<u>A Project Review-2 Report</u>

On

STOCK MARKET PREDICITION USING LONG SHORT-TERM MEMORY

Submitted in partial fulfilment of the requirement for the award ofthe degreeof

BTECH CSE

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of Mr.

E.GOUTHAM

Submitted By

NAMAN GUPTA

(19SCSE1010140)

NAMAN GUPTA

(19SCSE1120019)

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING GALGOTIAS UNIVERSITY, GREATER NOIDA

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"STOCK MARKET PREDICTION USING LSTM"** in partial fulfillment of the requirements for the award of the B.TECH ~~su~~bmitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of 07/21 to 12/21, under the supervision of E.Goutham Assistant professor, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

NAMAN GUPTA 19SCSCE1120019

NAMAN GUPTA 19SCSCE1O1O140

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

E.Goutham

Assistant professor

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of NAMAN GUPTA – 19SCSCE1120019 & NAMAN GUPTA 19SCSCE1010140 has been held on___________________and his/her work is recommended for the award of B.TECH

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date:

Place: Greater Noida

Abstract

In the finance world stock trading is one of the most important activities. Stock market prediction is an act of trying to determine the future value of a stock other financial instrument traded on a financial exchange. This paper explains the prediction of a stock using Machine Learning. The technical and fundamental or the time series analysis is used by the most of the stockbrokers while making the stock predictions.

We employ both random forests and LSTM networksas training methodologies to analyze their effectiveness in forecasting out-of-sample directional movements of constituent stockst. Our empirical results show that the multi-feature setting provides a daily return, prior to transaction costs We introduce a multi-feature setting consisting not only of the returns with respect to the
closing prices

The programming language is used to predict the stock market using machine learning is Python. In this paper we propose a Machine Learning (ML) approach that will be trained from the available stocks data and gain intelligence and then uses the acquired knowledge for an accurate prediction.
Multi layer Perceptrons(MP)
Artificial Neural Networks
Long Short-Term Memory Model(LSTM)

Based on historical prices available as data, these models are used to forecast stock prices.This system will provide accurate outcomes in comparison to currently available stock price predictor algorithms. The network is trained and evaluated with various sizes of input data to urge the graphical outcomes

The stock market is a human image emotions. Pure number crunching and analyzing their base limitations; a possible extension of this stock forecast the program can expand it with news feeds analysis from social media platforms such as Twitter, where emotions are measured in articles. This feeling analysis can be linked to LSTM for better training weights and improve accuracy

Table of Contents

List of Table

List of Figures

Chapter 1: Introduction

- ## Introduction

1.1

Stock markets tend to be extremely volatile and also create huge amounts of data on each trading day. A place where stocks or shares of listed companies are traded is stock market. It consists of two components,

1. Primary

2. Secondary.

Primary market can be defined as a place where issues are introduced newly through IPOs abbreviated as Initial public offerings, whereas in secondary market, investors will trade on derivatives / securities that are owned by them.

1.2

Stock markets follow non-linear time series containing high fluctuating data. Because of its random nature, prediction involves risks compared to other sectors.

the future stock prices of State Bank of India (SBIN) are predicted using the LSTM Recurrent Neural Network. Our task is to predict stock prices for a few days, which is a time series problem. The LSTM model is very popular in time-series forecasting, and this is the reason why this model is chosen in this task. The historical prices of SBIN are collected automatically using the nsepy library of python.

This data set contains 1483 observations with 12 attributes. After preprocessing, only dates and OHLC (Open, High, Low, Close) columns, a total of 5 columns, are taken as these columns have main significance in the dataset.

1.2.1 Tools and Technologies Used

The LSTM model is trained on this entire dataset, The stock prices for this new duration will be predicted by the already trained LSTM model, and the predicted prices will be plotted against the original prices to visualise the models accuracy. The experimental evaluation is based on the historical data set of National Stock Exchange (NSE). The proposed approach aims to provide models like Stacked LSTM which perform better than its contemporaries which have been achieved to a certain extent. This can be verified by the results embedded in the paper . The future research can be focused on adding more variables to the model by fetching live data from the internet as well as improving model by selecting more critical factors by ensemble learning.

