International Journal of Circuit, Computing and Networking

E-ISSN: 2707-5931
P-ISSN: 2707-5923
IJCCN 2023; 4(1): 20-24
Received: 22-01-2023
Accepted: 01-04-2023

Aditya Singh Rajpurohit
Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, Maharashtra, India

Pradnya Sangitbabu Gaikwad
Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, Maharashtra, India

Harshada Mhaske
Department of Computer Engineering, PCCOE, Pune, Maharashtra, India

Shravani Prakash Ahirrao
Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, Maharashtra, India

Nutan Bhairu Dhamale
Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, Maharashtra, India

Corresponding Author:
Aditya Singh Rajpurohit
Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, Maharashtra, India

Survey of different techniques used to predict the Stock price

Aditya Singh Rajpurohit, Pradnya Sangitbabu Gaikwad, Harshada Mhaske, Shravani Prakash Ahirrao and Nutan Bhairu Dhamale

DOI: https://doi.org/10.33545/27075923.2023.v4.i1a.55

Abstract
A stock market is a place where we can purchase the stocks of various companies (Part of the company), which makes it volatile, and predicting it becomes a tedious task. So we need various algorithms and methodologies to predict the stock prices. We cannot depend on one type of algorithm because each algorithm has its own pros and cons and also it depends on the style of the trader on how he trades stocks. This paper will deal with different aspects like quantitative aspect- LSTM, RNN, ARIMA, and qualitative with sentiment analysis for predicting the stock prices, in an efficient manner.

Keywords: Stock prediction, machine learning, LSTM, RNN, ARIMA, sentiment analysis, NLP

1. Introduction
Stock markets are highly capricious in nature. We have deduced a company-oriented platform to ease the process of stock prediction. We have involved four major algorithms to procure the minimum errored stock prices of a company. We have studied the below models-

1. **LSTM (Long Short-Term Memory) Model:** Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture. LSTM networks are well-suited to classifying, making predictions and processing based on time series data.
2. **ARIMA (Autoregressive Integrated Moving Average) Model:** This is one of the easiest and effective machine learning algorithms to perform time series forecasting. This is the combination of Auto Regression and Moving average.
3. **RNN (Recurrent Neural Network) Model:** RNN model is a type of neural network which is simple to design, less computation required, and is used mostly in Non-Sequential data (NLP), Named Entity Recognition (NER).
4. **Sentimental Analysis:** It is said that the "Stock market is sentiment driven", which means that the opinions and views of the people collectively play a very important role in determining any company's position in the stock market, and using sentiment analysis we can analyze and determine that. This analysis can be done on various social media platforms like Facebook, Twitter, Stock Twits, etc. For analyzing we can use Natural language processing and different machine learning classification models.

2. RNN Model
1. It is a simple type of algorithm which is used on sequential data. Derived from feed forward networks, RNNs exhibit similar behavior to how human brains Function. Recurrent Networks Produce predictive results in sequential data that other Algorithms can't.
2. The RNN gives the best accuracy when it is used to predict the short-term stock price. In short-term stock prices, there is pattern formation like the morning star, shooting star, bullish Engulfing pattern, Bearish Engulfing pattern, etc. Using these patterns we predict the prices of stock.
3. RNN is efficient when the prediction is not to be made in a year. RNN is not suitable for long-term stock prices because in the long term various factors are to be considered like company's production line, cagr value, the overall performance (annual report).
4. The accuracy which is observed is most for the 15 days.

2.1 Algorithm

1. Start
2. The input of start date, end dates are taken so that the raw data between those dates can be loaded.
3. The company's Ticker name is taken so to know which company's stock data is needed to be loaded.
4. The pandas_datareader is now used to load the data through the yahoo finance API.
5. The data is stored in a variable and the key values which are present in the data are Date, High, Low, Volume, Open, Close.
6. From data read only the opening price of the stock
7. Feed the data to the input layer of the RNN.
8. Output from the output layer of the RNN will be checked through mean squared error.
9. Stop.

3. LSTM Model

Long Short-Term memory is one of the most successful RNNs architectures. LSTM introduces the memory cell, a unit of computation that replaces traditional artificial neurons in the hidden layer of the network. With these memory cells, networks are able to effectively associate memories and input remotely in time, hence suited to grasp the structure of data dynamically over time with high prediction capacity. Long Short Term Memory stores data in memory cells for a long time which will be very useful for stock prediction .The main purpose of predicting a series of time series models is to construct future value simulation models given their previous values.

