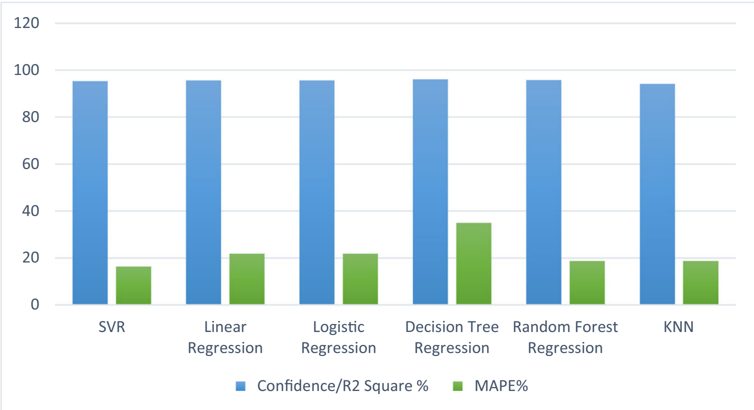


Fig. 20 Percentage ML Stock Prediction Results per 100 days

Table 3 ML models for Case3

Learning Algorithm	Confidence/R2 Square %	RMS	RMSE	MSE	MAE	MAPE%
SVR	94.9	72.28	72.28	5225.27	42.28	18.6
Linear Regression	93.8	79.428	79.43	6308.8	53.28	29.7
Logistic Regression	93.8	79.428	79.43	6308.8	53.28	29.7
Decision Tree Regression	94.96	71.906	71.91	5170.6	47.1	24.02
Random Forest Regression	94.9	71.867	71.87	5164.99	44.45	24
KNN	95.8	65.546	65.55	4296.75	44	24

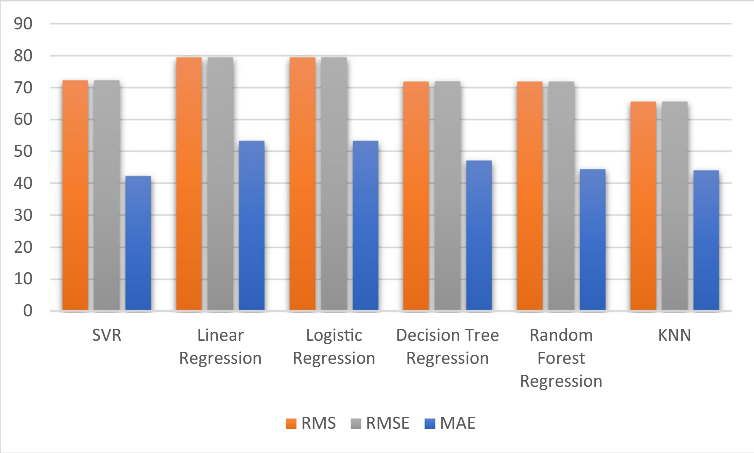


Fig. 21 Absolute ML Stock Prediction Results per 1000 days

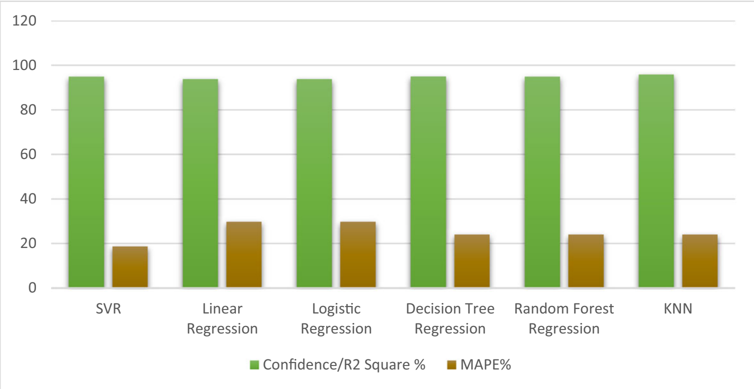


Fig. 22 Percentage ML Stock Prediction Results per 1000 days

Table 4 DL models Case 1 (100 epochs and 1 batch size)

Learning Algorithm	Confidence/R2 Square %	RMS	RMSE	MSE	MAE	MAPE%	Processing-time
LSTM	87.14	0.24	0.07	0.0048	0.04	52.37	1147.78
Stacked LSTM	61.77	0.22	0.12	0.013	0.07	41.67	3644.26
Bidirectional LSTM	99.3	0.25	0.02	0.0002	0.01	52.43	1644.29
CNN	99.8	0.288	0.28	0.08	0.219	42.67	1954.29