Tools Used:

i) Anaconda Navigator: Anaconda Navigator is a graphical user interface which includes Anaconda distribution that helps in the launching of application and in the management of conda packages, environment and channels without using the command line system. It is a package manager, an environment manager, a Python/R data science distribution, and a collection of over 7,500+ open-source packages. It is available for operating systems like Windows, macOS and Linux.

ii) Jupyter Notebook: The Jupyter Notebook is an open -source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter. It can be used for data science, statistical modeling, machine learning and many such things

Technologies Used:

i) Machine Learning: Machine Learning System automatically learn program from data. It used in web Search, spam filter, recommendation system, and placement, credit scoring, recommendation of stock trading, fraud detection etc. There are lot of types of Machine learning but the one that used in our system is

ii) Python: It is a high-level general-purpose programming language. It is dynamically typed and also is garbage-collected. Multiple Programming and object-oriented and functional programming are one of the main features of python programming. Flexibility in Python allows it to be a great option for Machine Learning. Developing Scripts in Python is much easier because of all the standard libraires that are present in Python

Chapter 2: Literature Review

Many researchers has contributed in this field in this paper Our first book survey was a test online learning algorithms are common and see if they can be converted to our condition of use i.e., operating at a real-time stock price data. This includes Online AUC Maximization [8], Online Transfer Reading [9], and Online Feature Selection [1]. However, as we were unable to find any possible alignment of these stock price predictions, we then decided to look existing systems [2], analyzing major barriers to the same, and see if we can improve in them. The interaction between stock data (in the form of dynamic, long-term dependence between stocks prices) as a key issue we wished to resolve. In short searching for common solutions to the above problem has led us to RNN's [4] and LSTM [3]. After deciding to use LSTM neural network to make stock forecasts, contact a number of papers to study the concept of gradient down time and its various forms. We concluded our book survey by saying to look at how gradient shrinkage can be used for tuning the weight of the LSTM network [5] and how this process can be well done

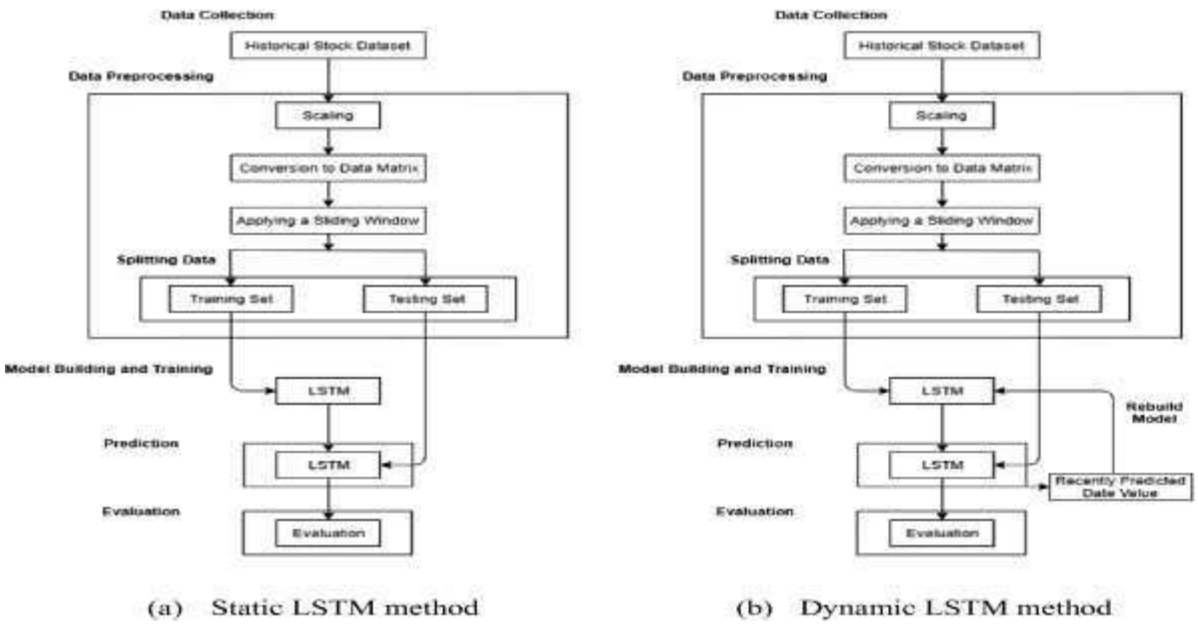
Traditional methods of stock market and stock market analysis Price forecasts include a basic, visual analysis in the previous operation of the stock and the general reliability of the company itself, and the statistical analysis, which is its only relating to abbreviations of numbers and patterns of identification on the price variation of the stock. The latter is often achieved with the help of Genetic Algorithms (GA) or Artificial Neural Networks (ANN's), but these failed to capture the link between stock prices in the short term dependence. Another major problem with using simple ANNs stock forecasting is an explosion / a perishable gradient [4], in which the masses of a large network either very large or very small (respectively), significantly reduce their merger to a fair value. This is usually due to two factors: the weights are initiated randomly, with weights close to the end of the network and they often change more than they did in the beginning.

