A PRELIMENERY REPORT ON

STOCK PRICE PREDICTION USING STACKED LSTM

SUBMITTED TO THE SAVITRIBAI PHULE PUNE
UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

SUJAN SHAIKH	B190814256
VAISHNAVI JADHAV	B190814269
MOHAN BADHEKAR	B190814207

DEPARTMENT OF COMPUTER ENGINEERING

ISB&M COLLEGE OF ENGINEERING,

NANDE, PUNE 412115

SAVITRIBAI PHULE PUNE UNIVERSITY

2022-23

ISBM COLLEGE OF ENGINEERING
NANDE, PUNE 412115
DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the project report entitles

"STOCK PRICE PREDICTION USING LSTM"

Submitted by

SUJAN SHAIKH	B190814256
VAISHNAVI JADHAV	B190814269
MOHAN BADHEKAR	B190814207

are bonafide students of this institute and the work has been carried out by him/her under the supervision of **Prof. B. B. Gite** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

(Prof. B. B. Gite)
Guide,
Department of Computer Engineering

(Prof. B. B. Gite)
Head,
Department of Computer Engineering

(Prof. P. K. Srivastava)
Principal,
ISBM College of Engineering Pune-412115

Place: Pune
Date

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **Prof. B. B. Gite**, Department of Computer Engineering, for his guidance with unsurpassed knowledge and immense encouragement. We are also grateful for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ISBM, Pune**, for their encouragement and cooperation to carry out this work.

We thank all **teaching faculty** of Department of CS, whose suggestions during reviews helped us in accomplishment of our project

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last, but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

NAME OF STUDENTS

SUJAN SHAIKH

VAISHNAVI JADHAV

MOHAN BADHEKAR

ABSTRACT

In this project we attempt to implement machine learning approach to predict stock prices. Machine learning is effectively implemented in forecasting stock prices. The objective is to predict the stock prices in order to make more informed and accurate investment decisions. We propose a stock price prediction system that integrates mathematical functions, machine learning, and other external factors for the purpose of achieving better stock prediction accuracy and issuing profitable trades.

There are two types of stock trading. You may know of intraday trading by the commonly used term "day trading." Intraday traders hold securities positions from at least one day to the next and often for several days to weeks or months. LSTMs are very powerful in sequence prediction problems because they're able to store past information. This is important in our case because the previous price of a stock is crucial in predicting its future price. While predicting the actual price of a stock is an uphill climb, we can build a model that will predict whether the price will go up or down for next 30 days

Keywords: LSTM, CNN, ML, DL, Trade Open, Trade Close, Trade Low, Trade High

TABLE OF CONTENTS

LIST OF SYMBOLS

SYMBOLS	ILLUSTRATIONS
Xt	Input at current state
X(t-1)	Input at Previous state
Ct	Current Cell State
C(t-1)	Previous Cell Stat
Ht	Current hidden/output State
h(t-1)	Previous hidden/output State
σ	Sigmoid Function
tanh	Hyperbolic tangent function

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

ABBREVIATION	ILLUSTRATION
LSTM	Long Short-Term Memory
ATS	Automated Trading System
GRU	Gated Recurrent Unit
ML	Machine Learning
SVM	Support Vector Machine
EMH	Efficient Market hypothesis
AI	Artificial Intelligence
NN	Neural Networks
ARMA	Autoregressive Moving Average
DRL	Deep Reinforcement Learning
LM	Least Mean Square
UML	Unified modelling Language
MSE	Mean Squared Error
RMS	Root Mean Squared Error

CHAPTER 1
INTRODUCTION

The financial market is a dynamic and composite system where people can buy and sell currencies, stocks, equities and derivatives over virtual platforms supported by brokers. The stock market allows investors to own shares of public companies through trading either by exchange or over the counter markets. This market has given investors the chance of gaining money and having a prosperous life through investing small initial amounts of money, low risk compared to the risk of opening new business or the need of high salary career. Stock markets are affected by many factors causing the uncertainty and high volatility in the market. Although humans can take orders and submit them to the market, automated trading systems (ATS) that are operated by the implementation of computer programs can perform better and with higher momentum in submitting orders than any human. However, to evaluate and control the performance of ATSs, the implementation of risk strategies and safety measures applied based on human judgements are required. Many factors are incorporated and considered when developing an ATS, for instance, trading strategy to be adopted, complex mathematical functions that reflect the state of a specific stock, machine learning algorithms that enable the prediction of the future stock value, and specific news related to the stock being analysed.

Time-series prediction is a common technique widely used in many real-world applications such as weather forecasting and financial market prediction. It uses the continuous data in a period of time to predict the result in the next time unit. Many timeseries prediction algorithms have shown their effectiveness in practice. The most common algorithms now are based on Recurrent Neural Networks (RNN), as well as its special type - Long-short Term Memory (LSTM) and Gated Recurrent Unit (GRU). Stock market is a typical area that presents time-series data and many researchers' studies on it and proposed various models. In this project, LSTM model is used to predict the stock price.

1.1 MOTIVATION FOR WORK

Businesses primarily run over customer's satisfaction, customer reviews about their products. Shifts in sentiment on social media have been shown to correlate with shifts in stock markets. Identifying customer grievances thereby resolving them leads to customer satisfaction as well as trustworthiness of an organization. Hence there is a necessity of an un biased automated system to classify customer reviews regarding any problem. In today's environment where we're justifiably suffering from data overload (although this does not mean better or deeper insights), companies might have mountains of customer feedback collected; but for mere humans, it's still impossible to analyse it manually without any sort of error or bias. Oftentimes, companies with the best intentions find themselves in an insights vacuum. You

know you need insights to inform your decision making and you know that you're lacking them, but don't know how best to get them. Sentiment analysis provides some answers into what the most important issues are, from the perspective of customers, at least. Because sentiment analysis can be automated, decisions can be made based on a significant amount of data rather than plain intuition.

1.2 PROBLEM STATEMENT

Time Series forecasting & modelling plays an important role in data analysis. Time series analysis is a specialized branch of statistics used extensively in fields such as Econometrics & Operation Research. Time Series is being widely used in analytics & data science. Stock prices are volatile in nature and price depends on various factors. The main aim of this project is to predict stock prices using Long short term memory (LSTM).

CHAPTER 2
LITERATURE SURVEY

2.1 INTRODUCTION

"What other people think" has always been an important piece of information for most of us during the decision-making process. The Internet and the Web have now (among other things) made it possible to find out about the opinions and experiences of those in the vast pool of people that are neither our personal acquaintances nor well-known professional critics — that is, people we have never heard of. And conversely, more and more people are making their opinions available to strangers via the Internet. The interest that individual users show in online opinions about products and services, and the potential influence such opinions wield, is something that is driving force for this area of interest. And there are many challenges involved in this process which needs to be walked all over in order to attain proper outcomes out of them. In this survey we analysed basic methodology that usually happens in this process and measures that are to be taken to overcome the challenges being faced.

2.2 EXISTING METHODS

2.2.1 Stock Market Prediction Using Machine Learning

The research work done by V Kranthi Sai Reddy Student, ECM, Sreenidhi Institute of Science and Technology, Hyderabad, India. 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. 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. In this context this study uses a machine learning technique called Support Vector Machine (SVM) to predict stock prices for the large and small capitalizations and in the three different markets, employing prices with both daily and up-to-the-minute frequencies.

2.2.2 Forecasting the Stock Market Index Using Artificial Intelligence Techniques

The research work done by Lufuno Ronald Marwala A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand,

Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Engineering. The weak form of Efficient Market hypothesis (EMH) states that it is impossible to forecast the future price of an asset based on the information contained in the historical prices of an asset. This means that the market behaves as a random walk and as a result makes forecasting impossible. Furthermore, financial forecasting is a difficult task due to the intrinsic complexity of the financial system. The objective of this work was to use artificial intelligence (AI) techniques to model and predict the future price of a stock market index. Three artificial intelligence techniques, namely, neural networks (NN), support vector machines and neuro-fuzzy systems are implemented in forecasting the future price of a stock market index based on its historical price information. Artificial intelligence techniques have the ability to take into consideration financial system complexities and they are used as financial time series forecasting tools.

Two techniques are used to benchmark the AI techniques, namely, Autoregressive Moving Average (ARMA) which is linear modelling technique and random walk (RW) technique. The experimentation was performed on data obtained from the Johannesburg Stock Exchange. The data used was a series of past closing prices of the All Share Index. The results showed that the three techniques have the ability to predict the future price of the Index with an acceptable accuracy. All three artificial intelligence techniques outperformed the linear model. However, the random walk method out performed all the other techniques. These techniques show an ability to predict the future price however, because of the transaction costs of trading in the market, it is not possible to show that the three techniques can disprove the weak form of market efficiency. The results show that the ranking of performances support vector machines, neuro-fuzzy systems, multilayer perceptron neural networks is dependent on the accuracy measure used.

2.2.3 Indian stock market prediction using artificial neural networks on tick data

The research work done by Dharmaraja Selvamuthu, Vineet Kumar and Abhishek Mishra Department of Mathematics, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India. A stock market is a platform for trading of a company's stocks and derivatives at an agreed price. Supply and demand of shares drive the stock market. In any country stock market is one of the most emerging sectors. Nowadays, many people are

indirectly or directly related to this sector.Therefore, it becomes essential to
know about market trends. Thus, with the development of the stock 5 market, people are interested in forecasting stock price. But, due to dynamic nature and liable to quick changes in stock price, prediction of the stock price becomes a challenging task. Stock m Prior work has proposed effective methods to learn event representations that can capture syntactic and semantic information over text corpus, demonstrating their effectiveness for downstream tasks such as script event prediction. On the other hand, events extracted from raw texts lacks of common-sense knowledge, such as the intents and emotions of the event participants, which are useful for distinguishing event pairs when there are only subtle differences in their surface realizations. To address this issue, this paper proposes to leverage external common-sense knowledge about the intent and sentiment of the event.

Experiments on three event-related tasks, i.e., event similarity, script event prediction and stock market prediction, show that our model obtains much better event embeddings for the tasks, achieving 78% improvements on hard similarity task, yielding more precise inferences on subsequent events under given contexts, and better accuracies in predicting the volatilities of the stock market1. Markets are mostly a nonparametric, non-linear, noisy and deterministic chaotic system (Ahangar et al. 2010). As the technology is increasing, stock traders are moving towards to use Intelligent Trading Systems rather than fundamental analysis for predicting prices of stocks, which helps them to take immediate investment decisions. One of the main aims of a trader is to predict the stock price such that he can sell it before its value decline, or buy the stock before the price rises. The efficient market hypothesis states that it is not possible to predict stock prices and that stock behaves in the random walk. It seems to be very difficult to replace the professionalism of an experienced trader for predicting the stock price. But because of the availability of a remarkable amount of data and technological advancements we can now formulate an appropriate algorithm for prediction whose results can increase the profits for traders or investment firms. Thus, the accuracy of an algorithm is directly proportional to gains made by using the algorithm.

2.2.4 The Stock Market and Investment

The research work done by Manh Ha Duong Boriss Siliverstovs. Investigating the relation between equity prices and aggregate investment in major European countries including

France, Germany, Italy, the Netherlands and the United Kingdom. Increasing integration of European financial markets is likely to result in even stronger correlation between equity prices in different European countries. This process can also lead to convergence in economic development across European countries if developments in stock markets influence real economic components, such as investment and consumption. Indeed, our vector autoregressive models suggest that the positive correlation between changes equity prices and investment is, in general, significant. Hence, 6 monetary authorities should monitor reactions of share prices to monetary policy and their effects on the business cycle.

