Stock Market Price Prediction using Long Short-Term Memory Neural Networks

A dissertation report submitted in fulfillment of the requirements for the degree of Bachelor of Technology.

by

Yukti Khurana (2017UCP1234)
Prakhar Jain (2017UCP1543)
Sourabh Yadav (2017UCP1554)

Supervisor- Dr. Girdhari Singh

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY
APRIL, 2021

Abstract

Stock prediction refers to forecasting the future value or movement of a company stock, a specific sector, or the whole market. It is one of the most challenging problems intriguing computer experts throughout the world today. Uncertain factors like interest rates, politics, and fluctuations in the economy greatly affect stock market volatility, thereby, making accurate predictions very difficult. A successful prediction can, however, be equally rewarding in the form of huge profits in the trading arena. Over the years, a variety of methods of predicting future prices of stocks have been researched. They often involve studying stock price change over time to accurately identify movement patterns. Deep learning has been increasingly being used for decoding such patterns to forecast market trends. This project explores the efficacy of such deep neural networks in the prediction of the general market index as well as stocks of individual companies. However, besides the historical stock patterns, another big factor that has a huge impact on the stock market is news. Financial news indicating corporate acquisitions and mergers, the announcement of new products, and economic plunges can translate into significant changes in company stock prices. Therefore, instead of using the traditional approach of simply training a neural network on historical stock data, this research project delves into discovering the effectiveness of incorporating financial news sentiment along with the deep neural network. Firstly, it explores two deep neural networks namely, simple Recurrent Neural Networks and Long Short-Term Memory neural networks for predicting the Adjusted close prices of a variety of stocks. Both the neural networks are then combined with financial news sentiment scores to consider the effect of changing world news on the stock market and increase the efficiency of the prediction. The biggest challenge of incorporating news sentiment for prediction is the lack of appropriate datasets to train the model. In the course of this project, both the models were trained on 16 datasets of entirely different kinds, out of which 15 were created by us while one dataset [1] created by the researchers in this field was used. One familiar dataset was used to ease the comparison of our proposed model with previously proposed solutions. The 15 new datasets served to further enhance efficiency and properly train neural networks so that the effect of the news on stock prices can be observed. The datasets were formed by combining financial news with the stock data of different companies and market indexes over several years. Thereafter, the concept of differential privacy was used which involved doping the datasets by adding some noise to mask the authors' biases prevalent in news headlines. This was done to improve the efficiency of stock prediction. This project thus aims to show a multi-way comparison between LSTM and simple RNN as well as between the performance of three different variations of each of these models. It was observed that the incorporation of financial news sentiment gave significantly better results for both RNN and LSTM. This performance was further enhanced when the differential privacy concept was intertwined with both the neural networks. The LSTM model outperformed the RNN model with a **0.388%** increase in mean performance accuracy, **42.39%** decrease in mean absolute error and **67.52%** decrease in mean squared error which was calculated based on their cumulative performance over all datasets. The LSTM using news sentiment scores outperformed traditional LSTM models by a **0.19%** increase in accuracy and a **21.41%** decrease in mean absolute error. Further, the use of differential privacy to this LSTM model achieved an improvement of **0.265%** in accuracy and **30.841%** in mean absolute error as compared to the general approach. The observations prove that the combination of LSTM with news sentiment and differential privacy beats all other variations by significant improvements in prediction accuracy and error, thereby it emerges as the ultimate winner in stock prediction.

Acknowledgments

Firstly, we would like to express our sincerest gratitude to our project supervisor Dr. Girdhari Singh for providing his invaluable guidance, comments, and suggestions throughout this project on the topic “Stock Market Price Prediction using Long Short Term Memory Neural Networks” for the partial fulfillment of the requirements leading to the award of Bachelor of Technology in Computer Science and Engineering degree. We would also like to especially thank Ms. Monu Verma for constantly motivating us to work harder and helping us make this project a reality. We extend our hearty thanks to our Head of Department Dr. Emanual Pilli and the department of computer science of MNIT, Jaipur for encouraging us and giving us the opportunity and resources for the completion of this project. Lastly, we would like to thank all the faculty and professors of our institute - Malaviya National Institute of Technology, Jaipur for their constant guidance and support to complete this project.

Yukti Khurana	Sourabh Yadav	Prakhar Jain
(2017UCP1234)	(2017UCP1554)	(2017UCP1543)

Contents

List of Figures

List of Tables

Chapter 1-Introduction

Stock price prediction is a very important aspect of the financial world due to its indispensability in evaluating effective strategies for stock exchange transactions, especially for investment firms. Stock trends are highly volatile due to economic and technological factors like inflation, interest rates, and financial news. In 2020-21 due to the covid-19 pandemic and recent Suez Canal Blockage, the tumultuous world scenario has had a drastic impact on stock markets throughout the globe and is a testament to the erratic nature of the stock market and its intrinsic dependence on world events. Such events and their long- and short-term effect on the stock market are needed to be carefully assessed by traders and investors by quantitative analysis of huge amounts of data, mostly in the form of market indices, company charts, newspapers, blogs, recent trends, and textual information, and therefore accurate prediction becomes a highly cumbersome as well as erroneous task. Analyzing huge volumes of such unstructured and non-uniform data from unreliable sources and inspecting their bearing on the market trends is becoming increasingly time-consuming and difficult and ultimately boils down to humane instincts which can be a risky proposition. Thus, artificial intelligence approaches like machine learning and deep learning are being investigated to automatically predict stock trends and prices and reduce the burden on investors.

After the success of classical machine learning algorithms, the emergence of deep learning has given new and effective models capable of analyzing these non-stationary, non-linear, and noise-prone data, while giving the advantage of cheaper computation. Therefore, in this project, we have worked to provide a more reliable solution for accurate stock prediction by employing the use of deep learning models like Recurrent neural networks which are capable of handling long time series data and tested them on a variety of different kinds of datasets. We used our proposed models on time-series SP500 market index as well as stock data of 7 globally known companies. RNN's can be computationally fast but have short-term memory and memory-gradient problems for long sequence data. Consequently, we explored an LSTM-based deep neural network for predicting and forecasting Adjusted close prices and observed a significant improvement in accuracy and prediction errors. To explore the role of financial news in stock price prediction, we used sentiment analysis to extract useful insights from news headlines and articles. Opinion mining helps uncover if articles or blogs are expressing a positive or negative emotion or sentiment about the financial market and to get fundamental insights about future market trends. VADER sentiment analyzer was used to extract sentiment compound (normalized) scores, as this sentiment analyzer is sensitive to both polarity and intensity of emotion. It can not only work with multiple domains but has also been found to be quite good at dealing with news articles and textual information on social media platforms. These extracted scores were used as input for the models to relay the effect of changing world news on the stocks of the global companies.