An alternative approach to stock market analysis is reduction apply the input data dimension [2] and features selection algorithm for putting the core set of a function in the candidate list (Example: GDP, oil price, inflation rate, etc.) .

This will impact on overall stock prices and exchange rates market [10]. However, this method does not cover the full history of, so it does not consider long-term trading strategies as it fails to take the entire history of trends into account; furthermore, there is no provision for outlier detection.

Chapter 3: Project Design

3.1 Proposed System:



Project Prerequisites
The experiment has been conducted by a decent powered Intel® core ™ i7-7500U CPU @ 2.70GHz (4 CPU's) with a memory size of 8GB.. Python has been used as the development language with Development environment being provided by Windows. Anaconda Tools have been used for providing integrated development environment.

3.2 Database:

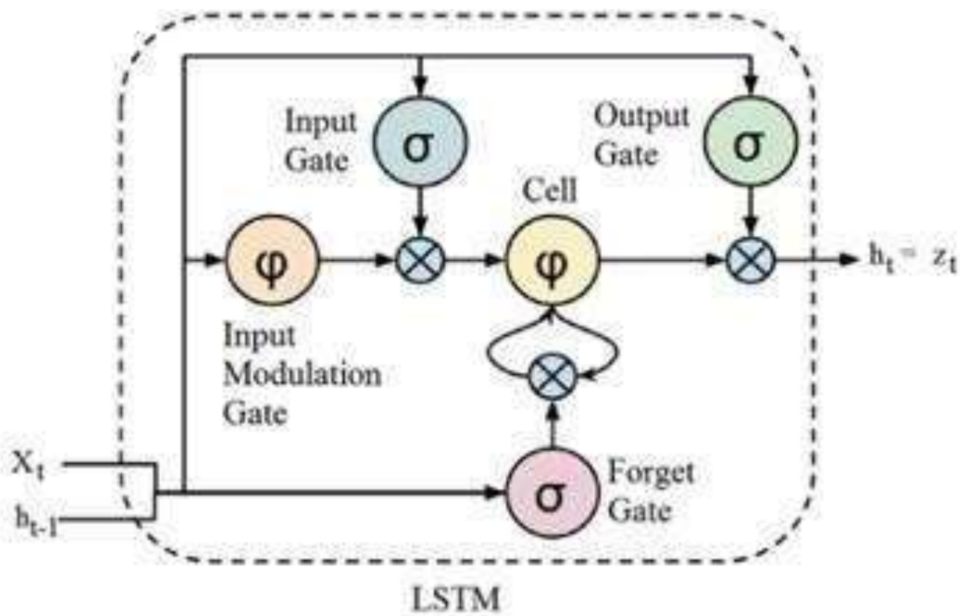
https://github.com/Namangupta021/-LGMVIP-Web/blob/main/NSE-TATAGLOBAL.csv

METHODOLOGY
We propose an online learning algorithm for predicting the
end-of-day price of a given stock with the help of Long Short Term Memory (LSTM), a type of Recurrent Neural Network(RNN).

Once the LSTM model is fit to the training data, it can be used to predict the end-of-day stock price of an arbitrary stock.
This prediction can be performed in two ways:
1. Static – a simple, less accurate method where the model is fit on all the training data. Each new time

2. Dynamic – a complex, more accurate approach where the model is refit for each time step of the test data as new observations are made available.