2.2.5 Automated Stock Price Prediction Using Machine Learning

The research work done by Mariam Moukalled Wassim El-Hajj Mohamad Jaber Computer Science Department American University of Beirut. Traditionally and in order to predict market movement, investors used to analyse the stock prices and stock indicators in addition to the news related to these stocks. Hence, the importance of news on the stock price movement. Most of the previous work in this industry focused on either classifying the released market news as (positive, negative, neutral) and demonstrating their effect on the stock price or focused on the historical price movement and predicted their future movement. In this work, we propose an automated trading system that integrates mathematical functions, machine learning, and other external factors such as news' sentiments for the purpose of achieving better stock prediction accuracy and issuing profitable trades. Particularly, we aim to determine the price or the trend of a certain stock for the coming end-of-day considering the first several trading hours of the day. To achieve this goal, we trained traditional machine learning algorithms and created/trained multiple deep learning models taking into consideration the importance of the relevant news. Various experiments were conducted, the highest accuracy (82.91%) of which was achieved using SVM for Apple Inc. (AAPL) stock.

2.2.6 Stock Price Correlation Coefficient Prediction with ARIMALSTM Hybrid Model

The research work done by Hyeong Kyu Choi, B.A Student Dept. of Business Administration Korea University Seoul, Korea. Predicting the price correlation of two assets for future time periods is important in portfolio optimization. We apply LSTM recurrent neural networks (RNN) in predicting the stock price correlation coefficient of two individual stocks. RNN's are competent in understanding temporal dependencies. The use of LSTM cells further

enhances its long-term predictive properties. To encompass both linearity and nonlinearity in the model, we adopt the ARIMA model as well. The ARIMA model filters linear tendencies in the data and passes on the residual value to the LSTM model. The ARIMA-LSTM hybrid model is tested against other traditional predictive financial models such as the full historical model, constant correlation model, single-index model and the multi-group model. In our empirical study, the predictive ability of the ARIMA-LSTM model turned out superior to all other financial models by a significant scale. Our work implies that it is worth considering the ARIMALSTM model to forecast correlation coefficient for portfolio optimization.

2.2.7 Event Representation Learning Enhanced with External Common-sense Knowledge

The research work done by Xiao Ding, Kuo Liao, Ting Liu, Zhongyang Li, Junwen Duan Research Center for Social Computing and Information Retrieval Harbin Institute of Technology, China. Prior work has proposed effective methods to learn event representations that can capture syntactic and semantic information over text corpus, demonstrating their effectiveness for downstream tasks such as script event prediction. On the other hand, events extracted from raw texts lacks of common-sense knowledge, such as the intents and emotions of the event participants, which are useful for distinguishing event pairs when there are only subtle differences in their surface realizations. To address this issue, this paper proposes to leverage external common-sense knowledge about the intent and sentiment of the event. Experiments on three event-related tasks, i.e., event similarity, script event prediction and stock market prediction, show that our model obtains much better event embeddings for the tasks, achieving 78% improvements on hard similarity task, yielding more precise inferences on subsequent events under given contexts, and better accuracies in predicting the volatilities of the stock market.

2.2.8 Forecasting directional movements of stock prices for intraday trading using LSTM and random forests

The research work done by Pushpendu Ghosh, Ariel Neufeld, Jajati Keshari SahooDepartment of Computer Science & Information Systems, BITS Pilani K.K.Birla Goa campus, India bDivision of Mathematical Sciences, Nanyang Technological University, Singapore cDepartment of Mathematics, BITS Pilani K.K.Birla Goa campus, India. We employ both random forests and LSTM networks (more precisely CuDNNLSTM) as training methodologies to analyse their effectiveness in forecasting outof-sample directional

movements of constituent stocks of the S&P 500 from January 1993 till December 2018 for intraday trading. We introduce a multi-feature setting consisting not only of the returns with respect to the closing prices, but also with respect to the opening prices and intraday returns. As trading strategy, we use Krauss et al. (2017) and Fischer & Krauss (2018) as benchmark and, on each trading day, buy the 10 stocks with the highest probability and sell short the 10 stocks with the lowest probability to outperform the market in terms of intraday returns – all with equal monetary weight. Our empirical results show that the multi-feature setting provides a daily return, prior to transaction costs, of 0.64% using LSTM networks, and 0.54% using random forests. Hence, we outperform the singlefeature setting in Fischer & Krauss (2018) and Krauss et al. (2017) consisting only of the daily returns with respect to the closing prices, having corresponding daily returns of 0 .41% and of 0 .39% with respect to LSTM and random forests, respectively. 1 Keywords: Random forest, LSTM, Forecasting, Statistical Arbitrage, Machine learning, Intraday trading

2.2.9 A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance

The research work done by Xiao-Yang Liu1 Hongyang Yang,Qian Chen4,Runjia ZhangLiuqing Yang Bowen Xiao Christina Dan Wang Electrical Engineering, 2Department of Statistics, 3Computer Science, Columbia University, 3AI4Finance LLC., USA, Ion Media Networks, USA, Department of Computing, Imperial College, 6New York University (Shanghai). As deep reinforcement learning (DRL) has been recognized as an effective approach in quantitative finance, getting hands-on experiences is attractive to beginners. However, to train a practical DRL trading agent that decides where to trade, at what price, and what quantity involves error-prone and arduous development and debugging. In this paper, we introduce a DRL library FinRL that facilitates beginners to expose themselves to quantitative finance and to develop their own stock trading strategies. Along with easily-reproducible tutorials, FinRL library allows users to streamline their own developments and to compare with existing schemes easily. Within FinRL, virtual environments are configured with stock market datasets, trading agents are trained with neural networks, and extensive back testing is analysed via trading performance. Moreover, it incorporates important trading constraints such as transaction cost, market liquidity and the investor’s degree of risk-aversion. FinRL is featured with completeness, hands-on tutorial and reproducibility that favors beginners: (i) at multiple

levels of time granularity, FinRL simulates trading environments across various stock markets, including NASDAQ-100, DJIA, S&P 500, HSI, SSE 50, and CSI 300; (ii) organized in a layered architecture with modular structure, FinRL provides finetuned state-of-the-art DRL algorithms (DQN, DDPG, PPO, SAC, A2C, TD3, etc.), commonly used reward functions and standard evaluation baselines to alleviate the debugging workloads and promote the reproducibility, and (iii) being highly extendable, FinRL reserves a complete set of user-import interfaces. Furthermore, we incorporated three application demonstrations, namely single stock trading, multiple stock trading, and portfolio allocation. The FinRL library will be available on GitHub at link https://github.com/AI4Finance-LLC/FinRL-Library.

2.2.10 An innovative neural network approach for stock market prediction

The research work done by Xiongwen Pang, Yanqiang Zhou, Pan Wang, Weiwei Lin. To develop an innovative neural network approach to achieve better stock market predictions. Data were obtained from the live stock market for real-time and off-line analysis and results of visualizations and analytics to demonstrate Internet of Multimedia of Things for stock analysis. To study the influence of market characteristics on stock prices, traditional neural network algorithms may incorrectly predict the stock market, since the initial weight of the random selection problem can be easily prone to incorrect predictions. 9 Based on the development of word vector in deep learning, we demonstrate the concept of "stock vector." The input is no longer a single index or single stock index, but multi-stock high-dimensional historical data. We propose the deep long short-term memory neural network (LSTM) with embedded layer and the long short-term memory neural network with automatic encoder to predict the stock market. In these two models, we use the embedded layer and the automatic encoder, respectively, to vectorize the data, in a bid to forecast the stock via long short-term memory neural network. The experimental results show that the deep LSTM with embedded layer is better. Specifically, the accuracy of two models is 57.2 and 56.9%, respectively, for the Shanghai A-shares composite index. Furthermore, they are 52.4 and 52.5%, respectively, for individual stocks. We demonstrate research contributions in IMMT for neural network-based financial analysis. 2.2.11 An Intelligent Technique for Stock Market Prediction

2.2.11 An Intelligent Technique for Stock Market Prediction

The research work done by M. Mekayel Anik · M. Shamsul Arefin (B) Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chittagong, Bangladesh. A stock market is a loose network of economic transactions between

buyers and sellers based on stocks also known as shares. In stock markets, stocks represent the ownership claims on businesses. These may include securities listed on a stock exchange as well as those only traded privately. A stock exchange is a place where brokers can buy and/or sell stocks, bonds, and other securities. Stock market is a very vulnerable place for investment due to its volatile nature. In the near past, we faced huge financial problems due to huge drop in price of shares in stock markets worldwide. This phenomenon brought a heavy toll on the international as well as on our national financial structure. Many people lost their last savings of money on the stock market. In 2010–2011 financial year, Bangladeshi stock market faced massive collapse [1]. This phenomenon can be brought under control especially by strict monitoring and instance stock market analysis. If we can analyse stock market correctly in time, it can become a field of large profit and may become comparatively less vulnerable for the investors. Stock market is all about prediction and rapid decision making about investment, which cannot be done without thorough analysis of the market. If we can predict the stock market by analysing historical data properly, we can avoid the consequences of serious market collapse and to be able to take necessary steps to make market immune to such situations

CHAPTER 3
METHODOLOGY

3.1 Proposed System

The prediction methods can be roughly divided into two categories, statistical methods and artificial intelligence methods. Statistical methods include logistic regression model, ARCH model, etc. Artificial intelligence methods include multi-layer perceptron, convolutional neural network, naive Bayes network, back propagation network, single-layer LSTM, support vector machine, recurrent neural network, etc. They used Long short-term memory network (LSTM).

Long short-term memory network:

Long short-term memory network (LSTM) is a particular form of recurrent neural network (RNN).

3.2 Working of LSTM:

LSTM is a special network structure with three “gate” structures.

Three gates are placed in an LSTM unit, called input gate, forgetting gate and output gate. While information enters the LSTM’s network, it can be selected by rules. Only the information conforms to the algorithm will be left, and the information that does not conform will be forgotten through the forgetting gate.

3.3 LSTM Architecture

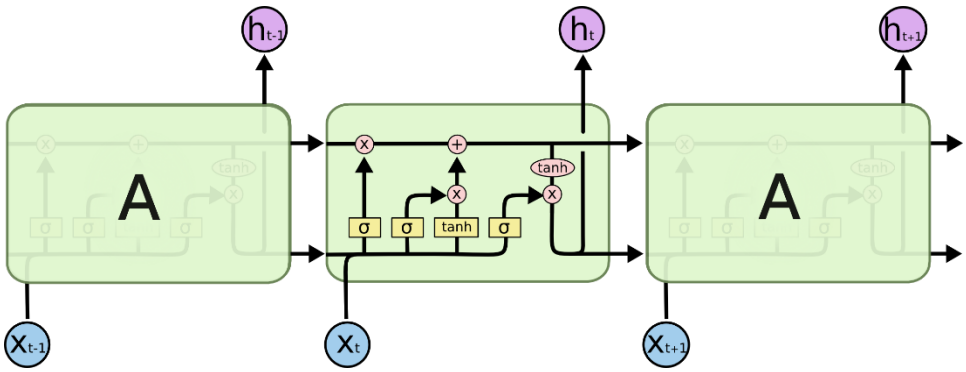


Fig. 3.1: LSTM Architecture

Forget Gate:

A forget gate is responsible for removing information from the cell state.