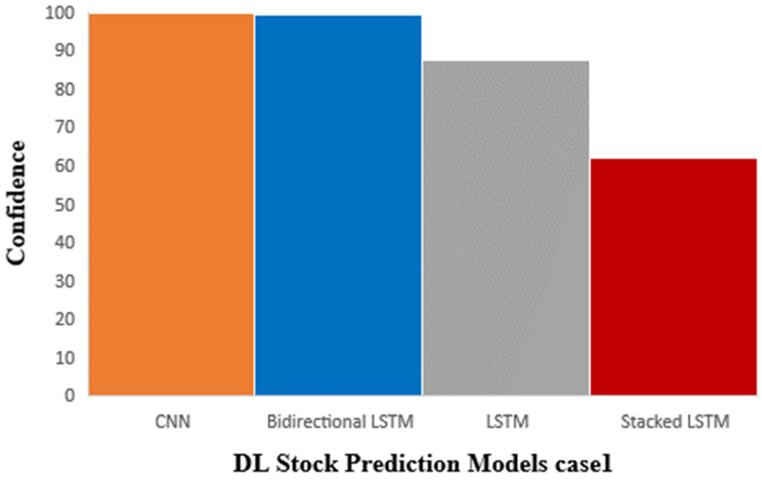


Fig. 23 Confidence ratio for DL Stock Prediction (Case 1)

In this case, the results for DL models such as LSTM, Stacked LSTM, Bidirectional LSTM, and CNN are shown in Table 5. These results are represented in Fig. 24.

Case 3: for epoch =1 and batch size = 100.

In this case, the results for DL models such as LSTM, Stacked LSTM, Bidirectional LSTM, and CNN are shown in Table 6. These results are represented in Fig. 25.

To evaluate the performance of the proposed work, first, we investigated different learning models on stock streaming data that can be requested by registering in the quandl site and can be requested by inserting the API key with requesting python code such as quandl.ApiConfig.api_key = "xzXxy7wnjW8DzqZnZmRz",

$$df = \text{quandl.get}(\text{“WIKI/AMZN”}, \text{start\_date} = \text{“2000−12−31”}, \text{end\_date} = \text{“2019−12−31”})$$

The quandl key allows the user from testing his code multiple times without facing problems while conducting with quandl finance data. Also, it can be tested without code but for a limited number of times.

Then, we conduct an experiment with ML models for three cases with different forecasting period to illustrate the accuracy variation. From the described results for ML models, the operation of ML models is varied according to the forecasting rate (the predicted samples for the future), when the forecast rate is increased, the accuracy is decreased. From the described results, the KNN model proved that it is the best model in both cases 1 and 3 as described in Tables 1 and 2. But, in case 2, the decision tree model outperformed all other models as

Table 5 DL models Case 2 (10 epochs and 10 batch size)

Learning Algorithm	Confidence/R2 Square %	RMS	RMSE	MSE	MAE	MAPE%	Processing-Time
LSTM	99.0	0.27	0.02	0.0003	0.013	56.18	116.83
Stacked LSTM	99.5	0.27	0.01	0.0001	0.008	54.8	403.57
Bidirectional LSTM	96.6	0.28	0.03	0.0011	0.02	52.43	206.75
CNN	**99.8**	**0.21**	**0.15**	**0.022**	**0.089**	**39.9**	**101.09**

Fig. 24 Confidence ratio for DL Stock Prediction (Case 2)

described in Table 3. So, to perform stock prediction with ML models, it is necessary to select a forecasting rate that provides high accuracy and balanced performance.