3.1 LSTM: An overview

A special type of RNN, which can learn long-term dependence, is called Long-Short Term Memory (LSTM). In Long Short Term Memory, there are three gates: input gate, forget gate, and output gate. These gates determine whether new input (input gate) should be allowed, data deleted because it is not important (forget gate), or allow it to affect output at the current timeline (output gate).

Forget gate: The forget gateway determines when certain parts of the cell will be inserted with information that is more recent. It subtracts almost 1 in parts of the cell state to be kept, and zero in values to be ignored.

1. **Input gate:** Based on the input, this network category reads the conditions under which any information should be updated (or stored) in the state cell.
2. **Output gate:** Depending on the input mode and the cell, this component determines which information is forwarded to the next location in the network.

The cell remembers the value over time intervals and three gates regulate the flow of information of the cell in the cell and out of the cell. The cell of the LSTM model keeps track of dependencies between elements in the input sequence.

3.2 Algorithm of LSTM Model

Step 1: Raw Data: In this stage, the historical stock data is collected from Yahoo finance and this historical data is used for the prediction of future stock prices.

Step 2: Data Preprocessing: The pre-processing stage involves

a. **Data discretization:** Part of data reduction but with particular importance, especially for numerical data
b. **Data transformation:** Normalization.
c. **Data cleaning:** Fill in missing values.
d. **Data integration:** Integration of data files.

After the dataset is transformed into a clean dataset, the dataset is divided into training and testing sets so as to evaluate. Here, the training values are taken as the more recent values. Testing data is kept as 5-10 percent of the total dataset.

Step 3: Feature Extraction: In this layer, only the features which are to be fed to the neural network are chosen. We will choose the feature from Date, open, high, low, close, and volume.

Step 4: Training Neural Network: In this stage, the data is fed to the neural network and trained for prediction assigning random biases and weights. Our LSTM model is composed of a sequential input layer followed by 2 LSTM layers and a dense layer with ReLU activation and then finally a dense output layer with a linear activation function.

Step 5: Output Generation: In this layer, the output value generated by the output layer of the RNN is compared with the target value. The error or the difference between the target and the obtained output value is minimized by using a back propagation algorithm that adjusts the weights and the biases of the network.

4. ARIMA model

In statistics, particularly in time series analysis, an autoregressive integrated moving average (ARIMA) model is a generalization of an autoregressive moving average (ARMA) model. Both of these models are fitted to time series data either to better understand the data or to predict future points in the series. Time series is a sequence where we require a metric over an interval. Forecasting refers to the future values which these series shall correspond to. AR-Auto-regression is a time series model that uses observations from previous time steps as input to the regression equation to predict the value at the next time step. MA stands for moving average which is also called rolling mean. We calculate the simple average in a particular time frame and divide it with the total number of time frames taken.

In terms of y, the general forecasting equation is:

$$\hat{y}_t = \mu + \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} - \theta_1 e_{t-1} - \ldots - \theta_q e_{t-q}$$

Non-seasonal ARIMA models are generally denoted ARIMA (p, d, q) where parameters p, d, and q are non-negative integers, p is the order (number of time lags) of the autoregressive model, d is the degree of differencing (the number of times the data have had past values subtracted), and q is the order of the moving-average model. Seasonal ARIMA models are usually denoted ARIMA (p, d, q) (P, D, Q) m, where m refers to the number of periods in each season, and the uppercase P,D,Q refer to the autoregressive, differencing, and moving average terms for the seasonal part of the ARIMA model.

ARIMA models are applied when data show evidence of non-stationarity in the sense of mean, where an initial differencing step can be applied one or more times to eliminate the non-stationarity of the mean function. The ARMA model, according to the Wold's decomposition theorem, is theoretically sufficient to describe a regular wide-sense stationary time series. In Auto ARIMA, the

model itself will generate the optimal values which would be suitable for the data set to provide better forecasting. It can be used for any non-seasonal series of numbers that exhibits patterns and is not a series of random events.

4.1 Algorithm of ARIMA model

Step 1: Installing the 'pmdarima' package.
Step 2: Preparing the data.
Step 3: Understanding the pattern

To understand the pattern of the data, we plot a graph using the .plot () method.