- The information that is no longer required for the LSTM to understand things or the information that is of less importance is removed via multiplication of a filter.
- This is required for optimizing the performance of the LSTM network.
- This gate takes in two inputs; h_t-1 and x_t. h_t-1 is the hidden state from the previous cell or the output of the previous cell and x_t is the input at that particular time step.

Input Gate:

1. Regulating what values need to be added to the cell state by involving a sigmoid function. This is basically very similar to the forget gate and acts as a filter for all the information from hi-1 and x_t.
2. Creating a vector containing all possible values that can be added (as perceived from h_t-1 and x_t) to the cell state. This is done using the tanh function, which outputs values from -1 to +1.
3. Multiplying the value of the regulatory filter (the sigmoid gate) to the created vector (the tanh function) and then adding this useful information to the cell state via addition operation.

Output Gate:

The functioning of an output gate can again be broken down to three steps:

- Creating a vector after applying tanh function to the cell state, thereby scaling the values to the range -1 to +1.

- Making a filter using the values of h_t-1 and x_t, such that it can regulate the values that need to be output from the vector created above. This filter again employs a sigmoid function.

- Multiplying the value of this regulatory filter to the vector created in step 1, and sending it out as a output and also to the hidden state of the next cell.

```
# LSTM
Inputs: dataset
Outputs: RMSE of the forecasted data

# Split dataset into 80% training and 20% testing data
size = length(dataset) * 0.80
train = dataset [0 to size]
test = dataset [size to length(dataset)]

# Procedure to fit the LSTM model
Procedure LSTMAlgorithm (train, test, train_size, epochs)
    X_train = train
    Y_train= test
model = Sequential ()
model.add(LSTM(50,return_sequences=True,input_shape=(100,1)))
model.add(LSTM(50,return_sequences=True))
model.add(LSTM(50))
model.add(Dense(1))

# Procedure to make predictions
Procedure getPredictonsFromModel (model, X)
    predictions = model.predict(X)
    return predictions
epochs = 100

neurons = 50
predictions = empty
# Fit the LSTM model
model = LSTMAlgorithm (train, epoch, neurons)

# Make predictions
pred = model.predict(train)

# Validate the model
n = len(dataset)
```

```
•
•   error = 0
•   for i in range(n): error += (abs(real[i] - pred[i])/real[i]) * 100
•   accuracy = 100 - error/n
```

3.4 Hardware Requirements:

- RAM: 4 GB
- Storage: 500 GB
- CPU: 2 GHz or faster
- Architecture: 32-bit or 64-bit

3.5 Software Requirements:

- Python 3.5 in Jupyter Notebook is used for data pre-processing, model training and prediction.
- Operating System: windows 7 and above or Linux based OS or MAC OS

3.6 Functional requirements:

Functional requirements describe what the model should do (the functions). Think about the core operations.

Because the “functions” are established before development, functional requirements should be written in the future tense. In developing the model for Stock Price Prediction, some of the functional requirements could include:

- The model shall accept the tw_spydata_raw.csv dataset as input.
- The model should shall do pre-processing (like verifying for missing data values) on input for model training.
- The model shall use LSTM ARCHITECTURE as main component.
- It processes the given input data by producing the most possible outcomes of a CLOSING STOCK PRICE.
- Notice that each requirement is directly related to what we expect the model to do. They represent some of the core functions.

3.7 Non-Functional requirements

Product properties

- Usability: It defines the simplicity of understanding for any kind of stock trader and other stakeholders in stock market.
- Efficiency: maintaining the possible highest accuracy in the closing stock prices in shortest time with available data.
- Performance: It is a quality attribute of the stock prediction model that describes the responsiveness to various user interactions with it.

CHAPTER 4
SYSTEM ARCHITECTURE

4.1 Preprocessing of data

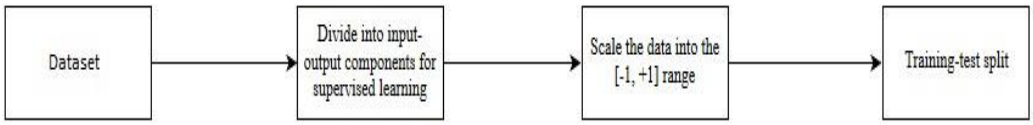


Fig. 4.1: Pre-processing of data

4.2 Overall Architecture

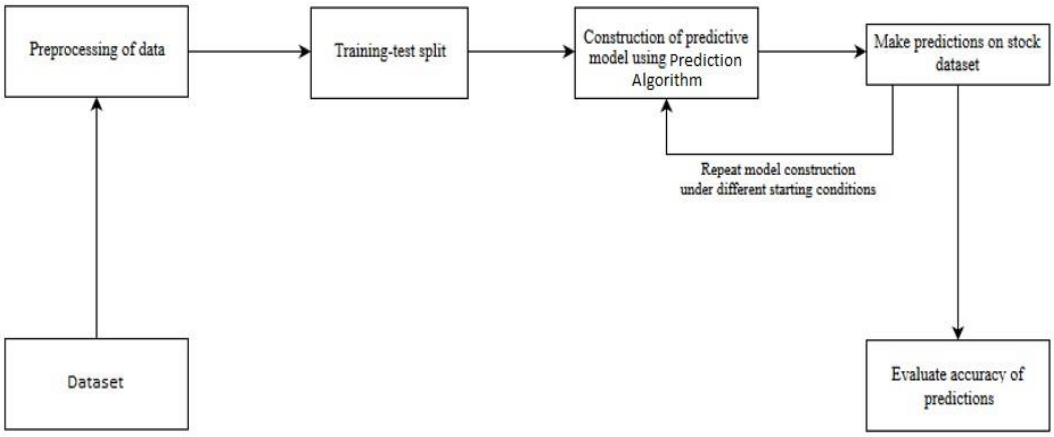


Fig. 4.2: Overall Architecture

CHAPTER 5
DESIGN

5.1 Structure Chart

A structure chart (SC) in software engineering and organizational theory is a chart which shows the breakdown of a system to its lowest manageable levels. They are used in structured programming to arrange program modules into a tree. Each module is represented by a box, which contains the module's name.

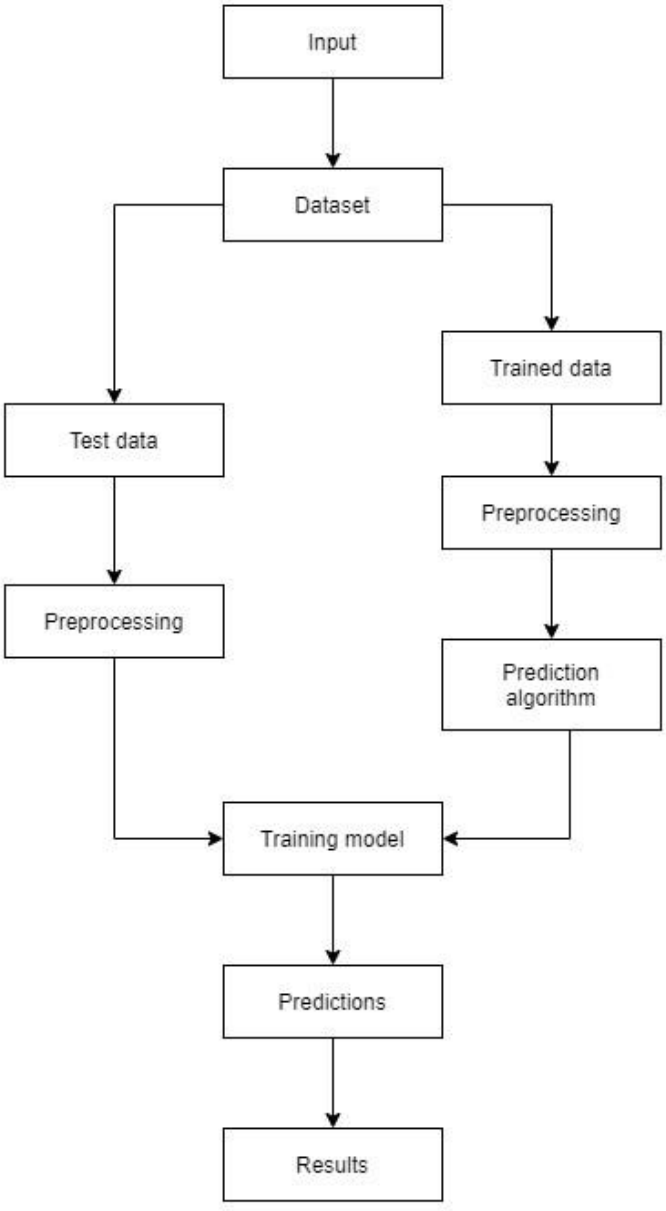


Fig. 5.1: Training and prediction

5.2 UML Diagrams

A UML diagram is a partial graphical representation (view) of a model of a system under design, implementation, or already in existence. UML diagram contains graphical elements (symbols) - UML nodes connected with edges (also known as paths or flows) - that represent elements in the UML model of the designed system. The UML model of the system might also contain other documentation such as use cases written as templated texts.

The kind of the diagram is defined by the primary graphical symbols shown on the diagram. For example, a diagram where the primary symbols in the contents area are classes is class diagram. A diagram which shows use cases and actors is use case diagram. A sequence diagram shows sequence of message exchanges between lifelines.

UML specification does not preclude mixing of different kinds of diagrams, e.g. to combine structural and behavioral elements to show a state machine nested inside a use case. Consequently, the boundaries between the various kinds of diagrams are not strictly enforced. At the same time, some UML Tools do restrict set of available graphical elements which could be used when working on specific type of diagram.

UML specification defines two major kinds of UML diagram: structure diagrams and behavior diagrams.

Structure diagrams show the static structure of the system and its parts on different abstraction and implementation levels and how they are related to each other. The elements in a structure diagram represent the meaningful concepts of a system, and may include abstract, real world and implementation concepts.

Behavior diagrams show the dynamic behavior of the objects in a system, which can be described as a series of changes to the system over time.