Lastly, we try out different DL models such as LSTM, Stacked-LSTM, Bidirectional - LSTM, and CNN for the same stock data. From experiment 2, we find that the operation of DL models is based on two hyperparameters such as epoch and batch size. Balancing between the number of epoch and the batch size results in the best performance for DL models that provides balancing between accuracy and processing time. The conclusion of these cases as follows:

- **Case 1:** In this case, we experimented with several epochs =100 and batch size =1, that work with internal training data for 100 iterations with 1 sample of training data. This results in the extraction of more features than required for the analysis process, as some of which are not required for the analysis process. In this case, the DL model takes more time in the extraction of unrequired features but provided the required accuracy. The overall performance cannot balance between the prediction accuracy and processing time; this is known as the overfitting problem. The results of this case were described in Table 4.
- **Case 2:** In this case, when an epoch equals 10 and batch size equals 10. In this case, the DL model trained for 10 iterations for the entire training data with 10 samples to work with before parameter updating. From Table 4, we find that this case provided the required accuracy with balanced processing time. This case proved that it is the best case that provides the best performance and avoids overfitting and the underfitting problem of DL models.
- **Case 3:** In this case, we experimented with DL models with one epoch and batch size equals 100, this case trained the stock data for 1 iteration while working with 100 samples

Table 6 DL models Case 3 (1 epoch and 100 batch size)

Learning Algorithm	Confidence/R2 Square %	RMS	RMSE	MSE	MAE	MAPE%	Processing Time(sec)
LSTM	−2.46	0.366	0.35	0.12	0.31	62.16	10.21
Stacked LSTM	−4.13	0.42	0.42	0.18	0.38	54.8	23.91
Bidirectional LSTM	−3.72	0.41	0.41	0.16	0.36	76.37	14.98
CNN	**−16.02**	**0.29**	**0.32**	**0.100**	**0.24**	**44.48**	**8.36**

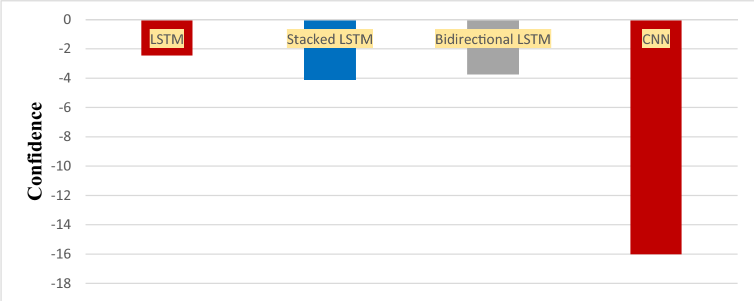


Fig. 25 Confidence ratio for DL Stock Prediction (Case 3)

before parameter updating. This case extracted fewer features than the required features for the stock prediction process. This experiment saved the processing time but provided the low accuracy for DL models. The resulted accuracy was in negative values that proved that DL models provided poor performance in this case. This case resulted in the underfitting problem of DL models. So, the selection of epoch and batch size is a key factor in stock prediction using DL models to avoid overfitting and underfitting problem. From the comparison between ML models and DL models, we find that DL models especially CNN and Bidirectional LSTM proved that they outperformed ML Models because these models learn the data deeply. As deep learning models operation based on deep extraction of features of the input and updating their parameters in a way that enhances accuracy. From the comparison between DL models, we find that the CNN DL model proved it is the best model for stock prediction in our case.

Finally, from the results, we can understand the effect of hyperparameters of NN on the performance. For our experiment, it is cleared that case2 (epoch = 10 & batch_size = 10) is the best choice for hyperparameters initialization as shown in Figs. 26, 27, and 28. This Figure shows the RMS variations for three cases with CNN. Future work can be in a modification of these two models to enhance performance. We can work on applying some preprocessing steps to put stock data in a form that can be more suitable for data analysis. The