However, Financial news reports are not always reliable and objective sources about events and might be inflicted with the author's bias or personal opinion, often making prediction faulty. To address this issue and boost the prediction model's robustness, a Differential privacy model [1] was employed which included adding random noise to the sentiment scores. DP has recently emerged as a promising approach for providing individual privacy to data being used on public platforms. It is based on the idea that a resulting outcome is approximately as likely to come irrespective of whether a data point opts in or out of the dataset i.e., any specific individual data point cannot largely affect the overall outcome. [1] In simple terms, Differential privacy refers to a system through which group pattern information about a dataset can be given publicly but individual information is withheld. For instance, noise using Gaussian or Laplace distribution can be injected, thereby, masking the contents or the biases of any feature's row. A DP-RNN and DP-LSTM were therefore used to increase the accuracy of the model and decrease stock prediction errors and this method showed a significant increase in performance overall 16 datasets used, as compared to traditional approaches.

Chapter 2- Important Terms and Concepts

2.1 Terms related to deep neural networks

2.1.1 RNN

RNNs are a robust and novel kind of neural network which are becoming increasingly popular for solving a variety of problem domains due to their unique advantage of being the only type equipped with internal memory. This memory gives them an edge over other neural networks by helping them to decode and retain important parts in the input that they receive, resulting in a very accurate prediction. They have now become a highly preferred algorithm for handling problems that require the processing of sequential or time series data like video, finance articles, audio, text, articles, speech, weather, and much more. Moreover, they are capable of forming a much deeper discernment of series data and their context in comparison to other models and techniques.[19] They were derived from simple feed-forward neural networks and show a behavior similar to that of the human brain.

2.1.2 LSTM

LSTM is the short form of Long Short-Term Memory Networks, which are a type of recurrent memory network proficient in understanding order dependence in time sequence prediction or regression problems. They operate on a wide range of problem domains namely translating machine code, recognizing speech, prediction of series data, music composition, rhythm learning, human action recognition, and many more. They have given exceptional performance in these fields due to their novel property of selectively remembering patterns for long stretches of time which other kinds of networks cannot do. Besides processing single data points, they are also adept at handling entire sequences of data at a time. These characteristics make them suitable for the prediction, classification as well as processing of time series data where uncertain lags between significant events are quite common.

2.1.3 Sigmoid

This function is defined as a monotonic mathematical function known for its distinctive “S”-shaped curve, which is known as the “sigmoid curve”. A sigmoid function is powerful and capable of mapping the complete number line into a very small range of numbers, for example between -1 to 1. Such small ranges are quite helpful as they can be interpreted as a probability. They have a bell-shaped first derivative and these functions were first inspired by the activation potential in biological neural networks. They are commonly used as activation functions in artificial neural networks. They are often placed as the last layer of a machine learning model that helps to output into a probability score, which can be simpler to understand and use. [22]

2.1.4 ReLU Layer

ReLU refers to the Rectified Linear Unit, which is a type of activation function present in a neural network, used to convert the summed weighted input from the node into the activation of the node or output for that input. These piecewise linear functions simply output the input directly if it is positive or return zero as output in case the input is negative. Since they can be easily trained and give better performance in most cases, they are popularly used in a variety of neural networks. Other activation functions like sigmoid or hyperbolic tangent functions often fail in networks that require a large number of layers.[21] However, since the ReLU function is able to overcome the problem of vanishing gradient, they can be employed in such networks. It can be defined as -

$$f(x) = \max(0, \text{input})$$

2.1.5 Dropout Layer

Overfitting is a common problem faced by deep neural networks which hamper their performance on new datasets as this causes high generalization error. The main objective of machine learning and deep learning models is to give a generalized model. One way to solve this problem is the approach of regularization, which reduces model overfitting by adding a penalty or fine to the model's loss function. This prevents the model undergoing training from learning an interdependent set of weights of the features. "Dropout" is a regularization method commonly used to reduce this interdependent learning amongst the neurons. As the name suggests, this layer drops out hidden and visible units in the network. This method is a remarkable way of effective regularization to avoid model overfitting and reduce generalization error. It randomly sets the outgoing edges of hidden units to 0 with a frequency rate at each update during the training phase. The inputs which are not zero are scaled up by the reciprocal of (1-rate) such that the sum over all inputs is not changed. [26]

2.1.6 Dense Layer

Dense layers are the most common and frequently used layer in a neural network. Their non-linearity property helps them model all mathematical functions, making them very useful for neural networks. A dense layer is one in which all the neurons receive input from all the neurons of the previous layer i.e., it is a completely connected layer. It is the most basic layer which often follows LSTM layers and is used for outputting a prediction. It applies the following on the input layer to return the output-

output = activation_function(dot_product(input_data, kernel/weights) + bias)

2.2 Model evaluation terms

For Performance evaluation of the models used for stock price prediction, the following four evaluation measures have been employed.

2.2.1 Mean Absolute Error (MAE)

It is a model evaluation technique which measures the average magnitude of error in prediction performed by the trained model. It does not consider their direction and simply takes the absolute difference between predicted and real values given all the individual differences have the same weight.[10] Since all the individual scores are equally weighted in this average, it is called a linear score and conveys how far the model predictions are from the real values.

$$MAE = \frac{1}{n}\sum_{j=1}^{n} |y_j - \bar{y_j}|$$

2.2.2 Mean Squared Error (MSE)

MSE is another model evaluation method which is commonly used with regression problems. It can be defined as the average of the square of prediction errors for all instances of the test dataset. The prediction error refers to the difference of predicted and real value for each data point. It is very easy to compute as compared to mean absolute error which requires complex linear programming tools for gradient computation.

$$MSE = \frac{1}{n}\sum_{j=1}^{n} \left(y_j - \bar{y_j}\right)^2$$

2.2.3 Root Mean Squared Error (RMSE)

This quadratic scoring method helps in measuring the mean magnitude of error of the prediction model.

It can be defined as the square root of the average squared distances between real and predicted values.[10] In other words, it is the square root of mean squared error. First the difference of prediction and actual observations for each instance of test data are squared and then averaged over the entire sample space. At last, root of this mean is taken to get root mean squared error.

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(y_j - \bar{y}_j\right)^2}$$

2.2.4 Mean Prediction Accuracy (MPA)

Higher the MPA of the model, the better it is at predicting the stock prices.

$$MPA = 1 - \frac{1}{n}\sum_{j=1}^{n}\frac{|y_j - \bar{y}_j|}{y_j}$$

Note- In the above formulae 1 to 4, y_j refers to real stock price value for each testing day and $\bar{y}_j$ refers to predicted stock price value. The smaller the value of MAE, MSE, and RMSE, the better is the model. On the other hand, higher accuracy means better model performance.

2.3 Terms relating to data processing

2.3.1 Data Anonymization

Anonymization of data refers to the procedure of erasing or encrypting an individuals' identifiers present in data that needs to be protected or contains sensitive information. For instance, personal information that can identify a person like names, addresses, phone numbers, emails, and social security numbers can be removed through a data anonymization process which holds on to the data but keeps the sources of information anonymous or unidentified.[17] Yet, even when the data is cleared off such identifiers, malicious attackers or crackers are able to use a variety of de-anonymization techniques to backtrack this anonymization process and get the original sources of information compromising the security of millions of people. Since the data usually passes through many different sources which might be public or openly available, these hackers can use linkage attacks or cross-reference techniques to reveal sensitive and private data.