5.3 Activity Diagram

An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

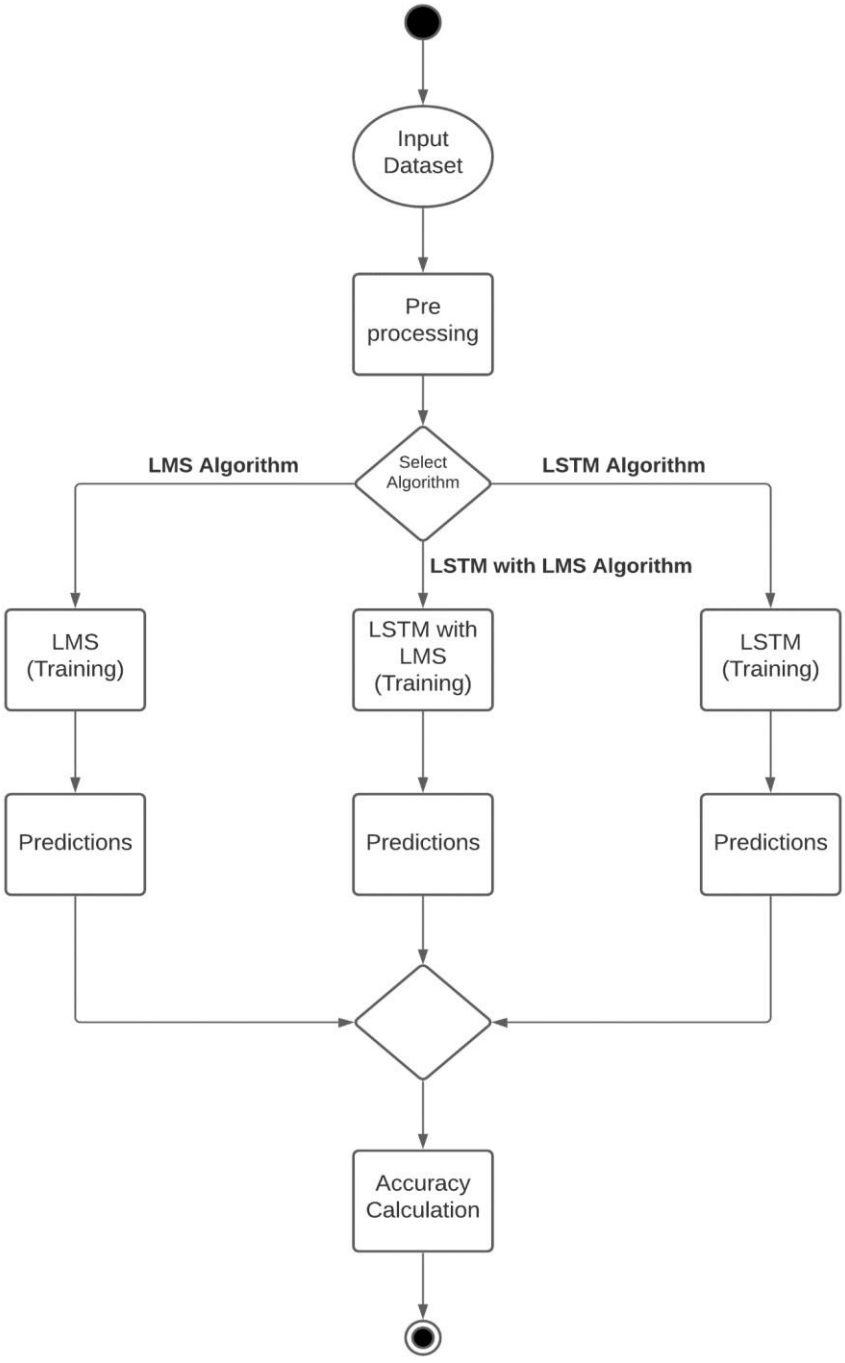


Fig. 5.3: Execution based on algorithm selection

5.4 Collaboration Diagram

Collaboration diagrams are used to show how objects interact to perform the behavior of a particular use case, or a part of a use case. Along with sequence diagrams, collaboration are used by designers to define and clarify the roles of the objects that perform a particular flow of events of a use case. They are the primary source of information used to determining class responsibilities and interfaces.

The collaborations are used when it is essential to depict the relationship between the object. Both the sequence and collaboration diagrams represent the same information, but the way of portraying it quite different. The collaboration diagrams are best suited for analyzing use cases

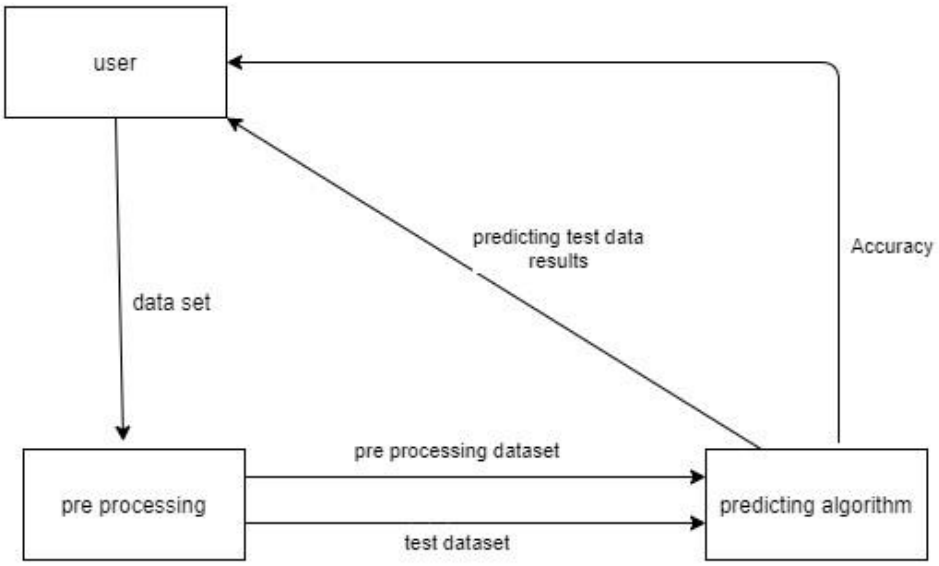


Fig. 5.4: Data transfer between modules

5.5 Flow Chart

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows.

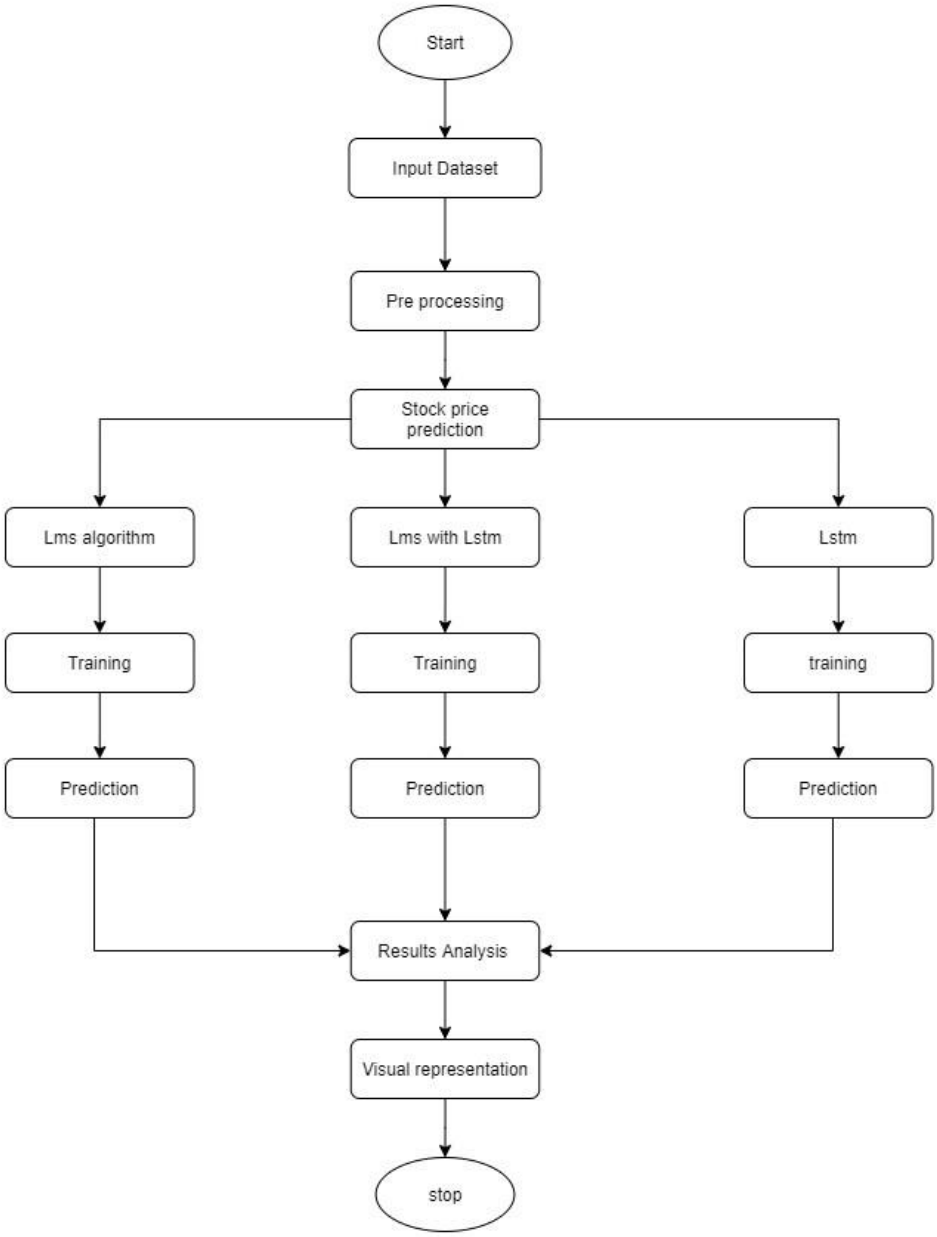


Fig. 5.5: Flow of execution

5.6 Component Diagram

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system

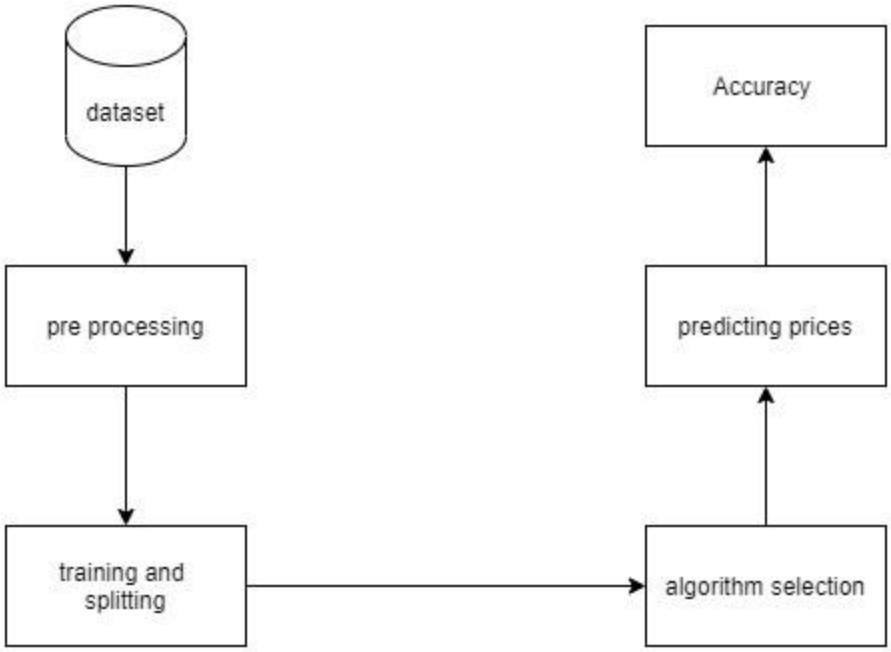


Fig. 5.6: Components present in the system

CHAPTER 6

EXPERIMENTAL ANALYSIS

6.1 System Configuration

This project can run on commodity hardware. We ran entire project on an Intel I5 processor with 8 GB Ram, 2 GB Nvidia Graphic Processor, It also has 2 cores which runs at 1.7 GHz, 2.1 GHz respectively. First part of the is training phase which takes 10-15 mins of time and the second part is testing part which only takes few seconds to make predictions and calculate accuracy.

6.1.1 Hardware Requirements:

- RAM: 4 GB
- Storage: 500 GB
- CPU: 2 GHz or faster
- Architecture: 32-bit or 64-bit

6.1.2 Software requirements

- Python 3.5 in Jupyter Notebook is used for data pre-processing, model training and prediction.
- Operating System: windows 7 and above.