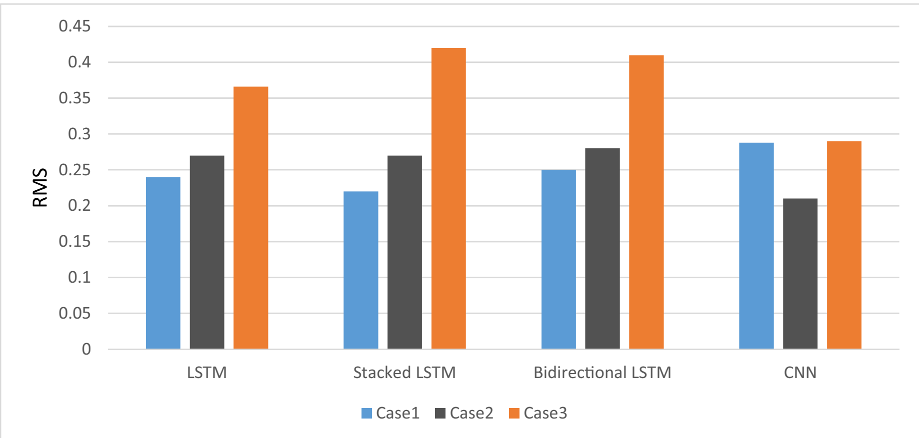


Fig. 26 RMS variation with epoch and batch size for DL models at different cases

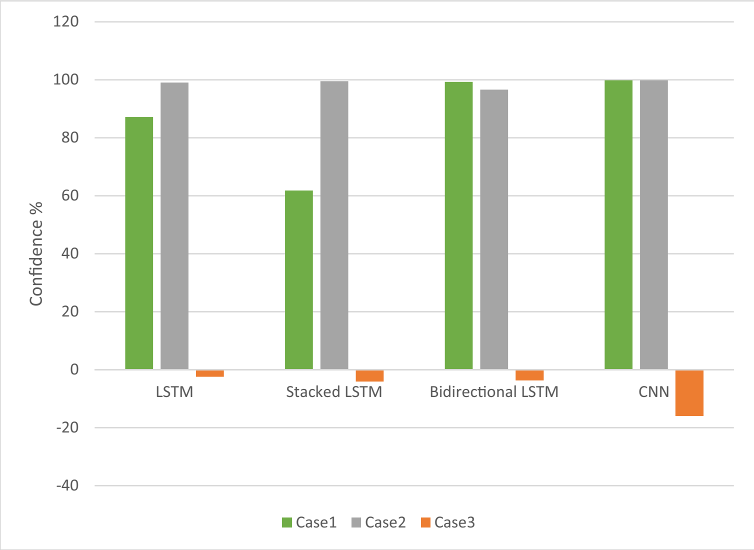


Fig. 27 Confidence variation with epoch and batch size for DL models at different cases

preprocessing can enhance the prediction performance. Also, in the future, we can apply text mining with data mining to provide a more accurate system for the stock price prediction process.

6 Conclusion and future scope

In recent years, stock price prediction has been recognized as an imperative and attractive research problem for many researchers around the world. Therefore, this paper provides an efficient framework for predicting the stock price using the machine and deep learning models such as LSTM, CNN, SVM, Linear, and Logistic Regression, K-Neighbors, Decision Tree,

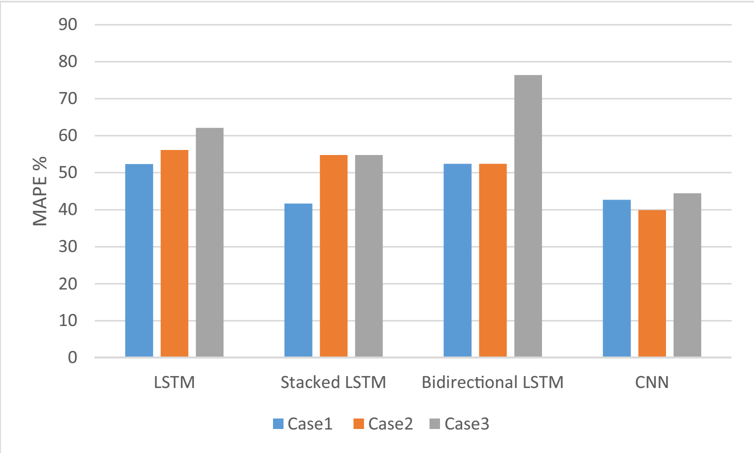


Fig. 28 MAPE variation with epoch and batch size for DL models at different cases

Random Forest, Stacked-LSTM, and Bidirectional-LSTM. The conducted results demonstrate that the CNN model outperforms the assessment algorithms consistently in different cases through various evaluation metrics. In the future, we plan to work in sentiment analysis and opinion mining for social multimedia and its relationship in the stock price market prediction. Likewise, we plan to suggest a new model based on hybridization between historical price and sentiment analysis of financial news and market movements.