2.3.2 Differential Privacy

Differential privacy is a novel approach for data security which can be defined as a procedure of sharing data publicly in the form of patterns present in it, while hiding information about any individual. This concept is based on the principle that if a random single and small substitution is done in a dataset, then a query result cannot give any inference about any single person that has participated in the creation of that dataset. For example, government organizations are increasingly using such differentially private algorithms to analyze statistical or demographic information while ensuring the confidentiality of individual responses given by the citizens. Companies openly collect data about consumer behavior to create better products and services, but also ensure that any particular user behavior is not revealed to even the internal analysts of the company. [18]

2.3.3 Plausible Deniability

Plausible Deniability can be defined as a formal guarantee of privacy notably for releasing sensitive

datasets: an output record can be released only if a certain input record is indistinguishable, up to a privacy parameter [21]. It refers to the condition wherein one can safely repudiate knowledge of any truth that might exist as the subject is deliberately made unaware of this truth. This indicates that a synthetic dataset obtained by applying the concept of differential privacy employing techniques like proper randomization would have a similar statistical structure and format as the original dataset, however, the plausible deniability property of differential privacy algorithms guarantees that an adversary would not be able to access any single individual's private or sensitive data, thereby protecting the privacy of that individual.

2.4 Terms related to information extraction

2.4.1 Sentiment Analysis

This is a text analysis procedure in natural language processing that helps in determining the polarity within a text which can be in the form of a complete document, a paragraph, a sentence, or a clause. This process is also known as "opinion mining". The polarity of any text can be defined as the kind of tone expressed or conveyed through it, which might be categorized as positive, neutral, or negative, about the topic under discussion. This text analysis process can be used to measure the opinion, attitude, emotions, views, or sentiments of the writer which are obtained by various textual computation algorithms that aim to decipher the subjectivity present in said text.[15]

2.4.2 VADER

Formally referred to as Valence Aware Dictionary and Sentiment Reasoner, this opinion mining tool is a rule-based and lexicon sentiment analyzer that uses a list of lexical features (e.g., words). These lexicons are usually tagged in accordance to their semantic orientation i.e., either positive, negative, or neutral.[16] It not only indicates the subjective or contextual tone expressed by a piece of text but also conveys the intensity of its positive or negative nature i.e., the degree of the text's semantic nature.

2.4.3 Compound Score

The compound score given by the VADER sentiment analyzer is calculated using the positive, negative, and neutral scores assigned by VADER to a text. It is essentially the normalized sum of these three types of valence scores, modified in accordance with the text analysis rules. The compound score lies in the range of negative one to positive one indicating negative and positive tone, respectively. The nearer the compound score is to the negative one, the more positive sentiment or opinion is expressed by the text. The compound score is usually considered as the most useful metric for obtaining the closest single unidimensional measure of the sentiment of the given piece of text. The following formula is used to calculate the compound score –

$$\bar{s} = \frac{s}{\sqrt{s^2 + \alpha}}, \quad \bar{s} \in [-1,1]$$

$\bar{s}$ is the compound score and, s is the sum of valence/ sentiment scores of all words, and α is the Normalization constant, with a default value of 15. [20]

Chapter 3 - Literature Survey

3.1 Deep Learning for Stock Prediction

It is a fact that neural networks were inspired by the working of the human brain. The high-level concept is that neurons of a neural network learn to recognize patterns just like a human brain. [6] But we cannot do even the simplest, seemingly mundane tasks like reading a page of a book or understanding a movie, without the ability of our brain to retain and use previous information, i.e., our brain does not start thinking from scratch; it accounts for our previous experiences and learnings. For instance, understanding previous words in a sentence to make logical sense of the upcoming word. This means that our thoughts have continuity or persistence. However, the biggest disadvantage of traditional neural networks is that they fail to consider the past. They pass the information in a single direction. The neural networks that can solve this problem are called recurrent neural networks, which are increasingly being used to solve a myriad of problems. They allow information to persist thereby being very popular in handling sequential data. Since stock data is a type of time series data, such neural networks become an ideal choice for stock price prediction. This project has broadly used the following three types of Recurrent neural networks for prediction adjusted close prices.

3.1.1 RNN- Simple Recurrent Neural Networks

This is a very robust and strong kind of artificial neural network, providing a promising approach to handle many problem domains due to their ability of retaining and using the past with the help of their internal memory. It is the first kind of neural network that remembers its input making it suitable for time series data. They precisely predict the future by remembering important aspects of input data. Information cycles through a loop in an RNN which helps it in considering the current input as well as what it has learned from the previous inputs for deciding. The RNN architecture appears to be multiple copies of a single traditional neural network where each single component passes a message to the next.

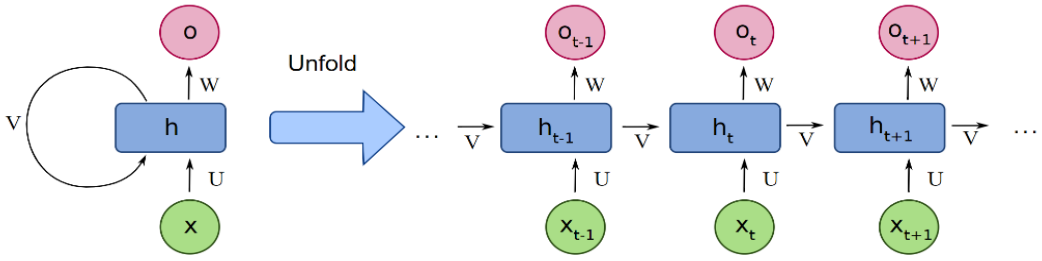


Figure 1. Unrolled form of a RNN showing a chain-like sequence of loops, suitable to handle lists and sequences. [7] Here x is the input, o is the output, h is the RNN block containing weights and activation functions, and V refers to communication from one time step to another.

3.1.2 RNN- Limitations

RNNs have a chain-like structure consisting of repeating modules of single neural networks. Each of these repeating modules have a simple structure in the form of a single tan(h) layer.

Due to their unique structure, they have been incredibly successful in dealing with different problems; however, they do suffer from some limitations of their own –

1. **Exploding Gradients** – When a lot of importance is assigned to weights during model training, they keep accumulating as we go down the model and ultimately result in huge updates to the model.