LSTM Memory Cell
An LSTM memory cell, as depicted in Figure 2, has the following three components, or gates



1. Forget gate: the forget gate decides when specific portions of the cell state are to be replaced with more recent information. It outputs values close to 1 for parts of the cell state that should be retained, and zero for values that should be neglected.

2. Input gate : based on the input (i.e., previous output o(t-1), input x(t), and previous cell state c(t-1)), this section of the network learns the conditions under which any information should be stored (or updated) in the cell state

3. Output gate: depending on the input and cell state, this portion decides what information is propagated forward (i.e., output o(t) and cell state c(t)) to the next node in the network.

Therefore, the LSTM network is ideal for the price of one stock can affect the price of several other stocks over a long period of time. You can also decide the length of information about a particular past trend to keep stock prices moving, rising and stable at a particular rate for long period of time. More accurate predictions of future trends in stock price change
The main advantage of an LSTM is its ability to learn contextspecific temporal dependence. Each LSTM unit remembers information for either a long or a shortperiod of time (hence the name) without explicitly using an activation function within the recurrent components

MERITS OF THE PROPOSED SYSTEM

While experimenting with LSTM, we have observed that dropouts introduce a bottleneck in the adjustment of the model’s parameters. In many machine learning processes it is useful to know how certain the output of a model is. For example, a prediction is more likely to be closer to the actual price when an input is very similar to elements of the training set [21]. The outputs of a dropout layer are randomly ignored, therefore having the effect of reducing the capacity of a network during training. Requiring more nodes in the context of dropout could potentially remove this bottleneck. we have observed that increasing the nodes gives more positive outcomes. The LSTM supports this argument since dropout layers are absence hence the better performance

IMPLEMENTATION DETAILS
As mentioned in Long Short-Term Memory (LSTM) was introduced by Hochreiter and Schmidhuber in 1997 [16] to cope with the problem of long-term dependencies. LSTM

consist of a similar RNN architecture that has been shown to outperform traditional RNN on numerous tasks [16]. LSTM networks work extremely well with sequence data and long-term dependencies due to their powerful learning capabilities and memory mechanisms. By introducing gates they were able to improve memory capacity and control the memory cell. One gate is dedicated for reading out the entries from the cell, the output gate. Another gate is needed to decide when data should be read into the cell, this is called the input gate. Finally a forget gate which resets the content of the cell. This design was used in order to decide when to remember and ignore inputs at the hidden state. A sigmoid activation function computes the values of the three gates, these values belong in the range of (0, 1), and represent the current time step and hidden state of the previous time step. The hidden states values are then calculated with a gated version of the tangent activation function of the memory cell which take values in the range of (-1, 1)

Data Size	Stock Name	LSTM (RMSE)	ANN (RMSE)
Small	Dixon hughes	0.04	0.17
Medium	Copper Tire And Rubber	0.25	0.35
Medium	PNC Financial	0.2	0.28
Large	Citi Group	0.02	0.04
Large	Alcoa Group	0.02	0.04

Table 1 Comparison of error rates from LSTM and ANN

In Table 1, the LSTM model gave an RMSE (Root Mean Squared Error) value of 0.04, while the ANN model gave 0.17 for Dixon Hughes. For Cooper Tire & Rubber, LSTM gave an RMSE of 0.25 and ANN gave 0.35. For PNC Financial, the corresponding RMSE values were 0.2 and 0.28 respectively. For Citigroup, LSTM gave an RMSE of 0.02 and ANN gave 0.04, while for American Airlines, LSTM gave an RMSE of 0.02 and ANN gave 0.04

4. Module Description

In python have use several libraries, and information about few of them that are used to develop this system is below:

- Numpy:

It offers powerful N-Dimensional Array that are fast and versatile and also helps vectorization and indexing. It also offers itself as a numerical computing tool which can solve many mathematical functions, linear algebra routines or fourier transformations. The core of Numpy is a well-optimized C code so it provides flexibility with the speed.

- Panda

Panda is the fast easy to use tool which is use for data analysis that have been built over the python programming language. It is a powerful tool which can be used for data manipulation. It is one of the most important libraries in the field of Data Analysis and Data Science.