6.2. Sample Code

Use pandas_datareader library to collect data for the Tesla's Stock from the Tiingo API and save it as a csv file

```
In [1]: ### Keras and Tensorflow >2.0

In [2]: ### Data Collection
        import pandas_datareader as pdr
        key="debee8d1a1092fd0ca4570863e6c2100eaa113bc"

In [3]: df = pdr.get_data_tiingo('TSLA', api_key=key)

        C:\Users\mohan\anaconda3\lib\site-packages\pandas_datareader\tiingo.py:234: FutureWarning: In a future version of pandas all ar
        guments of concat except for the argument 'objs' will be keyword-only.
          return pd.concat(dfs, self._concat_axis)

In [4]: df.to_csv('TSLA.csv')
```

Importing the pandas library and assign it the alias 'pd', and then reads a CSV file into a DataFrame called 'df'.

Then display the DataFrame using the head() and tail() methods.

This will show the first and last five rows of the DataFrame

```
In [5]: import pandas as pd

In [6]: df=pd.read_csv('TSLA.csv')

In [7]: df.head()
```

Out[7]:

	symbol	date	close	high	low	open	volume	adjClose	adjHigh	adjLow	adjOpen	adjVolume	divCash	splitFactor
0	TSLA	2018-05-21 00:00:00+00:00	284.49	291.49	281.30	281.33	9182571	18.966000	19.432667	18.753333	18.755333	137738565	0.0	1.0
1	TSLA	2018-05-22 00:00:00+00:00	275.01	288.00	273.42	287.76	8945756	18.334000	19.200000	18.228000	19.184000	134186340	0.0	1.0
2	TSLA	2018-05-23 00:00:00+00:00	279.07	279.91	274.00	277.76	5985053	18.604667	18.660667	18.266667	18.517333	89775795	0.0	1.0
3	TSLA	2018-05-24 00:00:00+00:00	277.85	281.11	274.89	278.40	4176708	18.523333	18.740667	18.326000	18.560000	62650620	0.0	1.0
4	TSLA	2018-05-25 00:00:00+00:00	278.85	279.64	275.61	277.63	3875082	18.590000	18.642667	18.374000	18.508667	58126230	0.0	1.0

```
In [8]: df.tail()
```

Out[8]:

	symbol	date	close	high	low	open	volume	adjClose	adjHigh	adjLow	adjOpen	adjVolume	divCash	splitFactor
1253	TSLA	2023-05-12 00:00:00+00:00	167.98	177.3800	167.2300	176.070	157849625	167.98	177.3800	167.2300	176.070	157849625	0.0	1.0
1254	TSLA	2023-05-15 00:00:00+00:00	166.35	169.7600	164.5499	167.655	105592510	166.35	169.7600	164.5499	167.655	105592510	0.0	1.0
1255	TSLA	2023-05-16 00:00:00+00:00	166.52	169.5184	164.3500	165.650	98288792	166.52	169.5184	164.3500	165.650	98288792	0.0	1.0
1256	TSLA	2023-05-17 00:00:00+00:00	173.86	174.5000	167.1850	168.410	125473558	173.86	174.5000	167.1850	168.410	125473558	0.0	1.0
1257	TSLA	2023-05-18 00:00:00+00:00	176.89	177.0600	172.4500	174.220	109520332	176.89	177.0600	172.4500	174.220	109520332	0.0	1.0

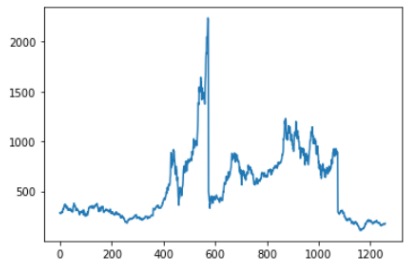
Now reset the index of DataFrame and select the ‘close’ column from it

The variable df1 will store the values from the ‘close’ column.

```
In [9]: df1=df.reset_index()['close']

In [10]: df1

Out[10]: 0       284.49
         1       275.01
         2       279.07
         3       277.85
         4       278.85
                  ...
         1253    167.98
         1254    166.35
         1255    166.52
         1256    173.86
         1257    176.89
         Name: close, Length: 1258, dtype: float64
```

Plot a DataFrame df1 using Matplotlib in python

```
In [11]: import matplotlib.pyplot as plt
         plt.plot(df1)

Out[11]: [<matplotlib.lines.Line2D at 0x1e922f02910>]
```



Apply MinMax scaling to the data in df1 using Minmaxscaler from sklearn.preprocessing module

The scaled data have values between 0 and 1, which can be suitable for training LSTM models

```
In [12]:
### LSTM are sensitive to the scale of the data. so we apply MinMax scaler

In [13]: import numpy as np

In [14]: from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler(feature_range=(0,1))
df1=scaler.fit_transform(np.array(df1).reshape(-1,1))

In [15]: print(df1)

[[0.08278694]
 [0.0783376 ]
 [0.08024312]
 ...
 [0.02741886]
 [0.03086382]
 [0.03228592]]
```

Split the dataset into training and test sets

```
In [16]: ##splitting dataset into train and test split
training_size=int(len(df1)*0.80)
test_size=len(df1)-training_size
train_data,test_data=df1[0:training_size,:],df1[training_size:len(df1),:1]

In [17]:
training_size,test_size

Out[17]: (1006, 252)

In [18]: train_data

Out[18]: array([[0.08278694],
       [0.0783376 ],
       [0.08024312],
       ...,
       [0.31046394],
       [0.28924037],
       [0.30671861]])
```

Convert an array of values into a dataset matrix and reshape the training and test data into input-output pairs with a 100 time step.

Print the shape of arrays.

```
In [19]: import numpy
         # convert an array of values into a dataset matrix
         def create_dataset(dataset, time_step=1):
             dataX, dataY = [], []
             for i in range(len(dataset)-time_step-1):
                 a = dataset[i:(i+time_step), 0]   ###i=0, 0,1,2,3-----99   100
                 dataX.append(a)
                 dataY.append(dataset[i + time_step, 0])
             return numpy.array(dataX), numpy.array(dataY)

In [20]: # reshape into X=t,t+1,t+2,t+3 and Y=t+4
         time_step = 100
         X_train, y_train = create_dataset(train_data, time_step)
         X_test, ytest = create_dataset(test_data, time_step)

In [21]: print(X_train.shape), print(y_train.shape)

         (905, 100)
         (905,)

Out[21]: (None, None)

In [22]: print(X_test.shape), print(ytest.shape)

         (151, 100)
         (151,)

Out[22]: (None, None)
```

Convert the input data into a 3D shape where first dimension represents the number of samples, the second dimension represents the number of time steps, and the third dimension represents the number of features

Import the necessary modules from Tensorflow Keras: Sequential, Dense and LSTM and create a sequential model.

Now we create a stacked LSTM model with three LSTM layers and a final dense layer.

```
In [23]: # reshape input to be [samples, time steps, features] which is required for LSTM
         X_train =X_train.reshape(X_train.shape[0],X_train.shape[1] , 1)
         X_test = X_test.reshape(X_test.shape[0],X_test.shape[1] , 1)

In [24]: ### Create the Stacked LSTM model
         from tensorflow.keras.models import Sequential
         from tensorflow.keras.layers import Dense
         from tensorflow.keras.layers import LSTM

In [25]: model=Sequential()
         model.add(LSTM(50,return_sequences=True,input_shape=(100,1)))
         model.add(LSTM(50,return_sequences=True))
         model.add(LSTM(50))
         model.add(Dense(1))
         model.compile(loss='mean_squared_error',optimizer='adam')
```

Summary of the model architecture

```
In [26]: model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 100, 50)           10400

 lstm_1 (LSTM)               (None, 100, 50)           20200

 lstm_2 (LSTM)               (None, 50)                20200

 dense (Dense)               (None, 1)                 51

=================================================================
Total params: 50,851
Trainable params: 50,851
Non-trainable params: 0
_________________________________________________________________
```

Initiate the training process of the model

```
In [27]: model.fit(X_train,y_train,validation_data=(X_test,ytest),epochs=100,batch_size=64,verbose=1)

Epoch 1/100
15/15 [==============================] - 24s 661ms/step - loss: 0.0263 - val_loss: 0.0031
Epoch 2/100
15/15 [==============================] - 6s 411ms/step - loss: 0.0079 - val_loss: 0.0025
Epoch 3/100
15/15 [==============================] - 6s 428ms/step - loss: 0.0065 - val_loss: 1.1938e-04
Epoch 4/100
15/15 [==============================] - 6s 407ms/step - loss: 0.0056 - val_loss: 7.2337e-04
Epoch 5/100
15/15 [==============================] - 6s 405ms/step - loss: 0.0051 - val_loss: 5.5578e-04
Epoch 6/100
15/15 [==============================] - 6s 409ms/step - loss: 0.0052 - val_loss: 1.4492e-04
Epoch 7/100
15/15 [==============================] - 6s 418ms/step - loss: 0.0045 - val_loss: 4.0099e-04
Epoch 8/100
15/15 [==============================] - 6s 394ms/step - loss: 0.0038 - val_loss: 1.5414e-04
Epoch 9/100
15/15 [==============================] - 6s 412ms/step - loss: 0.0036 - val_loss: 1.3813e-04
Epoch 10/100
```

Transform the predicted values back to their original scale and evaluate the performance of the model

In [28]:
```
import tensorflow as tf
```

In [29]:
```
### Lets Do the prediction and check performance metrics
train_predict=model.predict(X_train)
test_predict=model.predict(X_test)
```

```
29/29 [==============================] - 5s 72ms/step
5/5 [==============================] - 0s 71ms/step
```

In [30]:
```
##Transformback to original form
train_predict=scaler.inverse_transform(train_predict)
test_predict=scaler.inverse_transform(test_predict)
```

Calculate the Root Mean Squared Error (RMSE) and accuracy percentage.

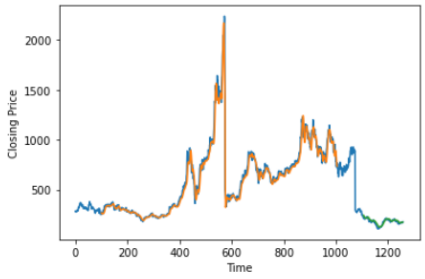
In [31]:
```
# calculate RMSE performance metrics
import math
from sklearn.metrics import mean_squared_error

train_rmse = math.sqrt(mean_squared_error(y_train, train_predict))
test_rmse = math.sqrt(mean_squared_error(ytest, test_predict))

# calculate accuracy percentage
max_val = scaler.inverse_transform([[1]])[0,0]
min_val = scaler.inverse_transform([[0]])[0,0]
train_accuracy = 100 - train_rmse * 100 / (max_val - min_val)
test_accuracy = 100 - test_rmse * 100 / (max_val - min_val)

# print results
print(f"Train RMSE: {train_rmse:.2f}, Train Accuracy: {train_accuracy:.2f}%")
print(f"Test RMSE: {test_rmse:.2f}, Test Accuracy: {test_accuracy:.2f}%")
```

```
Train RMSE: 698.61, Train Accuracy: 67.21%
Test RMSE: 186.05, Test Accuracy: 91.27%
```

Plot the baseline date as well as the predicted values for the training and test sets.