2. **Vanishing Gradients** – At each step of the model training process in RNN, the process of backpropagation takes places where the gradient is evaluated which helps in updating the network weights. The previous layer decides how much the current layer is affected in terms of the value of the gradient i.e., in case the previous layer's calculated gradient is small, the gradient of the current one would become much smaller. In this way, if the gradients keep shrinking exponentially as the backpropagation is done, virtually no weight updation will take place. This will result in the model either not learning anything or taking too long to learn. So, if the network is unable to learn the effect of former input, the model is said to be suffering from a short-term memory problem. This is a very undesirable situation. [8]

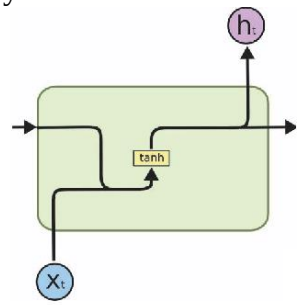


Figure 2. RNN Basic Architecture [8]

RNNs are quite successful for handling problems that require looking only at the recent past, to perform the present task. However, many times we require the knowledge and context of a distant past. E.g., a long sentence of words having multiple phrases might require the understanding of words that occurred much earlier than the current word that needs to be predicted. But RNNs commonly fail to use information when the interval between two important events is very large, which is often the case with time series data. This is because RNNs find it difficult to preserve significant information over many steps. The hidden state in a simple RNN keeps getting rewritten is constantly being rewritten. The solution to this problem is long short-term memory networks.

3.1.3 Long Short-Term Memory Networks

LSTMs are a unique kind of recurrent neural networks, applicable to complex problem domains and have become quite popular in recent years due to their ability to prevent long-term dependency problems usually faced by vanilla RNNs and other neural networks. This means that they can remember information for long periods using their unique structure. Their chain-like structure of repeating modules is quite similar to vanilla RNNs; however, their modules differ in their internal structure. They have four interaction layers in each module instead of a single one. These four gates are - input gate, output gate, forget gate and cell state. All the three gates namely- forget, output and input use sigmoid as activation function by default so all the values lie in the range from zero to one.

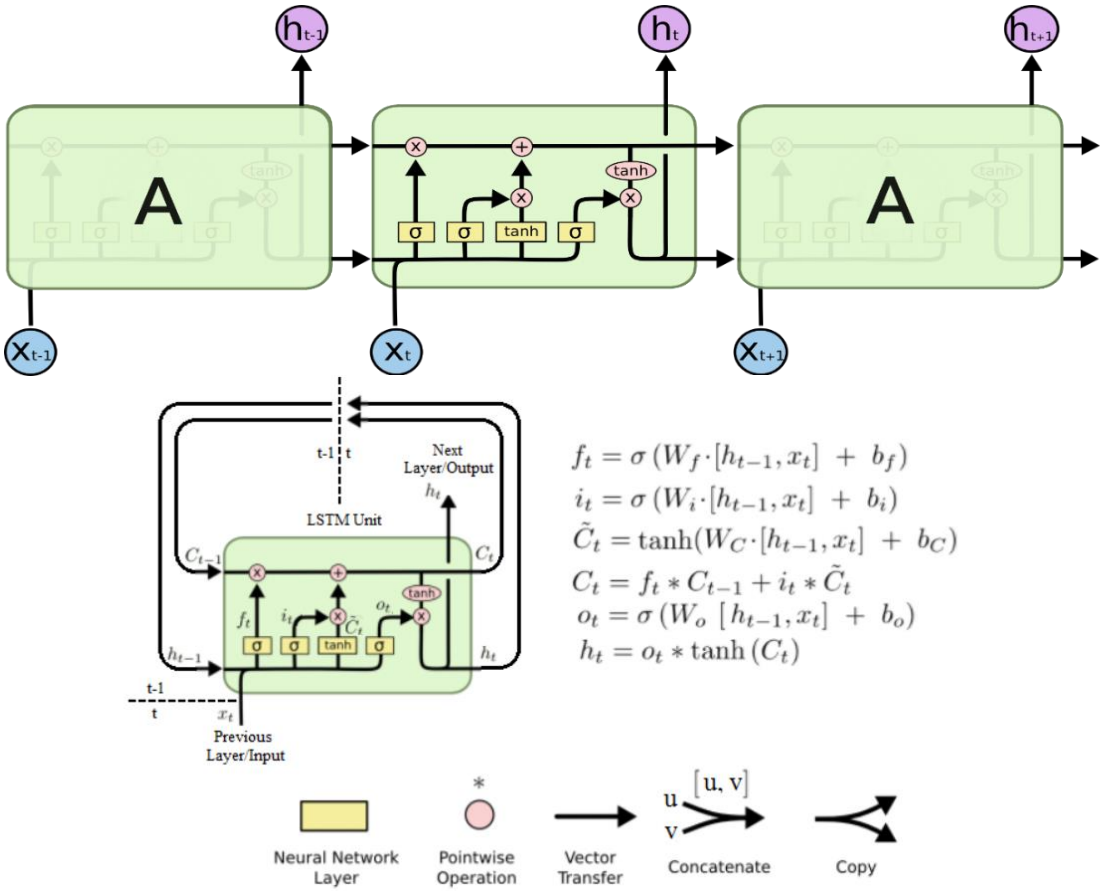


Figure 3. LSTM Architecture [9]

3.1.4 Interaction Layers of LSTM

The specially interacting gates and cell state of an LSTM model gives it its unique properties. The cell state behaves like the memory of the whole network, transporting information through the sequence chain. Theoretically, it can carry relevant data throughout the sequence processing procedure which allows information from earlier time steps to make their way to current and later steps. This greatly reduces the problems caused by short-term memory. Gates of the network determine which information is permitted on the cell state and which needs to be discarded i.e., they learn only relevant information and ignore useless data while model training.

3.1.4.1 Sigmoid

The gates use sigmoid activation function which is quite similar in action to simple tan(h) activation function. The only difference is that it maps values between negative one and positive one. This makes remembering or forgetting information easier as any value being multiplied by zero becomes zero, causing it to be forgotten or ignored. On the other hand, if it is multiplied by one, it remains as it is which signifies that the model has learned or kept it. In this way the sigmoid aids network in learning important information and at the same time forgetting what is not required.

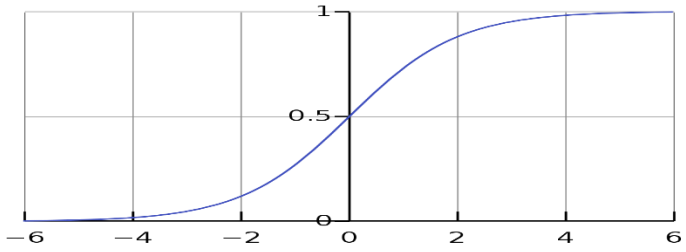
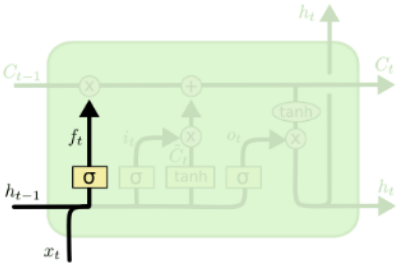


Figure 4. Sigmoid Curve [26]

3.1.4.2 Forget Gate

Also referred to as the "remember vector", this gate takes the decision of what should be learned and what must be thrown away. To do so, this gate sends the information obtained from the previous layer or hidden state as well as the information from the current input to the sigmoid function. Sigmoid returned values lie between zero and 1. If the value is very near to 0, means it can be forgotten, but if it is closer to one, then it is retained and sent to the next layer. This retained information is stored in the cell state.