Step 4: Test for Stationarity
We use the 'Augmented Dickey-Fuller Test' to check whether the data is stationary or not, which is available in the 'pmdarima' package. It returns false if the data is not stationary. If the data is not stationary, we would need to use the "Integrated (I)" concept, denoted by value 'd' in time series to make the data stationary while building the Auto ARIMA model.

Step 5: Train and Test split
We split into train and test datasets to build the model on the training dataset and forecast using the test dataset. We plot Train and Test datasets.

Step 6: Building Auto ARIMA model Step 7: Forecasting on the test data. We use the trained model which was built in the earlier step to forecast on the test data. Then we plot the trained data, Test Data, and Predicted values data.

Step 8: Displaying the plotted data.

4.2 Dataset for RNN, LSTM and ARIMA models

The dataset is taken from the yahoo finances API, by using pandas_datareader which is used as a data frame to read the data from the various internet data sources like Yahoo Finance, Google Finance, St. Louis FED (FRED), Kenneth French's data library, World Bank, Google Analytics. The data which is loaded from yahoo finance contains Date, High, Low, Open, Close, Volume, ADJ close.

4.3 Steps involved in data preprocessing

1. Importing the required libraries
2. Getting the data from yahoo finance or the dataset can also be taken from the kaggle.
3. Doing the analysis of the data
4. Checking the shape and also looking for the outliers
5. Plotting the outliers
6. Scaling the data using the min max scalar.
7. Building the model
8. Observing the results with different epochs and batch sizes

Date	High	Low	Open	Close	Volume	Adj Close
2013-11-01	37.000000	32.099998	35.000000	36.200001	14667600.0	33.839130
2013-11-04	36.799999	34.689999	36.799999	35.349998	1586600.0	33.044563
2013-11-05	35.500000	34.820000	35.110001	35.349998	337700.0	33.044563
2013-11-06	36.490002	35.610001	35.610001	35.900002	549600.0	33.558697
2013-11-07	36.799999	34.119999	36.799999	35.310001	883700.0	33.007175

Fig 1: Historic prices Dataset

5. Sentiment Analysis

It has become a very common thing that before buying something online or even installing an application on our mobile, we tend to go through the reviews. These reviews help us to get a basic idea of how most of the people responded to the product and also contribute to our decision of getting it or not.
Sometimes the reviews tend to build an opinion about it even before we have used or tried it. Similarly, as a preliminary thing before investing, people search for the market's opinion about the company. If there are any negative opinions or rumors about a company then investors hesitate to invest in that particular company. So for predicting the stock prices this "sentiment" factor plays a major role. Sentiment Analysis can be incorporated to study these sentiments and analyze the overall opinion of people about different companies in the stock market.
Various online platforms like Facebook, Twitter, etc. are there wherein people express their opinions and thoughts. For example To manually analyze reviews and feedback given by people and classify them based on saying positive and negative feedback is not an easy task as there may be thousands and lakhs of people giving those reviews. So to simplify this task, we can make use of natural language processing to analyze the text along with machine learning models to classify them.

5.1 Libraries for natural language processing

NLTK is a python library that has built-in functions for natural language processing. Various other libraries that can be used are Gensim, Stanford core NLP, Spacy, TextBlob, and pattern.

5.2 Dataset

The dataset can be taken from the Twitter API. But for using the API it is necessary to have a Twitter account. The other way to get the dataset is through twint, which is a Twitter scraping tool and it doesn't make use of the Twitter API. The dataset can be downloaded in the CSV file format. Various columns like the information of Twitter id, user id, created at the time, conversation id, username, the tweet, etc. will be in the dataset. The tweets from the StockTwits platform can also be taken. Stocktwits is a platform where people who are especially interested in the stock market use and share their opinions, thoughts, and ideas.