- Sklearn:

It is a simple and efficient tool which is used for data prediction. It is open source so it is available to everyone and is reusable to various context. It is built on the basis of NumPy, SciPy and matplotlib. It can be used for classification, Regression and Clustering

CODE -:

```
import numpy as np
import math
import matplotlib.pyplot as plt
import pandas as pd
data_set = pd.read_csv('NSE-TATAGLOBAL.csv')
data_set
data_set.head(5)
data_set.dtypes
training_set = data_set.iloc[:, 1:2].values
training_set
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range = (0, 1))

data_training_scaled = scaler.fit_transform(training_set)
features_set = []
labels = []
for        i         in         range(60,         586):
    features_set.append(data_training_scaled[i-60:i, 0])
    labels.append(data_training_scaled[i, 0])
features_set, labels = np.array(features_set), np.array(labels)
features_set = np.reshape(features_set, (features_set.shape[0],
features_set.shape[1], 1))
features_set.shape
import tensorflow as tf
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dense
from tensorflow.python.keras.layers import LSTM
m = Sequential()

m.compile(optimizer = 'adam', loss = 'mean_squared_error')

m.fit(features_set, labels, epochs = 50, batch_size = 20)
dataset_test = pd.read_csv('NSE-TATAGLOBAL.csv')
real_stock_price = dataset_test.iloc[:, 1:2].values
real_stock_price
data_total = pd.concat((data_set['Open'], data_set['Open']), axis=0)
test_inputs = data_total[len(data_total) - len(data_set) - 60:].values
test_inputs.shape
test_inputs = test_inputs.reshape(-1,1)
test_inputs = scaler.transform(test_inputs)
test_features = []
for i in range(60, 89):
    test_features.append(test_inputs[i-60:i, 0])
test_features = np.array(test_features)
test_features = np.reshape(test_features, (test_features.shape[0],
test_features.shape[1], 1))
test_features.shape
predictions = m.predict(test_features)
predictions
```

```
plt.figure(figsize=(24,9))
plt.title("STOCK PREDICTED")
plt.plot(dataset_test['Close'])
plt.xlabel('PRICE',fontsize=14)
plt.ylabel('Quantity',fontsize=14)
plt.show()
```

3 Data Connectivity:

In the option 1, we use the concept of Data Visualization . Below are the code snippets of the program that was used during that process:

```
In [2]: data_set = pd.read_csv('NSE-TATAGLOBAL.csv')
        data_set
```

Out[2]:

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-09-28	234.05	235.95	230.20	233.50	233.75	3069914	7162.35
1	2018-09-27	234.55	236.80	231.10	233.80	233.25	5082859	11859.95
2	2018-09-26	240.00	240.00	232.50	235.00	234.25	2240909	5248.60
3	2018-09-25	233.30	236.75	232.00	236.25	236.10	2349368	5503.90
4	2018-09-24	233.55	239.20	230.75	234.00	233.30	3423509	7999.55
...	...	...	...	...	...	...	...	...
2030	2010-07-27	117.60	119.50	112.00	118.80	118.65	586100	694.98
2031	2010-07-26	120.10	121.00	117.10	117.10	117.60	658440	780.01
2032	2010-07-23	121.80	121.95	120.25	120.35	120.65	281312	340.31
2033	2010-07-22	120.30	122.00	120.25	120.75	120.90	293312	355.17
2034	2010-07-21	122.10	123.00	121.05	121.10	121.55	658666	803.56

Importing all the required libraires and establishing connections

```
import numpy as np
import math
import matplotlib.pyplot as plt
import pandas as pd
```

Reading in the system and checking for itsTYPES:

```
data_set.dtypes
```

```
Date                      object
Open                     float64
High                     float64
Low                      float64
Last                     float64
Close                    float64
Total Trade Quantity       int64
Turnover (Lacs)          float64
dtype: object
```

CODE SNIPPETS:

1.) Loading of dataset in the program

```
data_set.tail(5)
```

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
2030	2010-07-27	117.6	119.50	112.00	118.80	118.65	586100	694.98
2031	2010-07-26	120.1	121.00	117.10	117.10	117.60	658440	780.01
2032	2010-07-23	121.8	121.95	120.25	120.35	120.65	281312	340.31
2033	2010-07-22	120.3	122.00	120.25	120.75	120.90	293312	355.17
2034	2010-07-21	122.1	123.00	121.05	121.10	121.55	658666	803.56

```
data_set.dtypes
```

```
Date                      object
Open                     float64
High                     float64
Low                      float64
Last                     float64
Close                    float64
Total Trade Quantity       int64
Turnover (Lacs)          float64
dtype: object
```