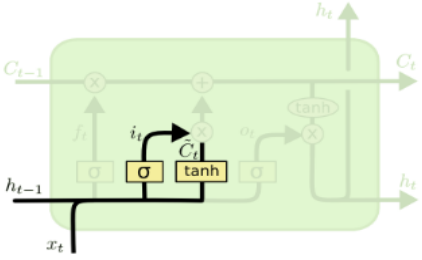


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Figure 5. Forget Gate [27]

3.1.4.3 Input Gate

Also called the "save vector", this is used to update the cell state. At first, information from preceding state and the input from current state are passed to a sigmoid function which returns a value in the range of zero and one. Thereafter, it is decided which values are to be kept. To regulate the network, hidden state and current input are also passed into the tan(h) function to map values in the range of -1 to 1. Then the output of the tan(h) is multiplied with sigmoid output, which ultimately helps in deciding which information is important enough to be remembered from this tan(h) output.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Figure 6. Input Gate [27]

3.1.4.4 Cell State

After collecting appropriate information, from other gates, cell state can be calculated. This process starts with taking the product of the cell state and the forget vector in a pointwise manner. If multiplied with values close to zero, the values may be dropped in cell state. Thereafter, input gate's output is used, and a pointwise addition is carried out which helps in updating the cell state to new values that the neural network thinks are important or relevant. This creates a new cell state.

3.1.4.5 Output Gate

Also called the focus vector, this gate of LSTM takes the important decision of what the next hidden state has to be. The hidden state, also called the working memory, consists of information about preceding inputs and is responsible for predictions. First of all, the previous hidden state and the current input is given to the sigmoid function and then the modified cell state is passed to tan(h) function. The output given by tan(h) function is then multiplied by the output of sigmoid to decide which information should be kept by the hidden state. The new cell state and the new hidden are then carried over to the next step.

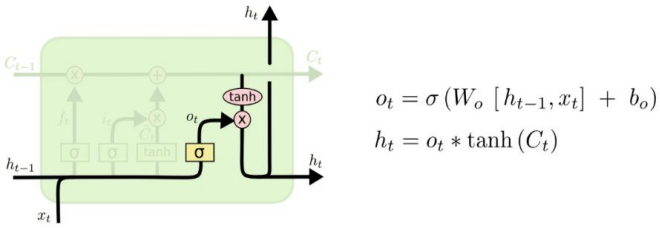


Figure 7. Output Gate [27]

3.2 Differential Privacy

Organizations and companies throughout the world need data to improve their services, so that they can provide products which are useful to the masses. But this inevitable jeopardizes individual privacy as such data might contain sensitive and personal information which can be misused by malicious entities to cause harm. To satisfy the needs of companies and to help users protect their privacy simultaneously, the technique of differential privacy was introduced. It allows the collection of such information from people but ensures that the privacy of no particular person is compromised at any cost. [11]

3.2.1 Data Anonymization

The anonymization process is carried out by various companies in an attempt to protect the private information of their users. It generally takes place on their servers and people have to trust them to remove public identifiers or records that might contain user-identifiable sensitive data. Even though the process of anonymization seems to be solving the problem, it has been proven to be ineffective against a variety of attacks.

The following examples shall illustrate that anonymization is not an effective process for data safety –

Example 1.

In 2006, Netflix started a competition called the “Netflix Price" in which the participants were required to create a robust algorithm that could easily predict how a person would rate a given movie. For this, Netflix gave a dataset containing over 100 million ratings submitted by over 480 thousand users for more than 17,000 movies. Netflix anonymized this data set by removing the names of users and by replacing some ratings with fake and random ratings. It might seem that this anonymization is enough, the reality is different. Two computer scientists from the University of Texas published a paper in 2008 that said that they had managed to reveal the identity of the people from the same dataset by combining it with data from IMDB. These types of attacks are called “Linkage attacks” in which different pieces of data that might seem to be anonymous are merged to reveal the individual identities that were supposed to be protected.

Example 2.

Case of Governor of Massachusetts in the mid-1990s, when the state’s group insurance commission decided to publish the hospital visits of state employees. They anonymized this data by removing names, addresses, and other fields that could identify people. However, computer scientist Latanya

Sweeney proved how easy it was to reverse this by combining the published health records with voter registration records and simply reduced the list. There was only one person in the medical data that lived in the same zip code, had the same gender and the same date of birth as the governor, thus exposing his medical records. Therefore, this technique is not enough to protect users' privacy.

3.2.2 How can Differential privacy help?

Differential privacy neutralizes the aforementioned types of attacks. Differential privacy is a process of gaining insights from large datasets while still maintaining privacy. It uses an algorithm to add a controlled amount of randomness or noise into the computation. **How many people like pineapple on their pizza?** We can set up a survey to get how the majority at a particular location votes so that a new pizza joint can decide the variety of pizzas they will make to increase their sales.

Do you like pineapple on pizza?

A) Yes
B) No

We should note here that the pizza joint wants to know the collective opinion of people about pineapple on pizza and not individual preferences.

Here, we collect all the answers on a server somewhere but instead of sending the real answers, we introduce some noise.

E.g., Yukti does not like pineapple on pizza and she clicked on option B.

Before her response is sent to the server, our differential privacy algorithm will flip a coin!

Figures 8 and 9 demonstrate a simple algorithm that explains the concept of differential privacy. If the coin is Heads, the algorithm sends Yukti's real answer to our server. If it is Tails, the algorithm flips a second coin and sends 'Yes' if it tails and 'No' if its heads. On the server, the data comes but we cannot trust individual records. The record for Yukti might say that she does not like pineapple on pizza but there is at least a one in four chance that she actually likes pineapple on pizza and the answer was simply the effect of the coin toss that the algorithm performed. As shown in Figures 8 and 9, we changed the responses of people at a predetermined frequency which helps to protect the privacy of the participants. Since we do not know which responses have been changed, there is no way to figure out any single individual's privacy.