Compare the performance of the model's predictions with the actual data

In [32]:
```
### Plotting
# shift train predictions for plotting
look_back=100
trainPredictPlot = numpy.empty_like(df1)
trainPredictPlot[:, :] = np.nan
trainPredictPlot[look_back:len(train_predict)+look_back, :] = train_predict
# shift test predictions for plotting
testPredictPlot = numpy.empty_like(df1)
testPredictPlot[:, :] = numpy.nan
testPredictPlot[len(train_predict)+(look_back*2)+1:len(df1)-1, :] = test_predict
# plot baseline and predictions
plt.plot(scaler.inverse_transform(df1))
plt.plot(trainPredictPlot)
plt.plot(testPredictPlot)
plt.xlabel('Time')
plt.ylabel('Closing Price')

plt.show()
```



Convert a portion of the test data to a list and preparing it for further processing.

In [33]:
```
len(test_data)
```

Out[33]: 252

In [34]:
```
x_input=test_data[152:].reshape(1,-1)
x_input.shape
```

Out[34]: (1, 100)

In [35]:
```
temp_input=list(x_input)
temp_input=temp_input[0].tolist()
```

In [36]:
```
temp_input
```

Out[36]:
```
[0.007063572149344104,
 0.0004693403421491 1324,
 0.0021636589773073936,
 0.006439349494285783,
 0.007077652359608576,
 0.0,
 0.002600145495506072,
 0.0010513223664140106,
 0.0023279280970595895,
 0.00547720179288011,
 0.005045408678102929,
 0.007096425973294539,
```

Demonstrate prediction for next 30 days.

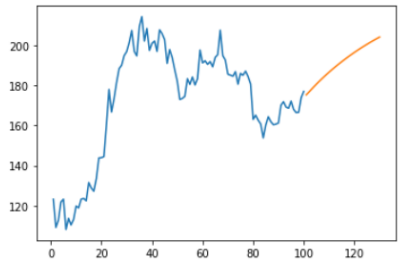
In [37]:
```
# demonstrate prediction for next 30 days
from numpy import array

lst_output=[]
n_steps=100
i=0
while(i<30):

    if(len(temp_input)>100):
        #print(temp_input)
        x_input=np.array(temp_input[1:])
        print("{} day input {}".format(i,x_input))
        x_input=x_input.reshape(1,-1)
        x_input = x_input.reshape((1, n_steps, 1))
        #print(x_input)
        yhat = model.predict(x_input, verbose=0)
        print("{} day output {}".format(i,yhat))
        temp_input.extend(yhat[0].tolist())
        temp_input=temp_input[1:]
        #print(temp_input)
        lst_output.extend(yhat.tolist())
        i=i+1
    else:
        x_input = x_input.reshape((1, n_steps,1))
        yhat = model.predict(x_input, verbose=0)
        print(yhat[0])
        temp_input.extend(yhat[0].tolist())
        print(len(temp_input))
        lst_output.extend(yhat.tolist())
        i=i+1


print(lst_output)
```

Print the output for next 30 days

```
[0.03151016]
101
1 day input [0.00046934 0.00216366 0.00643935 0.00707765 0.         0.00260015
 0.00105132 0.00232793 0.0054772  0.00504541 0.00709643 0.007256
 0.00671157 0.01097787 0.00970596 0.00895032 0.0118837  0.01673198
 0.01679769 0.01705113 0.02448549 0.03275996 0.02748457 0.03056344
 0.03440734 0.03762702 0.03842959 0.04067303 0.04163518 0.04373783
 0.04656795 0.04167273 0.04061671 0.04747378 0.04981578 0.04408983
 0.0470326  0.04189801 0.04353601 0.04410391 0.04166804 0.04671344
 0.04581231 0.04443245 0.03886138 0.04209514 0.04022716 0.03736418
 0.03468425 0.03042264 0.0306667  0.03115481 0.03527562 0.03395677
 0.03568395 0.03380658 0.03527093 0.04199657 0.03897872 0.03948091
 0.0386314  0.03928848 0.03805881 0.04026001 0.04091709 0.04663366
 0.04067773 0.03964987 0.03633633 0.03612043 0.0358623  0.03693239
 0.03399901 0.03651468 0.03609227 0.03704973 0.03576843 0.03402248
 0.02576209 0.02674301 0.02555558 0.02467322 0.02142539 0.02444794
 0.02638162 0.02521766 0.02450426 0.02464506 0.02492197 0.02908033
 0.02989229 0.02865323 0.02836693 0.0300284  0.0281041  0.02733907
 0.02741886 0.03086382 0.03228592 0.03151016]
```

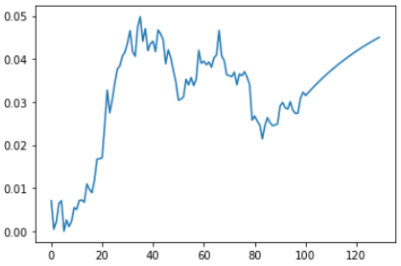
Plot the original data and the predicted values on the same graph

```
In [38]: day_new=np.arange(1,101)
         day_pred=np.arange(101,131)

In [39]: import matplotlib.pyplot as plt

In [40]: len(df1)
Out[40]: 1258

In [42]: plt.plot(day_new,scaler.inverse_transform(df1[1158:]))
         plt.plot(day_pred,scaler.inverse_transform(lst_output))
Out[42]: [<matplotlib.lines.Line2D at 0x1e9461eee20>]
```



Plot the combine data then we can visualize the continuity between the original date and the predicted values

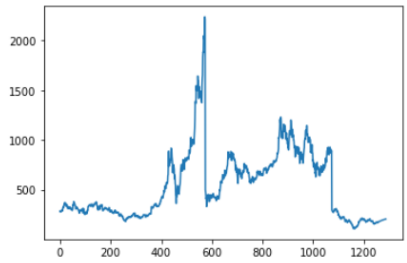
```
In [43]: df3=df1.tolist()
         df3.extend(lst_output)
         plt.plot(df3[1158:])
Out[43]: [<matplotlib.lines.Line2D at 0x1e9462673d0>]
```



Convert the combined data back to its original scale.

We can visualize the original data and the predicted values together, providing a clearer understanding of their relationship and how well the predictions align with actual values.

```
In [44]: df3=scaler.inverse_transform(df3).tolist()

In [45]: plt.plot(df3)

Out[45]: [<matplotlib.lines.Line2D at 0x1e9462bbca0>]
```



6.3 Performance Measure

6.3.1 Google

Epochs	Accuracy		RMSE	
	Train	Test	Train	Test
50	36.00%	97.34%	1864.57	77.42
100	34.21%	96.08%	1916.80	34.21
150	35.79%	95.36%	1870.68	135.07

Table 6.3.1: Epochs for Google Dataset

6.3.2 Apple

Epochs	Accuracy		RMSE	
	Train	Test	Train	Test
50	45.45%	62.71%	217.78	148.89
100	45.50%	62.97%	217.59	147.85
150	46.10%	62.36%	215.18	150.27

Table 6.3.2: Epochs for Apple Dataset

6.3.3 Amazon

Epochs	Accuracy		RMSE	
	Train	Test	Train	Test
50	27.06%	98.50%	2662.18	54.74
100	25.89%	98.32%	2704.82	61.27
150	26.67%	96.92%	2676.23	112.33

Table 6.3.3: Epochs for Amazon Dataset

6.3.4 Tesla

Epochs	Accuracy		RMSE	
	Train	Test	Train	Test
50	66.76%	90.88%	708.30	194.37
100	66.23%	89.80%	719.47	217.31
150	64.56%	87.62%	755.17	263.87

Table 6.3.4: Epochs for Tesla Dataset

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

In this project, we are predicting closing stock price for the next 30 days of any given organization, we developed a model for predicting close stock price using LSTM algorithms for prediction. We have applied datasets belonging to Google, Apple, Microsoft and Tesla Stocks and achieved average 70% accuracy.

7.2 Future Work

- We want to design the web application for predicting stock price.
- We want to add sentiment analysis for better analysis.

Appendix A

Name of Journal: International Journal for Research in Applied Science and Engineering Technology (IJRASET**)**

Certificate:


iJRASET
International Journal for Research in Applied Science & Engineering Technology
ISSN No. : 2321-9653
IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com
ISRA Journal Impact Factor: 7.429
45.98
INDEX COPERNICUS
THOMSON REUTERS
doi
cross ref
TOGETHER WE REACH THE GOAL
SJIF 7.429
Certificate
It is here by certified that the paper ID : IJRASET51757, entitled
Stock Price Prediction using Stacked LSTM
by
Sujan Shaikh
after review is found suitable and has been published in
Volume 11, Issue V, May 2023
in
International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors
Editor in Chief, iJRASET


ISSN No. : 2321-9653
iJRASET
International Journal for Research in Applied Science & Engineering Technology
IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com
ISRA Journal Impact Factor: 7.429
45.98
INDEX COPERNICUS
THOMSON REUTERS
doi
cross ref
TOGETHER WE REACH THE GOAL
SJIF 7.429
Certificate
It is here by certified that the paper ID : IJRASET51757, entitled
Stock Price Prediction using Stacked LSTM
by
Vaishnavi Jadhav
after review is found suitable and has been published in
Volume 11, Issue V, May 2023
in
International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors
Editor in Chief, iJRASET


iJRASET
ISSN No. : 2321-9653
International Journal for Research in Applied Science & Engineering Technology
IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com
ISRA Journal Impact Factor: 7.429
45.98
INDEX COPERNICUS
THOMSON REUTERS
doi crossref
TOGETHER WE REACH THE GOAL
SJIF 7.429
Certificate
It is here by certified that the paper ID : IJRASET51757, entitled
Stock Price Prediction using Stacked LSTM
by
Prof. Balasaheb Gite
after review is found suitable and has been published in
Volume 11, Issue V, May 2023
in
International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors
Editor in Chief, iJRASET


iJRASET
International Journal For Research in
Applied Science and Engineering Technology
INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY
Volume: 11 Issue: V Month of publication: May 2023
DOI:
www.ijraset.com
Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

Stock Price Prediction Using Stacked LSTM

Mohan Badhekar[1], Sujan Shaikh[2], Vaishnavi Jadhav[3], Prof. Balasaheb Gite[4]
[1, 2, 3, 4]*Computer Engineering Department, ISBM College of Engineering Nande, Pune-412115, India*
Savitribai Phule Pune University

Abstract: ***In this project we attempt to implement machine learning approach to predict stock prices. Machine learning is effectively implemented in forecasting stock prices. The objective is to predict the stock prices in order to make more informed and accurate investment decisions. We propose a stock price prediction system that integrates mathematical functions, machine learning, and other external factors for the purpose of achieving better stock prediction accuracy and issuing profitable trades. There are two types of stock trading. You may know of intraday trading by the commonly used term "day trading." Intraday traders hold securities positions from at least one day to the next and often for several days to weeks or months. LSTMs are very powerful in sequence prediction problems because they're able to store past information. This is important in our case because the previous price of a stock is crucial in predicting its future price. While predicting the actual price of a stock is an uphill climb, we can build a model that will predict whether the price will go up or down for next 30 days***

I. INTRODUCTION

The financial market is a dynamic and composite system where people can buy and sell currencies, stocks, equities and derivatives over virtual platforms supported by brokers. The stock market allows investors to own shares of public companies through trading either by exchange or over the counter markets. This market has given investors the chance of gaining money and having a prosperous life through investing small initial amounts of money, low risk compared to the risk of opening new business or the need of high salary career. Stock markets are affected by many factors causing the uncertainty and high volatility in the market. Although humans can take orders and submit them to the market, automated trading systems (ATS) that are operated by the implementation of computer programs can perform better and with higher momentum in submitting orders than any human. However, to evaluate and control the performance of ATSs, the implementation of risk strategies and safety measures applied based on human judgements are required. Many factors are incorporated and considered when developing an ATS, for instance, trading strategy to be adopted, complex mathematical functions that reflect the state of a specific stock, machine learning algorithms that enable the prediction of the future stock value, and specific news related to the stock being analysed.