Dividing data in training and test data and features selections also:

]:
```
training_set = data_set.iloc[:, 1:2].values
```

]:
```
training_set
```

]:
```
array([[234.05],
       [234.55],
       [240.  ],
       ...,
       [121.8 ],
       [120.3 ],
       [122.1 ]])
```

```
dataset_test = pd.read_csv('NSE-TATAGLOBAL.csv')
real_stock_price = dataset_test.iloc[:, 1:2].values
```

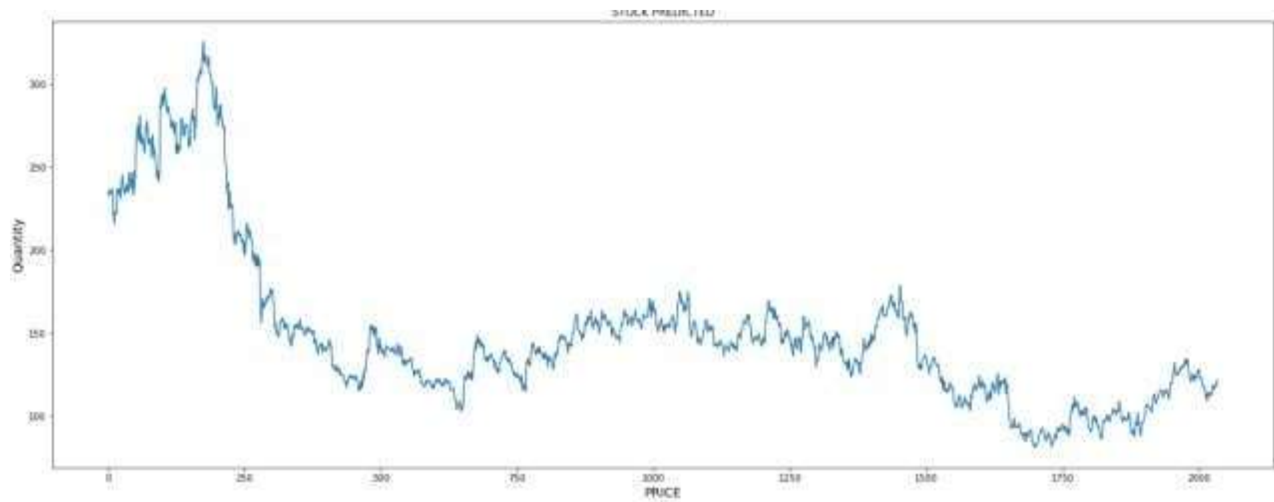
```
real_stock_price
```

```
rray([[234.05],
     [234.55],
     [240.  ],
     ...,
     [121.8 ],
     [120.3 ],
     [122.1 ]])
```

Importing and Training Model

```
data_total = pd.concat((data_set['Open'], data_set['Open']), axis=0)
test_inputs = data_total[len(data_total) - len(data_set) - 60:].values
test_inputs.shape
test_inputs = test_inputs.reshape(-1,1)
test_inputs = scaler.transform(test_inputs)
test_features = []
for i in range(60, 89):
    test_features.append(test_inputs[i-60:i, 0])
test_features = np.array(test_features)
test_features = np.reshape(test_features, (test_features.shape[0], test_features.shape[1], 1))
test_features.shape
predictions = m.predict(test_features)
predictions
```

OUTPUT



We trained a Machine Learning Model and used that to predict the best fertilizers. Below is the prediction test result that came after when we checked our model for the test data with the code snippet too

Chapter 5: Conclusion and Future Scope

Conclusion

The objective for this study is to identify directions for future machine learning stock market prediction research based upon a review of current literature. Given the ML-related systems, problem contexts, and findings described in each selected article, and the taxonomy categories presented earlier, several conclusions can be made about our current knowledge in this research area. First, there is a strong link between ML methods and the prediction problems they are associated with. This is analogous to task-technology fit (Goodhue and Thompson, 1995) where system performance is determined by the appropriate match between tasks and technologies. Artificial neural networks are best used for predicting numerical stock market index values. Support vector machines best fit classification problems such as determining whether the overall stock market index is forecast to rise or fall.