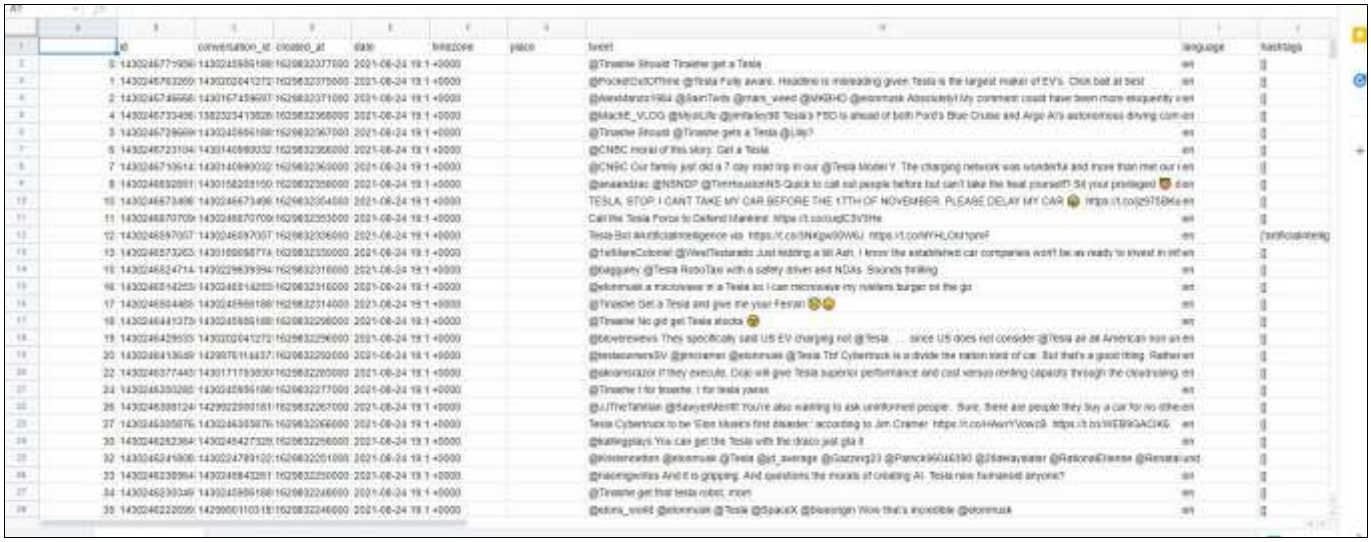

Fig 2: Dataset for sentiment Analysis

5.3 Data Preprocessing

Data preprocessing is a crucial step that needs to be performed before the data is given as input. Natural Language Toolkit can be used to perform the preprocessing. The following are some of the important tasks that are to be performed:

1. As the machine treats the lower and uppercase words differently, the complete text is converted to lowercase.
2. Stop words or words that are not very significant are removed. This can be done by first downloading all the stop words and then only keeping those words that are not present in the downloaded stop words list.
3. All the special characters, hashtags, extra spaces, and blank spaces are removed.
4. Tokenization or breaking the text into small independent parts is done, this can be breaking a
5. Checking if there are any emojis and then replacing them.
6. Removal of the hyperlinks as the dataset can contain them.
7. Removal of the '@' mentions
8. Removal of the digits as they do not contribute towards the analysis of the sentiment analysis

1. Bag of words

In this method, a vocabulary is built from the given input of unique words. Then based on the occurrence of the word, the value 1(if present) or 0 (not present) is placed in the matrix. With a larger input, the vocabulary size will increase and this method does not take into account the sequence of occurrence or the grammar.

For example, training data consists of the following two statements," Stock Prediction using Sentiment Analysis and Machine Learning" and "Stock Prediction using LSTM".

Vocabulary :['Stock',' Prediction',' using',' Sentiment',' Analysis',' and',' Machine',' learning',' LSTM']

For statement 1 vector: [1,1,1,1,1,1,1,1,0]

2. Tokenization

Breaking the paragraph into sentences or sentences into words as per our requirement. There are various ways in which this can be performed. Mostly it can be done by carrying out the tokenization by considering the white spaces. There's also rule based tokenization. Example," Data Preprocessing is an important step in building any model".

The tokens produced-['Data', 'Preprocessing',' is', 'an', 'important', 'step', 'in', 'building', 'any',' model']

3. Stemming

Finding the root for a particular word is done. But stemming is not very efficient as sometimes it may produce meaningless words so we can use lemmatization. Stemming is an important part of the NLP processing. This can be explained with an example that, consider a document with the words 'play',' played' and 'plays'. All this words mean the same but the machine learning model will evaluate it as different words. For this the best way to tackle it is by converting all the words to their root form. This will improve the performance of the model.