Time-series prediction is a common technique widely used in many real-world applications such as weather forecasting and financial market prediction. It uses the continuous data in a period of time to predict the result in the next time unit. Many timeseries prediction algorithms have shown their effectiveness in practice. The most common algorithms now are based on Recurrent Neural Networks (RNN), as well as its special type - Long-short Term Memory (LSTM) and Gated Recurrent Unit (GRU). Stock market is a typical area that presents time-series data and many researchers' studies on it and proposed various models. In this project, LSTM model is used to predict the stock price.

II. STOCK MARKET OVERVIEW

Almost every country has one or more stock exchanges, where the shares of listed companies can be sold or bought. It is a secondary market place. When a company first lists itself in any stock exchange to become a public company, the promoter group sells substantial number of shares to public as per government norms. During incorporation of a company shares are bought by promoter groups or institutional investors in a primary market. Once promoter offloads major portion of the shares to public retail investors, then those could be traded in secondary market i.e., in stock exchanges. In India the BSE (Bombay Stock Exchange) and the NSE (National Stock Exchange) are the two most active stock exchange. The BSE has around 5000 listed companies where as NSE had around 1600. Both the exchange has similar trading mechanism and market open time, closing time and settlement process. Stock exchanges helps individual investors to take part in the share market and allows to buy even a single share of some listed company with the help of a trading account and Demat account. These online markets have revolutionized the Indian investment arena along with government initiative like tax benefit on equity investment, National Pension Scheme (NPS) investing in share market etc. Due to continuous reduction in bank interest rates and increasing inflation middle class investors are moving towards equity market from the safe haven of fixed deposits. All these have helped to grow the capitalization of both the exchanges.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

III. RELATED STUDIES

A. Stock Market Prediction Using Machine Learning

The research work done by V Kranthi Sai Reddy Student, ECM, Sreenidhi Institute of Science and Technology, Hyderabad, India. 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. 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. In this context this study uses a machine learning technique called Support Vector Machine (SVM) to predict stock prices for the large and small capitalizations and in the three different markets, employing prices with both daily and up-to-the-minute frequencies

B. Forecasting the Stock Market Index Using Artificial Intelligence Techniques

The research work done by Lufuno Ronald Marwala A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Engineering. The weak form of Efficient Market hypothesis (EMH) states that it is impossible to forecast the future price of an asset based on the information contained in the historical prices of an asset. This means that the market behaves as a random walk and as a result makes forecasting impossible. Furthermore, financial forecasting is a difficult task due to the intrinsic complexity of the financial system. The objective of this work was to use artificial intelligence (AI) techniques to model and predict the future price of a stock market index. Three artificial intelligence techniques, namely, neural networks (NN), support vector machines and neuro-fuzzy systems are implemented in forecasting the future price of a stock market index based on its historical price information. Artificial intelligence techniques have the ability to take into consideration financial system complexities and they are used as financial time series forecasting tools.

C. Stock Price Correlation Coefficient Prediction with ARIMA-LSTM Hybrid Model

The research work done by Hyeong Kyu Choi, B.A Student Dept. of Business Administration Korea University Seoul, Korea. Predicting the price correlation of two assets for future time periods is important in portfolio optimization. We apply LSTM recurrent neural networks (RNN) in predicting the stock price correlation coefficient of two individual stocks. RNN's are competent in understanding temporal dependencies. The use of LSTM cells further enhances its long-term predictive properties. To encompass both linearity and nonlinearity in the model, we adopt the ARIMA model as well. The ARIMA model filters linear tendencies in the data and passes on the residual value to the LSTM model. The ARIMA-LSTM hybrid model is tested against other traditional predictive financial models such as the full historical model, constant correlation model, single-index model and the multi-group model. In our empirical study, the predictive ability of the ARIMA-LSTM model turned out superior to all other financial models by a significant scale. Our work implies that it is worth considering the ARIMALSTM model to forecast correlation coefficient for portfolio optimization.

IV. LSTM ARCHITECTURE

A. An overview of Recurrent Neural Network (RNN)

In a classical neural network, final outputs seldom act as an output for the next step but if we pay attention to a real-world phenomenon, we observe that in many situations our final output depends not only the external inputs but also on earlier output. For example, when humans read a book, understanding of each sentence depends not only on the current list of words but also on the understanding of the previous sentence or on the context that is created using past sentences. Humans don't start their thinking from scratch every second.

As you read this essay, you understand each word based on your understanding of previous words. This concept of 'context' or 'persistence' is not available with classical neural networks. Inability to use context-based reasoning becomes a major limitation of traditional neural network. Recurrent neural networks (RNN) are conceptualized to alleviate this limitation are networked with feedback loops within to allow persistence of information. The Figure 1Error! Reference source not found. shows a simple RNN with a feedback loop and its unrolled equivalent version side by side.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

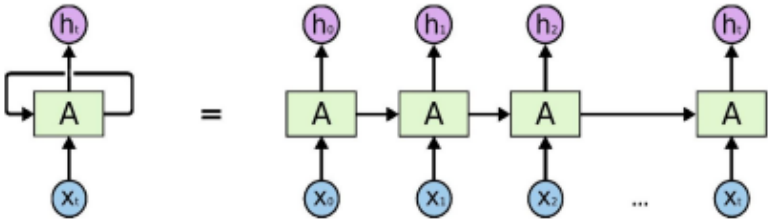


Figure 1: An unrolled recurrent neural network

Initially (at time step t) for some input Xt the RNN generates an output of ht. In the next time step (t+1) the RNN takes two input Xt+1 and ht to generate the output ht+1. A loop allows information to be passed from one step of the network to the next. RNNs are not free from limitations though. When the 'context' is from near past it works great towards the correct output. But when an RNN has to depend on a distant 'context' (i.e., something learned long past) to produce correct output, it fails miserably. This limitation of the RNNs was discussed in great detail by Hochreiter [8] and Bengio, et al. [9]. They also traced back to the fundamental aspects to understand why RNNs may not work in long-term scenarios. The good news is that the LSTMs are designed to overcome the above problem.

B. LSTM Networks

Hochreiter & Schmidhuber introduced a special type of RNN which is capable of learning long term dependencies. Later on, many other researchers improved upon this pioneering work. LSTMs are perfected over the time to mitigate the long-term dependency issue. The evolution and development of LSTM from RNNs are explained in given references. Recurrent neural networks are in the form of a chain of repeating modules of the neural network. In standard RNNs, this repeating module has a simple structure like a single tanh layer as shown in Figure 2

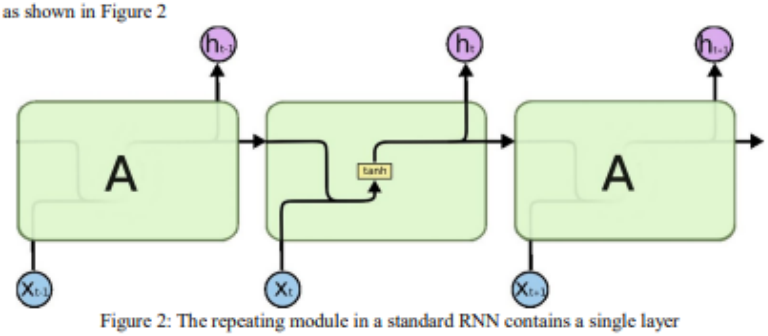


Figure 2: The repeating module in a standard RNN contains a single layer

LSTMs follow this chain-like structure; however, the repeating module has a different structure. Instead of having a single neural network layer, there are four layers, interacting in a very special way as shown in Figure 3.

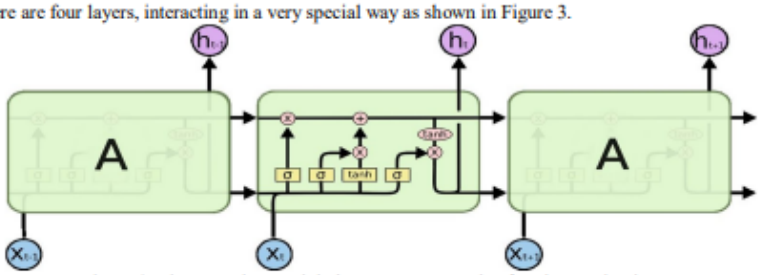


Figure 3: The repeating module in an LSTM contains four interacting layers

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

In Figure 3, every line represents an entire feature vector, from the output of one node to the inputs of others. The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers. Lines merging denote concatenation, while a line forking denotes its content being copied and the copies going to different locations.

C. The Working of LSTM

LSTM is a special network structure with three "gate" structures. Three gates are placed in an LSTM unit, called input gate, forgetting gate and output gate. While information enters the LSTM's network, it can be selected by rules. Only the information conforms to the algorithm will be left, and the information that does not conform will be forgotten through the forgetting gate.

1) Forget Gate

A forget gate is responsible for removing information from the cell state.

- The information that is no longer required for the LSTM to understand things or the information that is of less importance is removed via multiplication of a filter.
- This is required for optimizing the performance of the LSTM network.
- This gate takes in two inputs; h_t-1 and x_t. h_t-1 is the hidden state from the previous cell or the output of the previous cell and x_t is the input at that particular time step.

2) Input Gate

- Regulating what values need to be added to the cell state by involving a sigmoid function. This is basically very similar to the forget gate and acts as a filter for all the information.
- Creating a vector containing all possible values that can be added (as perceived from h_t-1 and x_t) to the cell state. This is done using the tanh function, which outputs values from -1 to +1.
- Multiplying the value of the regulatory filter (the sigmoid gate) to the created vector (the tanh function) and then adding this useful information to the cell state via addition operation.

3) Output Gate

The functioning of an output gate can again be broken down to three steps:

- Creating a vector after applying tanh function to the cell state, thereby scaling the values to the range -1 to +1.
- Making a filter using the values of h_t-1 and x_t, such that it can regulate the values that need to be output from the vector created above. This f filter again employs a sigmoid function.
- Multiplying the value of this regulatory filter to the vector created in step 1, and sending it out as a output and also to the hidden state of the next cell.

V. FRAMEWORK TO FORECAST DIFFERENT SHARES PRICE

In this section, we shall first analyze some existing techniques and their merits to finally arrive at our methodology. Next, we shall discuss the algorithmic and implementation steps in detail.

A. Analyzing Different Methods

The prediction methods can be roughly divided into two categories, statistical methods and artificial intelligence methods. Statistical methods include logistic regression model, ARCH model, etc. Artificial intelligence methods include multi-layer perceptron, convolutional neural network, naive Bayes network, back propagation network, single-layer LSTM, support vector machine, recurrent neural network, etc. They used Long short-term memory network (LSTM).