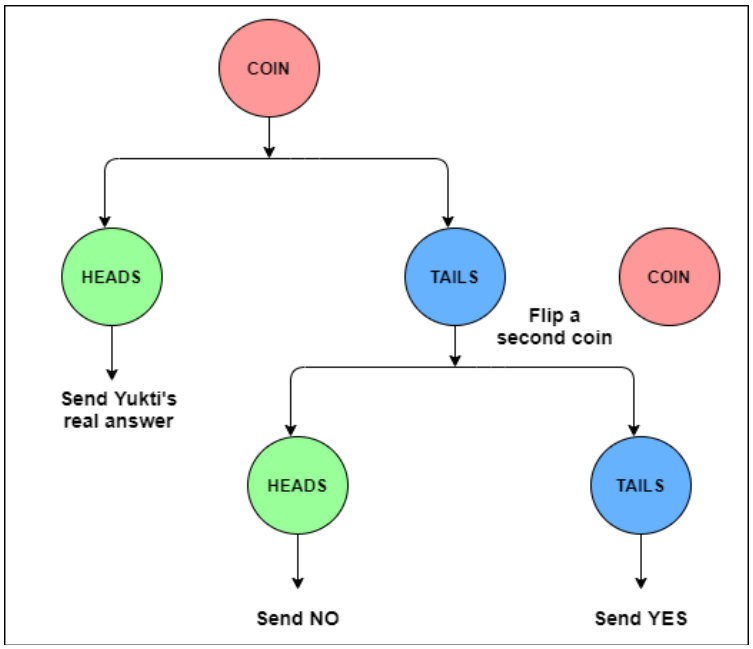


Figure 8. Illustration of A Simple Differential Privacy Algorithm at work

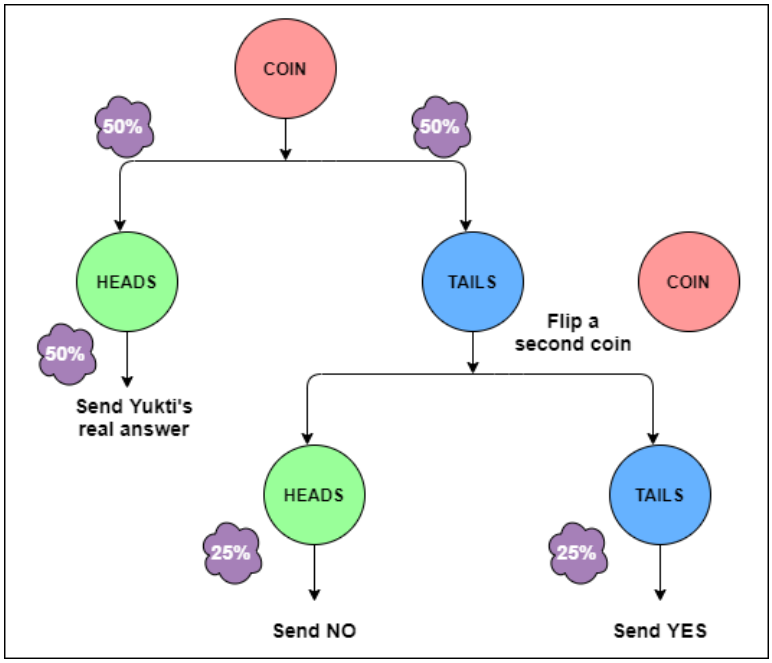


Figure 9. Illustration of the effectiveness of Differential Privacy

This is called "**Plausible Deniability**". Such an algorithm is particularly useful for collecting data about illegal habits like drug use. If we are aware of how the noise is distributed, we can compensate for it and end up with a fairly accurate view of how many people like pineapple on pizza. The above coin toss algorithm is just an example as it is too simple. Real-world algorithms use the Laplace distribution to spread data over a larger range and increase the level of anonymity. In the paper- The Algorithmic Foundations of Differential Privacy by Cynthia Dwork and Aaron Roth, it was noted that differential privacy promises that the outcome of a survey will stay the same whether you participate in it. Differential privacy is a mathematically provable measure of privacy. Therefore, you do not have any reason not to participate in the survey. People need not fear that their data will be exposed. Companies like Apple and Google are already using this technique. It should be noted that this technique should only be used for large data sets because of the injected noise. Using it on a tiny dataset will likely result in inaccurate data. The more noise we add to the original responses, the more privacy is protected but the less accurate the data becomes. Therefore, there is a tradeoff between accuracy and privacy, and we must reach a balance. [12]

Formally, Differential privacy can be defined as a way to share information in public about any dataset in the form of patterns present in it but ensuring that no person's information can be deciphered from this dataset. A differential private algorithm analyses the dataset and evaluates its statistics like mean, variance, mode and more, and uses them to alter the dataset in such a way that one cannot infer if an individual's data was included in it or not. The information obtained from such a dataset, such as the general pattern or popular preference of the people, is not affected by the changes that differential privacy makes. It further ensures that the behavior is not changed when a single individual leaves or joins the dataset. Any conclusion drawn from this dataset is as likely to be the case in the presence of an individual's data as is in its absence. This guarantee given by differential privacy holds for all individuals over all datasets. The details of any single individual do not affect this guarantee, thereby ensuring that no information about the participants is leaked.[13]

3.3 Sentiment Analysis Using VADER

Opinion mining is an important technique to understand the subjective context and tone of the text that needs to be analyzed. This is easily done by humans but in case of machines, such a procedure can be carried out

with the help of sentiment analysis tools like Vader. It uses sentiment lexicon labeled according to their semantic orientation.

This tool allows us to understand a sentiment of huge amounts of data within seconds, in the form of numerical numbers that indicate if the tone is negative or positive. Moreover, it also indicted the level or intensity of the text's positive or negative nature. These sentiment scores are calculated by taking the sum of the intensity of each word present in that piece of text. For instance, words like 'wow', like', and 'great' signify a positive tone. Further, Vader can also easily decode the basic context of word usage. This means that it will correctly understand the statement "does not like" as a negative one rather than declaring it positive merely due to the presence of the positive word 'like' in it. Besides that, it also works incredibly well with text containing slangs, punctuations, or emphasis like capitalization.[15]

Therefore, it is quite good at dealing with various sentiments expressed in the text such as, when used in the field of marketing, it can help to understand customers' feelings towards a service or to decipher how people respond to a company's campaigns and product launches or why consumers do not want a particular product.

3.3.1 Working of VADER

Vader uses a lexicon dictionary containing semantic scores. In this approach, each of the words present in lexicon is by default rated as neutral, positive, or negative along with the strength of their positive or negative sentiment. Higher negative rating is assigned to more negative words and vice versa. Positive words are handled in a similar way. Table 1 shows sentiment ratings of various words provided in Vader lexicon.

Word	Sentiment Rating
Apprehension	-2.1
Euphoric	3.2
Horrible	-2.5
Forbid	-3.1
Growth	1.6
Drop	-1.1

Table 1. VADER word-sentiment rating example

For any given piece of text, sentence or article, Vader first checks to find out if any words of the text are present in the lexicon.[25] For instance, the sentence, **"The furniture is dull, but the house looks good."**, has two words present in the lexicon - dull and good which have a sentiment rating of -1.7 and 1.9, respectively. Vader uses these ratings to calculate four types of sentiment metrics as shown in Table 2.

Sentiment Metric	Score
Positive	0.303
Neutral	0.551
Negative	0.146
Compound	0.458

Table 2. VADER sentiment scores for non-financial text

Table 2 demonstrates the various types of sentiment scores i.e., positive, neutral, and negative valence scores calculated by Vader which represent the percentage of text that falls in those segments. This means that the given sentence is rated as 14.6% negative but 30.3% positive and 55.1% neutral. The compound score is usually considered as the final metric which is calculated as the sum of all lexicon ratings after normalizing them to lie in the range of -1 and 1. This sentence has a compound score of 0.45 indicating an overall positive emotion or tone, which is indeed the case.