References

1. Althelaya, K. A., El-Alfy, E. S. M., & Mohammed, S. (2018, April). Evaluation of bidirectional LSTM for short-and long-term stock market prediction. In 2018 9th international conference on information and communication systems (ICICS) (pp. 151–156). IEEE.
2. Ananda DB, Wibisono A (2014) C4. 5 decision tree implementation in Sistem Informasi zakat (Sizakat) to automatically determining the amount of zakat received by Mustahik. Jurnal Sistem Informasi 10(1):28–35
3. Awad M, Khanna R (2015) Efficient learning machines: theories, concepts, and applications for engineers and system designers (p 268). Springer Nature
4. Banyal S, Goel P, Grover D.(2020) Indian stock-market prediction using stacked LSTM AND multi-layered perceptron. Int J Innovative Technol Explor Eng (IJITEE) 9(3)
5. Basaldella M, Antolli E, Serra G, Tasso C (2018) Bidirectional lstm recurrent neural network for keyphrase extraction. In: *Italian Research Conference on Digital Libraries*. Springer, Cham, pp 180–187
6. Chen S, He H (2018) Stock prediction using convolutional neural network. In IOP Conference Series: Materials Science and Engineering (Vol. 435, no. 1, p 012026)
7. Cui Z, Ke R, Pu Z, Wang Y (2018) Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. arXiv preprint arXiv:1801.02143.
8. Daho MEH, Settouti N, Lazouni MEA, Chikh MEA (2014) Weighted vote for trees aggregation in random forest. In 2014 International Conference on Multimedia Computing and Systems (ICMCS) (pp. 438–443). IEEE
9. Deng L (2012) Three classes of deep learning architectures and their applications: a tutorial survey. APSIPA transactions on signal and information processing
10. Deng L, Yu D (2014) Deep learning: methods and applications. Foundations Trends Signal Process 7(3–4): 197–387
11. Derakhshan A, Beigy H (2019) Sentiment analysis on stock social media for stock price movement prediction. Eng Appl Artif Intell 85:569–578
12. Ding G, Qin L (2020) Study on the prediction of stock price based on the associated network model of LSTM. Int J Mach Learn Cybern 11(6):1307–1317
13. Eck D, Schmidhuber J (2002) A first look at music composition using lstm recurrent neural networks. Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale 103:48
14. Fan GF, Peng LL, Hong WC, Sun F (2016) Electric load forecasting by the SVR model with differential empirical mode decomposition and auto regression. Neurocomputing 173:958–970
15. Gao SE, Lin BS, Wang CM (2018) Share price trend prediction using CRNN with LSTM structure. In 2018 International Symposium on Computer, Consumer and Control (IS3C) (pp. 10-13). IEEE
16. Global FI (2018) International journal of synthetic emotions (IJSE). Emotion 9(2)
17. Graves A, Mohamed AR, Hinton G (2013) Speech recognition with deep recurrent neural networks. In 2013 IEEE international conference on acoustics, speech, and signal processing (pp. 6645-6649). IEEE
18. Han J, Pei J, Kamber M (2011) Data mining: concepts and techniques. Elsevier
19. Hegazy O, Soliman OS, Salam MA (2014) A machine learning model for stock market prediction. arXiv preprint arXiv:1402.7351
20. Hemdan EED, Manjaiah DH (2020). Digital investigation of cybercrimes based on big data analytics using deep learning. In Deep Learning and Neural Networks: Concepts, Methodologies, Tools, and Applications (pp. 615-632). IGI global
21. Hong WC (2011) Electric load forecasting by seasonal recurrent SVR (support vector regression) with chaotic artificial bee colony algorithm. Energy 36(9):5568–5578
22. Hong WC, Dong Y, Zhang WY, Chen LY, Panigrahi BK (2013) Cyclic electric load forecasting by seasonal SVR with chaotic genetic algorithm. Int J Electr Power Energy Syst 44(1):604–614
23. Hoseinzade E, Haratizadeh S (2019) CNNpred: CNN-based stock market prediction using a diverse set of variables. Expert Syst Appl 129:273–285