For time series data, such as text, signals, stock prices, etc. LSTM is better suited to learn temporal patterns in deep neural networks. An LSTM solves the 'vanishing gradient' problem that exists in a RNN while learning long-term dependencies with time series dataset with the use of memory cell (states) and (input and forget) gates. So, LSTM may be a better option for future prediction of the company's share price as well as growth.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

B. Methodology

Used Long Short-term Memory (LSTM) with embedded layer and the LSTM neural network with automatic encoder.
LSTM is used instead of RNN to avoid exploding and vanishing gradients.
In this project python is used to train the model, MATLAB is used to reduce dimensions of the input.
The historical stock data table contains the information of opening price, the highest price, lowest price, closing price, transaction date, volume and so on.

C. Implementation Steps

1) *Step1: Data Collection:* Day-wise past stock prices of selected companies are collected from the Tiingo's official website.
2) *Step2: Data Pre-processing*: This step incorporates the following:
a) Data Transformation: Normalization
b) Data Cleaning: Fill in missing values.
c) 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. Creating a data structure with 60 timesteps and 1 output.

3) *Step3: Feature Selection:* In this step, data attributes are chosen that are going to be fed to the neural network. In this study Closing Price is chosen as selected feature.
4) *Step 4: Train the model:* The model is trained by feeding the training dataset. The model is initiated using random weights and biases. Proposed LSTM model consists of a sequential input layer followed by 3 LSTM layers and then a dense layer with activation. The output layer again consists of a dense layer with a linear activation function.
5) *Step5: Output Generation:* The generated output is compared with the target values and error difference is calculated. The Backpropagation algorithm is used to minimize the error difference by adjusting the biases and weights of the neural network.
6) *Step 6: Test Dataset Update*: Step 2 is repeated for the test data set.
7) *Step 7: Error Calculation*: By calculating deviation we check the percentage of error of our prediction with respect to actual price.
8) *Step 8: Visualization*: Using Keras and their function APIs the prediction is visualized.

```
# LSTM
Inputs: dataset
Outputs: RMSE of the forecasted data
# Split dataset into 80% training and 20% testing data
size = length(dataset) * 0.80
train = dataset [0 to size]
test = dataset [size to length(dataset)]
# Procedure to fit the LSTM model
Procedure LSTMAlgorithm (train, test, train_size, epochs)
X_train = train
Y_train = test
model = Sequential ()
model.add(LSTM(50,return_sequences=True,input_shape=(100,1)))
model.add(LSTM(50,return_sequences=True))
model.add(LSTM(50))
model.add(Dense(1))
model.compile(loss='mean_squared_error',optimizer='adam')
model.fit(X_train, Y_train, validation_data=(X_test.ytest),epoch=100,batch_size=64,versbose=1)
return model

# Procedure to make predictions
Procedure getPredictonsFromModel (model, X)
```

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

```
predictions = model.predict(X)
return predictions
epochs = 100
neurons = 50
predictions
# Fit the LSTM model
model = LSTMAlgorithm (train, epoch, neurons)

# Make predictions
pred = model.predict(train)

# Validate the model
n = len(dataset)
error = 0
for i in range(n): error += (abs(real[i] - pred[i])/real[i]) * 100
 accuracy = 100 - error/n
```

VI. PERFORMANCE MEASURE

A. Google

Epochs	Accuracy		RMSE	
	Train	Test	Train	Test
50	36.00%	97.34%	1864.57	77.42
100	34.21%	96.08%	1916.80	34.21
150	35.79%	95.36%	1870.68	135.07

Table 1: Epochs for Google Dataset

B. Apple

Epochs	Accuracy		RMSE	
	Train	Test	Train	Test
50	45.45%	62.71%	217.78	148.89
100	45.50%	62.97%	217.59	147.85
150	46.10%	62.36%	215.18	150.27

Table 2: Epochs for Apple Dataset

C. Amazon

Epochs	Accuracy		RMSE	
	Train	Test	Train	Test
50	27.06%	98.50%	2662.18	54.74
100	25.89%	98.32%	2704.82	61.27
150	26.67%	96.92%	2676.23	112.33

Table 3: Epochs for Amazon Dataset

D. Tesla

Epochs	Accuracy		RMSE	
	Train	Test	Train	Test
50	66.76%	90.88%	708.30	194.37
100	66.23%	89.80%	719.47	217.31
150	64.56%	87.62%	755.17	263.87

Table 4: Epochs for Tesla Dataset

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

VII. CONCLUSIONS

In this project, we are predicting closing stock price for the next 30 days of any given organization, we developed a model for predicting close stock price using LSTM algorithms for prediction. We have applied datasets belonging to Google, Apple, Microsoft and Tesla Stocks and achieved average 60% accuracy for training and average 80% accuracy for test datasets.

REFERENCES

[1] Stock Price Prediction Using LSTM on Indian Share Market by Achyut Ghosh, Soumik Bose1, Giridhar Maji, Narayan C. Debnath, Soumya Sen
[2] S. Selvin, R. Vinayakumar, E. A. Gopalkrishnan, V. K. Menon and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in International Conference on Advances in Computing, Communications and Informatics, 2017.
[3] Murtaza Roondiwala, Harshal Patel, Shraddha Varma, "Predicting Stock Prices Using LSTM" in Undergraduate Engineering Students, Department of Information Technology, Mumbai University, 2015.
[4] Xiongwen Pang, Yanqiang Zhou, Pan Wang, Weiwei Lin, "An innovative neural network approach for stock market prediction", 2018
[5] Ishita Parmar, Navanshu Agarwal, Sheirsh Saxena, Ridam Arora, Shikhin Gupta, Himanshu Dhiman, Lokesh Chouhan Department of Computer Science and Engineering National Institute of Technology, Hamirpur – 177005, INDIA - Stock Market Prediction Using Machine Learning.
[6] Pranav Bhat Electronics and Telecommunication Department, Maharashtra Institute of Technology, Pune. Savitribai Phule Pune University - A Machine Learning Model for Stock Market Prediction.
[7] Anurag Sinha Department of computer science, Student, Amity University Jharkhand Ranchi, Jharkhand (India), 834001 - Stock Market Prediction Using Machine Learning.
[8] V Kranthi Sai Reddy Student, ECM, Sreenidhi Institute of Science and Technology, Hyderabad, India - Stock Market Prediction Using Machine Learning.
[9] Asset Durmagambetov currently works at the mathematics, CNTFI. Asset does research in Theory of Computation and Computing in Mathematics, Natural Science, Engineering and Medicine. Their current project is 'The Riemann Hypothesis-Millennium Prize Problems' - stock market predictions.
[10] Mariam Moukalled Wassim El-Hajj Mohamad Jaber Computer Science Department American University of Beirut - Automated Stock Price Prediction Using Machine Learning.


doi
crossref
10.22214/IJRASET
INDEX COPERNICUS
45.98
JISRA F
IMPACT FACTOR:
7.129
TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429
ISI
INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY
Call : 08813907089 (24*7 Support on Whatsapp)

REFERENCES

Datasets: Tiingo API(trading data)

[1] Stock Price Prediction Using LSTM on Indian Share Market by Achyut Ghosh, Soumik Bose1, Giridhar Maji, Narayan C. Debnath, Soumya Sen

[2] S. Selvin, R. Vinayakumar, E. A. Gopalkrishnan, V. K. Menon and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in International Conference on Advances in Computing, Communications and Informatics, 2017.

[3] Murtaza Roondiwala, Harshal Patel, Shraddha Varma, “Predicting Stock Prices Using LSTM” in Undergraduate Engineering Students, Department of Information Technology, Mumbai University, 2015.

[4] Xiongwen Pang, Yanqiang Zhou, Pan Wang, Weiwei Lin, “An innovative neural network approach for stock market prediction”, 2018

[5] Ishita Parmar, Navanshu Agarwal, Sheirsh Saxena, Ridam Arora, Shikhin Gupta, Himanshu Dhiman, Lokesh Chouhan Department of Computer Science and Engineering National Institute of Technology, Hamirpur – 177005, INDIA - Stock Market Prediction Using Machine Learning.

[6] Pranav Bhat Electronics and Telecommunication Department, Maharashtra Institute of Technology, Pune. Savitribai Phule Pune University - A Machine Learning Model for Stock Market Prediction.

[7] Anurag Sinha Department of computer science, Student, Amity University Jharkhand Ranchi, Jharkhand (India), 834001 - Stock Market Prediction Using Machine Learning.

[8] V Kranthi Sai Reddy Student, ECM, Sreenidhi Institute of Science and Technology, Hyderabad, India - Stock Market Prediction Using Machine Learning.

[9] Asset Durmagambetov currently works at the mathematics, CNTFI. Asset does research in Theory of Computation and Computing in Mathematics, Natural Science, Engineering and Medicine. Their current project is 'The Riemann Hypothesis-Millennium Prize Problems' - stock market predictions.

[10] Mariam Moukalled Wassim El-Hajj Mohamad Jaber Computer Science Department American University of Beirut - Automated Stock Price Prediction Using Machine Learning.

[11] Manh Ha Duong Boriss Siliverstovs June 2006 - The Stock Market and Investment.

[12] Dharmaraja Selvamuthu , Vineet Kumar and Abhishek Mishra Department of Mathematics, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India - Indian stock market prediction using artificial neural networks on tick data.

[13] Lufuno Ronald Marwala A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Engineering - Forecasting the Stock Market Index Using Artificial Intelligence Techniques.

[14] Xiao-Yang Liu1 Hongyang Yang,Qian Chen4,Runjia ZhangLiuqing Yang Bowen Xiao Christina Dan Wang Electrical Engineering, 2Department of Statistics, 3Computer Science, Columbia University, 3AI4Finance LLC., USA, Ion Media Networks, USA, Department of Computing, Imperial College, 6New York University (Shanghai) - A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance.

[15] Pushpendu Ghosh, Ariel Neufeld, Jajati Keshari SahooDepartment of Computer Science & Information Systems, BITS Pilani K.K.Birla Goa campus, India bDivision of Mathematical Sciences, Nanyang Technological University, Singapore cDepartment of Mathematics, BITS Pilani K.K.Birla Goa campus, India - Forecasting directional movements of stock prices for intraday trading using LSTM and random forests.

[16] Xiao Ding, Kuo Liao, Ting Liu, Zhongyang Li, Junwen Duan Research Center for Social Computing and Information Retrieval Harbin Institute of Technology, China - Event Representation Learning Enhanced with External Commonsense Knowledge.

[17] Huicheng Liu Department of Electrical and Computer Engineering Queen's University, Canada - Leveraging Financial News for Stock Trend Prediction with Attention-Based Recurrent Neural Network.

[18] Hyeong Kyu Choi, B.A Student Dept. of Business Administration Korea University Seoul, Korea = Stock Price Correlation Coefficient Prediction with ARIMA-LSTM Hybrid Model.

[19] M. Nabipour Faculty of Mechanical Engineering, Tarbiat Modares University, 14115-143 Tehran, Iran; Mojtaba.nabipour@modares.ac.ir - Deep Learning for Stock Market prediction.

[20] Lavanya Ra SRM Institute of Science and Technology | SRM · Department of

Computer Science - Stock Market Prediction.

[21] M. Mekayel Anik · M. Shamsul Arefin (B) Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chittagong, Bangladesh - An Intelligent Technique for Stock Market Prediction.