Future Enhancements

The second conclusion from this review of past studies is that generalizability of findings needs to be improved. Most studies evaluate their ML system using one market and/or one time period without considering whether the system will be effective in other situations. Three enhancements can be made for the experimental system assessment. First, many of the studies are based on results from Asian stock markets. These systems could also be tested in the same time period for US or European markets. Second, the systems could be evaluated using data from times where markets are rising or when markets are declining to assess how they perform in different market environments. For example, would an approach accurately predict market values in the US during the financial crisis of 2008-2009 and also during the recent market growth period from 2018-2019? If systems are able to predict market growth, are they also able to predict market contraction? Finally, proposed methods could be used to evaluate predictive performance for stock market indices that include only small firms vs. only large firms. Are systems effective under different risk and volatility environments? Any of these experimental method enhancements will provide a stronger research and practice contribution. The final set of conclusions was also apparent after reflection. Financial investment theory needs to be a stronger driver underlying the ML systems' inputs, algorithms, and performance measures. If this is not the case then results may just be random and not have any practical use. Too many studies use techniques without consideration of the vast amount of financial theory that has been developed over the past centuries. Reporting failures where techniques do not improve predictive performance would also be informative. At this point this rarely occurs so it is impossible to find patterns where there is a mismatch between a particular stock market prediction problem and a machine learning technique. Finally, the irony in this research area is that it is a zero-sum game for investors. If the best machine learning stock market prediction technique is found, and all investors adopt this system, the result is that no one is better off. Large investment firms researching the best machine learning methods have no incentive to share this information with others.

REFERENCES

1) Cabitza, F., Locoro, A., & Banfi, G. (2018). Machine learning in orthopedics: a literature review. Frontiers in Bioengineering and Biotechnology, 6, 75.

2) Chavan, P. S., & Patil, S. T. (2013). Parameters for stock market prediction. International Journal of Computer Technology and Applications, 4(2), 337.

3) Chiu, D. Y., & Chen, P. J. (2009). Dynamically exploring internal mechanism of stock market by fuzzy-based support vector machines with high dimension input space and genetic algorithm. Expert Systems with Applications, 36(2), 1240-1248.

4) Chong, E., Han, C., & Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. Expert Systems with Applications, 83, 187-205.

5) Dai, W., Wu, J. Y., & Lu, C. J. (2012). Combining nonlinear independent component analysis and neural network for the prediction of Asian stock market indexes. Expert systems with applications, 39(4), 4444-4452.

6) Das, S. P., & Padhy, S. (2012). Support vector machines for prediction of futures prices in Indian stock market. International Journal of Computer Applications, 41(3)

7) Ruder, Sebastian. "An overview of gradient descentoptimization algorithms." arXiv preprint arXiv:1609.04747.(2016).

8) Nazar, Nasrin Banu, and Radha Senthilkumar. "An online approach for feature selection for classification in big data." Turkish Journal of Electrical Engineering & Computer Sciences 25.1 (2017): 163-171

9) S. Siami-Namini and A. S. Namin. Forecasting economics and financial time series: Arima vs. lstm, 2018

10) Recht, Benjamin, et al. "Hogwild: A lock-free approach to parallelizing stochastic gradient descent." Advances in neural information processing systems. 2011.

11) Ding, Y., Zhao, P., Hoi, S. C., Ong, Y. S. "An Adaptive Gradient Method for Online AUC Maximization"In AAAI (pp. 2568-2574). (2015, January).

12) Soulas, Eleftherios, and Dennis Shasha. "Online machine learning algorithms for currency exchange prediction." Computer Science Department in New York University, Tech. Rep 31 (2013)

.

13) Zhao, P., Hoi, S. C., Wang, J., Li, B. “Online transfer learning”. Artificial Intelligence, 216, 76-102. (2014)

14) Altinbas, H., Biskin, O. T. “Selecting macroeconomic influencers on stock markets by using feature selection algorithms”. Procedia Economics and Finance, 30, 22-29. (2015).