Applying VADER to a financial news text –
Example sentence – "**Dow Jones hits record high as materials, energy stocks jump.**"

Sentiment Metric	Score
Positive	0.189
Neutral	0.811
Negative	0.0
Compound	0.273

Table 3. VADER sentiment scores for financial text

The compound score of the above example sentence, shown in Table 3, is positive concluding that the sentence is showing an overall positive financial sentiment, which indeed should be the case, as the news headline indicates that Dow Jones have hit a record high in energy and material stocks. This news can have a very positive bearing on the stock prices of both the company and hence Vader has correctly expressed the sentiment of this sentence.

This process of understanding the sentiment of a word along with the degree to which the word expresses that positive or negative tone, needs to be manually done by people which is very expensive and time-consuming. The lexicon must also cover a good enough number of words relating to the topic of interest otherwise, sentiment analysis will not be accurate. But when an appropriate lexicon is available which suits the needs of the given text, Vader gives quite accurate results in a very short time even when the data that needs to be analyzed is enormous.

3.3.2 Advantages of VADER

1. It can tell the intensity of positive or negative nature of text and can work with multiple domains.
2. No training data is required to work with Vader.
3. It works particularly well on articles, news, or social media texts.
4. Conjunctions, slangs, capital words, punctuations and emoticons do not limit its ability to understand the context and tone of the text.

Chapter 4 – Related Work

Over the years, there have been many methods and tools used for stock market prediction due to its complex and dynamic nature. Such predictions help investors to opt the stocks that may give a higher return in the future. Many machine learning models have been worked upon to give stock predictions like logistic regression, support vector machines (SVM), and Convolutional Neural Networks (CNN) to determine next minute price movement of the stock market index.[3] Due to the increase in the amount of data and the expectation of more accurate results, deep learning models are being used nowadays. These outperform traditional approaches in both speed and accuracy. Many Artificial Neural Networks have been used to predict the stock price and trends. Even convolutional neural networks, which are most generally used for image processing problems, have also been used to predict the movement of stock prices by treating it as a classification problem. Recently, many researchers have explored the effectiveness of LSTMs in stock price prediction due to their advantages like storing long-term memory and solving the vanishing gradient problem. Appropriate LSTM models have been proposed by tuning their parameters like layers, dropout regularization, and batch size to get better accuracy.[2] Recurrent Neural networks like LSTMs have also been employed for stock volatility prediction to help traders in making bets or providing liquidity in the options market, because of their suitability for time series data.[24] Due to a high dependence of stock prices and financial markets on public sentiments expressed by articles and social media, many researchers like Rubi Gupta and Min Chen [23] investigated the effect of such sentiments on the stock market using many types of machine learning and text featurization algorithms. The paper - Differential Privacy-inspired LSTM for Stock Prediction Using Financial News explored how the combination of sentiment analysis and LSTM can prove to be very useful in stock price prediction. Authors Xinyi Li, Yinchuan Li, Hongyang Yang, Liuqing Yang, and Xiao-Yang Liu have also proposed how differential privacy can be used to enhance the effectiveness of such predictions.[1]

Creating datasets that can be used to perform predictions that incorporate financial sentiments to historical stock data to reduce prediction error, is one of the most difficult tasks due to the largely unstructured and noise-prone nature of such data. This leads to problems in analyzing the actual effectiveness of such models. Therefore, in this project, we have created a total of 15 datasets by combining global financial news and stock price data of 7 different companies and the SP500 market index. The data was extracted through many sources and cleaned through a series of stages. Sentiment analysis was then performed to extract important sentiments that might have a huge impact on the future prices of stock of these companies. Thereafter, we performed an extensive analysis of two types of recurrent neural networks i.e., Simple RNNs and LSTMs to compare their performance on all these datasets. Our proposed model also accommodated concepts like differential privacy to create two high-performing models - DP-RNN and DP-LSTM which were tested multiple times over all the datasets. A multiway comparison was done to analyze the performance of each of the models for different types of datasets. We studied how global news sentiments can be much more useful in stock price prediction rather than solely concentrating on news that affects a single company. LSTM model outperformed Simple RNN's in almost all datasets and sample runs while the use of differential privacy and news sentiments greatly enhanced the results of price prediction. In the end, we also employed the trained models to forecast future stock price values beyond the test data available.

Chapter 5 – Proposed Method

5.1 Concept of Solution Proposed

In this project, we have worked on solving a variety of prevalent problems posed by the problem of stock price prediction. To automate the process of stock prediction and ease the workload of investors and traders, we have used artificial neural networks like vanilla RNN and LSTMs to learn the historical stock price patterns and predict the future movement. Moreover, to combine the power of sentiment analysis of financial news and articles with that of these deep learning models, we have incorporated news sentiment scores as input to the proposed models. The biggest challenge to the thorough study and research of various factors that affect stock prediction is that appropriate datasets that use both news sentiment scores and historical stock price patterns as input for properly training neural networks are not readily available and the ones available often are unstructured, noisy, and too small. To address this issue, we have created a total of 15 datasets spanning from 2008 to 2020 after proper cleaning, noise removal, and sentiment analysis. Even after proper cleaning and noise removal, news headlines and articles are often found to be fake and might contain authors' bias or personal opinions affecting their reliability. Therefore, to further enhance the accuracy and robustness of our models, we have used the concept of differential privacy.

In this project, Differential Privacy has been employed not just to provide privacy of data i.e., sentiment scores of financial news but to increase the prediction performance of the model because along with helping in protecting individual sentiment privacy, it serves to mask individual biases often found in news articles giving more accurate representation of the whole dataset.
Even though differentially private algorithms were developed to help in secure data analysis to handle sensitive information, this concept can also be used to improve the efficiency of the model.[1] More robust data will help in improving the accuracy of prediction. News datasets containing fake or biased news can hamper model performance, thereby, making it important to counter this problem. Using differential privacy to add appropriate noise to the data can improve prediction results, thereby making this approach meaningful. The various properties of differential privacy can contribute to effective textual information extraction, resulting in an accurate tone analysis of news headlines. [14] It guarantees that that accuracy will not be affected by some false data that might be present in the dataset. DP-mechanism is used for sentiment compound scores extracted by sentiment analysis. The noise-added data is then fed to the model along with stock prices for each trading day.