4. Lemmatization

An advanced form of stemming i.e. lemmatization is done so that meaningful root words are produced. It is preferred over the stemming. The reason for this is that most of the times stemming works by just removing the suffixes from the words without paying much attention to what the word will turn out to be, even when the words produced do not make any sense. Therefore lemmatization is preferred and used.

5.4 Vectorizing tokens

The input to be given to the model cannot be in the text format and needs to be converted to the numeric form. For this purpose following methods can be used:

For statement 2 vector: [1,1,1,0,0,0,0,0,1]

The value is '0' if the word is not present and '1' if present.

1. TF-IDF (Term Frequency-Inverse Document Frequency)

In this method first the frequency of each word is calculated in the document it is present in and then its frequency is calculated in all the documents present. Then the product of both the frequencies gives the final vector value for the word.

2. N-Gram model

This model works just like the bag of words, but the major difference between the two is that it takes into account the sequence. There can be unigrams, bigrams, trigrams, etc.: based on the number of words taken into consideration at a time. Example: Stock prediction using sentiment analysis

Unigram: [‘Stock’,’ prediction’,’ using’,’ sentiment’,’ analyses]

Bigram: [‘Stock prediction’,’ prediction using’,’ using sentiment’,’ sentiment analysis’]

Trigram: [‘Stock prediction using’,’ prediction using sentiment’,’ using sentiment analysis’]

5.5 Classification Algorithm Model

The dataset can be divided into training and testing datasets. The classification model is selected and then the results are evaluated. Different models that can be used are Naive Bayes, SVM, Decision tree, Random Forest, and Logistic Regression.

5.6 Challenges

With sentiment analysis, it is very difficult to process sarcasm or contradictions made in the same statement. For example,” The laptop design is good but the battery life is worst”, for processing this statement it would be difficult as both ‘good’ and ‘worst’ words with positive and negative sentiment are used in the same statement. Processing statements with lexical ambiguity or a statement with more than one meaning is also difficult. For example,’ I like apple’, so in this statement, the word ‘apple’ can refer to the fruit apple or the company apple.

6. Conclusion

Our paper facilitates a comparative and well-formulated study of different algorithms to efficiently predict the stock prices of a company. The precise advance prediction of the stock prices would enable the companies to adapt to the changing market scenario in an efficient way. This would aid the companies to amend and adopt necessary policies which would stabilize and boost their stock prices. The system would predict the stock prices with minimum or zero errors.

7. References

1. Mohan S, Mullapudi S, Sammeta S, Vijayvergia P, Anastasiu DC. Stock Price Prediction Using News Sentiment Analysis, IEEE-Institute of Electrical and Electronics Engineers; c2019.
2. Gupta R, Chen M. Sentiment Analysis for Stock Price Prediction, IEEE-Institute of Electrical and Electronics Engineers; c2020.
3. Xu Y, Keselj V. Stock Prediction using Deep Learning and Sentiment Analysis, IEEE-Institute of Electrical and Electronics Engineers; c2019.
4. Wang Z, Ho SB, Lin Z. Stock Market Prediction Analysis by Incorporating Social and News Opinion and Sentiment, IEEE-Institute of Electrical and Electronics Engineers; c2018.
5. Makrehchi M, Shah S, Liao W. Stock Prediction Using Event-Based Sentiment Analysis, IEEE-Institute of Electrical and Electronics Engineers; c2013.
6. Khandelwal S, Mohanty D. Stock Price Prediction Using Arima Model, International Journal of Marketing & Human Resource Research; c2021.
7. Khan S, Alghulaiakh H. ARIMA Model for Accurate Time Series Stocks Forecasting, International Journal of Advanced Computer Science and Applications; c2020.
8. Adebiyi AA, Adewumi AO, Ayo CK. International Conference on Computer Modelling and Simulation, Stock Price Prediction Using the ARIMA Model; c2014.
9. Dhyani B, Kumar M, Verma P, Jain A. Stock Market Forecasting Technique using Arima Model, International Journal of Recent Technology and Engineering (IJRTE); c2020.
10. Ahire P, Lad HK, Parekh S, Kabrawala S. LSTM based stock price prediction, international journal of creative research thoughts (IJCRT).
11. Roondiwala M, Patel H, Varma S. Predicting stock price using LSTM, International Journal of science and research.