24. Hssina B, Merbouha A, Ezzikouri H, Erritali M (2014) A comparative study of decision tree ID3 and C4. 5. Int J Adv Comput Sci Appl 4(2):13–19
25. https://www.investopedia.com/terms/f/financial_sector.asp/ (Last Access 20/4/2020)
26. Hussain M, Bird JJ, Faria DR (2018) A study on cnn transfer learning for image classification. In: UK Workshop on Computational Intelligence. Springer, Cham, pp 191–202
27. Jiang S, Pang G, Wu M, Kuang L (2012) An improved K-nearest-neighbor algorithm for text categorization. Expert Syst Appl 39(1):1503–1509
28. Jin Z, Yang Y, Liu Y (2019) Stock closing price prediction based on sentiment analysis and LSTM. Neural Comput Appl:1-17
29. Jozefowicz R, Zaremba W, Sutskever I (2015) An empirical exploration of recurrent network architectures. In International conference on machine learning (pp. 2342-2350
30. Kim A, Myung J, Kim H (2020) Random forest ensemble using a weight-adjusted voting algorithm. J Korean Data Inform Sci Soc 31(2):427–438
31. Lamba A, Kumar D (2016) Survey on KNN and its variants. Int J Advanced Res Comput Commun Eng 5(5):430–435
32. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature *521*(7553):436–444
33. Li MW, Geng J, Hong WC, Zhang LD (2019) Periodogram estimation based on LSSVR-CCPSO compensation for forecasting ship motion. Nonlinear Dynamics 97(4):2579–2594
34. Liang X, Qu F, Yang Y, Cai H. (2015) An improved ID3 decision tree algorithm based on attribute weighted. In 2nd International Conference on Civil, Materials and Environmental Sciences. Atlantis Press
35. Nam K, Seong N (2019) Financial news-based stock movement prediction using causality analysis of influence in the Korean stock market. Decis Support Syst 117:100–112
36. Nandakumar R, Uttamraj K, Lokeswari YV (2018) Stock price prediction using long short term memory. Int Res J Eng Technol 5:3342–3348
37. Nguyen TT, Yoon S (2019) A novel approach to short-term stock Price movement prediction using transfer learning. Appl Sci 9(22):4745
38. O'Shea K, Nash R (2015) An introduction to convolutional neural networks. arXiv preprint arXiv: 1511.08458
39. Pal S, Ghosh S, Nag A (2018) Sentiment analysis in the light of LSTM recurrent neural networks. ICTACT J Soft Comput 9(3):33–39
40. Patel J, Patel M, Darji M (2018) Stock Price prediction using clustering and regression: a
41. Pejić Bach M, Krstić Ž, Seljan S, Turulja L (2019) Text mining for big data analysis in financial sector: a literature review. Sustainability 11(5):1277
42. Qiu J, Wang B, Zhou C (2020) Forecasting stock prices with long-short term memory neural network based on attention mechanism. PLoS One 15(1):e0227222
43. Randhawa K, Loo CK, Seera M, Lim CP, Nandi AK (2018) Credit card fraud detection using AdaBoost and majority voting. IEEE access 6:14277–14284
44. Ren R, Wu DD, Liu T (2018) Forecasting stock market movement direction using sentiment analysis and support vector machine. IEEE Syst J 13(1):760–770
45. Santos Pinheiro L, Dras M (2017) Stock market prediction with deep learning: a character-based neural language model for event-based trading. In Proceedings of the Australasian Language Technology Association Workshop 2017 (pp 6-15)
46. Shah D, Isah H, Zulkernine F (2018) Predicting the effects of news sentiments on the stock market. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 4705-4708). IEEE
47. Sharma H, Kumar S (2016) A survey on decision tree algorithms of classification in data mining. Int J Sci Res (IJSR) 5(4):2094–2097
48. Sherstinsky A (2020) Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. Physica D: Nonlinear Phenomena 404:132306
49. Smagulova K, James AP (2019) A survey on LSTM memristive neural network architectures and applications. Eur Phys J Special Topics 228(10):2313–2324
50. Sohangir S, Wang D, Pomeranets A, Khoshgoftaar TM (2018) Big data: deep learning for financial sentiment analysis. J Big Data 5(1):3
51. Sperandei S (2014) Understanding logistic regression analysis. Biochemia Medica: Biochemia Med 24(1): 12–18
52. Sudha VP, Kowsalya R (2015) A survey on deep learning techniques applications and challenges. Int J Advance Res Sci Eng (IJARSE) 4(03):311–317
53. Sundjaja, A. Sentiment analysis on twitter for predicting stock exchange movement. Pdf
54. Sutawinaya IP, Astawa INGA, Hariyanti NKD (2018) Comparison of Adaline and Multiple linear regression methods for rainfall forecasting. JPhCS 953(1):012046
55. Tipirisetty A (2018) Stock Price prediction using deep learning

56. Vinyals O, Toshev A, Bengio S, Erhan D (2015) Show and tell: a neural image caption generator. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3156-3164)
57. Wang Y (2017) A new concept using lstm neural networks for dynamic system identification. In 2017 American Control Conference (ACC) (pp. 5324-5329). IEEE
58. Xiaohu W, Lele W, Nianfeng L (2012) An application of decision tree based on id3. Phys Procedia 25: 1017–1021
59. Yamashita R, Nishio M, Do RKG, Togashi K (2018) Convolutional neural networks: an overview and application in radiology. Insights Imaging 9(4):611–629
60. Yu Z, Ramanarayanan V, Suendermann-Oeft D, Wang X, Zechner K, Chen L, ..., Qian Y (2015) Using bidirectional LSTM recurrent neural networks to learn high-level abstractions of sequential features for automated scoring of non-native spontaneous speech. In 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU) (pp. 338–345). IEEE
61. Zacharis NZ (2018) Classification and regression trees (CART) for predictive modeling in blended learning. IJ Intell Syst Appl 3:1–9
62. Zhang Z, Hong WC (2019) Electric load forecasting by complete ensemble empirical mode decomposition adaptive noise and support vector regression with quantum-based dragonfly algorithm. Nonlinear Dynamics 98(2):1107–1136
63. Zhang Z, Hong WC, Li J (2020) Electric load forecasting by hybrid self-recurrent support vector regression model with variational mode decomposition and improved cuckoo search algorithm. IEEE Access 8:14642–14658
64. Zhao R, Yan R, Wang J, Mao K (2017) Learning to monitor machine health with convolutional bi-directional LSTM networks. Sensors 17(2):273

Marwa Sharaf has received her B.Sc from the Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University, Egypt, in 2013. She received her M.Sc. From the Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University, Egypt, in 2018. Recently, she working towards her PhD degree at the Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University. She has several publications in national/international conferences and journals. Her research area of interest includes; Financial Data Analytics, Machine Learning, Deep Learning, Data Science, Database Systems, and Big Data Analytics.

Ezz El-Din Hemdan has received his B.Sc from the Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University, Egypt, in 2009. He received his M.Sc. From the Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University, Egypt, in 2013. He received his Ph.D. degree in the Department of Computer Science, Mangalore University, India in 2018. He has several publications in national/international conferences and journals. His research area of interest includes; Canacelable Biometric, Blockchain, Digital Twins, Image Processing, Virtualization, Cloud Computing, Internet of Things/Nano-Things, Cryptography, Data Hiding, Digital Forensics, Cloud Forensics, Big Data Forensics, Data Science and Big Data Analytics.

Ayman El-Sayed a Professor and Dean of Faculty of Electronic Engineering, Menoufia University, Egypt. He received a B.Sc. degree in computer science and engineering in 1994, a master's degree in computer networks in 2000 from the Menoufia University, Egypt, and a Ph.D. degree in computer networks in 2004 from (Institute National De Poly technique De Grenoble) INPG, France. He is specialized in soft computing, algorithms, data structure, and cloud computing and big data analysis. Besides, his interests include multicast routing protocols, application-level multicast techniques, multicast on both mobile network and mobile IP, and image processing techniques. In addition, there are other interesting topics such as bioinformatics, Biocomputing, and bio computer. He is an approved supervisor for M.Sc. and Ph.D. programs in various University. He has completed various project in government and private organization. He has published more than 130 research papers in international Journals and two books about OSPF protocol and multicast protocols. Currently, he is serving as an editorial board member in various international Journals and conferences. He is a senior member of the IEEE.

Nirmeen A. El-Bahnasawy received her B.S. in Electronic Engineering in 1998 and M.Sc. and Ph.D. degrees in Computer Science and Engineering from Menoufia University in 2003 and 2013, respectively. Currently, she has been appointed as an Associate Professor at Menoufia University in 2019. She has publishing of 42 different research papers in highly ranked scientific peer-reviewed journals. Her research interests include distributed computing, grid computing, IoT, Artificial Intelligence, Fog Computing, and Cloud computing. She has deep experience in dealing with electronics H/W kits, different software tools, and different programming languages. She did and supervised different H/W and S/W implementations projects. In terms of research, nine M.Sc. candidate students and five Ph.D. candidate students are working under his supervision in different research topics, Medical Image Processing, Speech Processing, Security Algorithms, Software Defined Networks, Internet of Things Applications, Medical Diagnoses Applications, Artificial Intelligence, Data compression and Parallel Processing