5.2 Prediction Models and Methods used

This project shows a multiway comparison of the following proposed models and methods that serve to enhance their prediction efficiency -

1. The following two types of Recurrent neural networks have been used:
    a. Vanilla Recurrent Neural Networks
    b. Long Short-Term Memory Neural Networks

2. The above two deep learning models have been trained with three types of dataset setting:
    a. Without news sentiments
    b. With news sentiments
    c. With Differential Privacy on News

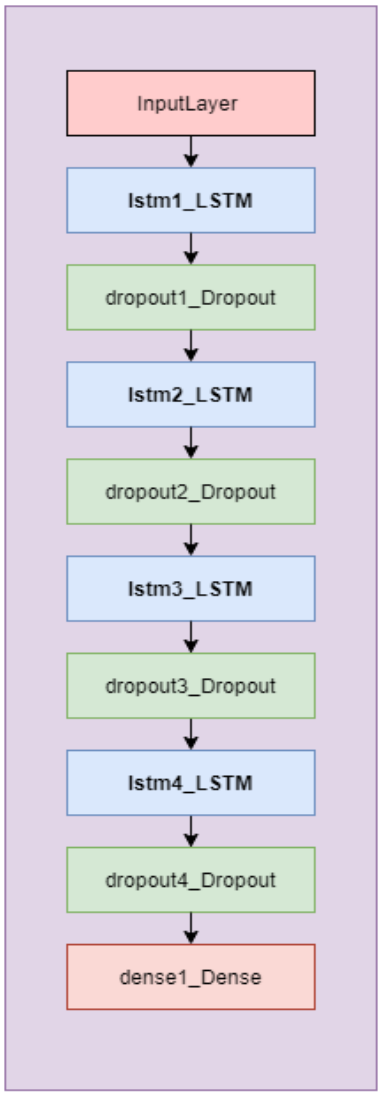


Figure 10. High-Level Architecture of LSTM Model

5.3 Architecture of Proposed Neural Networks

- The LSTM model used to train the data has 10 layers – an LSTM layer of 60 neurons, a dropout layer, an LSTM layer of 60 neurons, a dropout layer, an LSTM layer of 80 neurons, a dropout layer, an LSTM layer of 120 neurons, a dropout layer, and a dense layer for reshaping the output.
- Each of the 4 LSTM layers uses 'RELU' as an activation function. A dropout regularization with a rate of 0.2 is employed for each layer, to prevent overfitting of the model.
- In all, we have used 4 LSTM layers. In each of these four layers, the loss function that we have used Mean Squared Error (MSE). As mentioned in Section 2.2.2, MSE refers to the sum of square of the differences between the predicted values and the actual values.
- Batch size is critical for LSTM to learn the common pattern, so a batch size of 32 is used for the model.
- The model is compiled with ADAM as an optimizer because this is very straightforward to implement. Moreover, it is not only computationally efficient but also appropriate for problems with huge data sets and lots of parameters.
- We have used a rule-based sentiment analyzer i.e., VADER for sentiment analysis of financial news which is fed as one of the features of the LSTM model.
- RNN has the same architecture except the fact that vanilla RNN layers have been used instead of LSTM layers.

Chapter 6 – Data Processing

6.1 Dataset and Features

We have used two different datasets for training the models-

A. Dataset-A consists of Historical S&P stocks obtained from Yahoo Finance over 178 days (from 12th July 2017 to 6th January 2018) and US financial news from "Webhose archived" data (JSON format) from Dec 2017 to June 2018, consisting of headlines from 4 different sources- CNBC, WSJ, Fortune and Reuters (the top US and global business newspapers and magazines). After cleaning and combining the two components, a single dataset was formed consisting of Adjust close price and news sentiment score of a total of 121 trading days. The split size used was 0.85 i.e., 15% of data was used for testing the model while the rest for the training of models.[1]

B. Dataset-B also consists of two parts –
   a. Historical SP500 market index data dated 2nd January 2009 to 29th August 2017 as well as stock price data of 7 different companies i.e., Apple, Amazon, Google, eBay, ADM, ABT, and Exxon Mobil (XOM) downloaded from Yahoo finance dating 1st January 2008 to 1st January 2021.
   b. The US equity historical news dataset contains 2,21,513 financial news from 2nd October 2008 to 13th February 2020. It consists of 9 columns - Id, Ticker, Title, Category, Content, Release Date, Provider, URL, and Article Id, out of which 'Release Date' and 'Content' are utilized for prediction.

   After combining the components (a) and (b), 15 different datasets are formed:
      1. SP500 – 391 stock values from 2008-10-03 to 2017-08-29
      2. Apple, Amazon, Google, eBay, ADM, ABT, XOM - 2662 stock values from 2008-10-02 to 2020-02-13 with global us financial news sentiment scores
      3. The aforementioned company datasets were created with company-specific stock news sentiment scores-
         a. Google - 1376 stock values from 2009-05-06 to 2020-02-05
         b. Amazon - 1187 stock values from 2012-09-19 to 2020-01-07
         c. Apple – 1739 stock value from 2012-07-16 to 2020-01-28
         d. eBay – 1376 stock value from 2009-04-22 to 2020-02-05
         e. ADM – 1376 stock value from 2009-04-22 to 2020-02-05
         f. ABT – 1376 stock value from 2009-04-22 to 2020-02-05
         g. XOM – 1376 stock value from 2009-04-22 to 2020-02-05

For both Dataset type A and Dataset Type B, two features – Adjust close price and Compound News sentiment score are extracted and used for training the models. All the models were first trained without the news sentiment feature column and later including it to compare the difference in prediction results. The Adjusted Close price (close price adjusted for dividends and splits) of test data is predicted and testing data is used as a validation dataset while training.

6.2 Dataset Creation Preprocessing Stages

For the creation of the Dataset-B, three data-preprocessing steps were performed-

STEP 1 - Loading and Cleaning News Data

In this step, the news dataset is preprocessed. The missing news content rows and unnecessary feature columns are dropped. After which 2,21,505 rows of news records are left. This dataset is sorted by ascending order of news release dates. After all the cleaning and feature extraction is complete, the cleaned dataset is saved as "new_headlinest.csv".

STEP 2 - Financial News Sentiment Analysis

In the next step, VADER, referred to as valence aware dictionary and sentiment reasoner, is used to extract financial market sentiment scores from the news. Polarity scores for each article record are calculated in the form of positive sentiment, negative sentiment, neutral sentiment as well as compound score [2]

Mathematically, the compound score can be defined as the sum of negative, neutral, and positive valence scores which are later normalized to lie in the range of -1 to 1. The more positive this score is, the more would be the tone of the text. On the other hand, the more the score closer to -1, the higher the negativity of the text. As the compound score gives enough information about the headline sentiment, only the compound score for each news record is retained. The mean of compound score aggregated by article release date is evaluated i.e., A single date might have multiple financial news released, thus, the mean of compound scores for the multiple news released at a single date is calculated for all unique dates. This reduces the dataset size to 3641 news sentiment records. The scores and mean scores datasets are saved as "vader_scores.csv" and "vader_mean_compound_scores.csv" respectively.

STEP 3 - Combination of News sentiment and Stock Data

Yahoo finance and "Panda" data-reader is used to retrieve stock data of SP500 and specific companies consisting of columns like data, high, close, adjusted close, open, volume and low. Only Adjust close price columns are retained. The compound scores CSV files created in previous steps are then joined/merged with these adjusted prices based on the same date. The formed dataset is the final dataset having columns - "date", "adj close", and "compound score", to be used for model training.

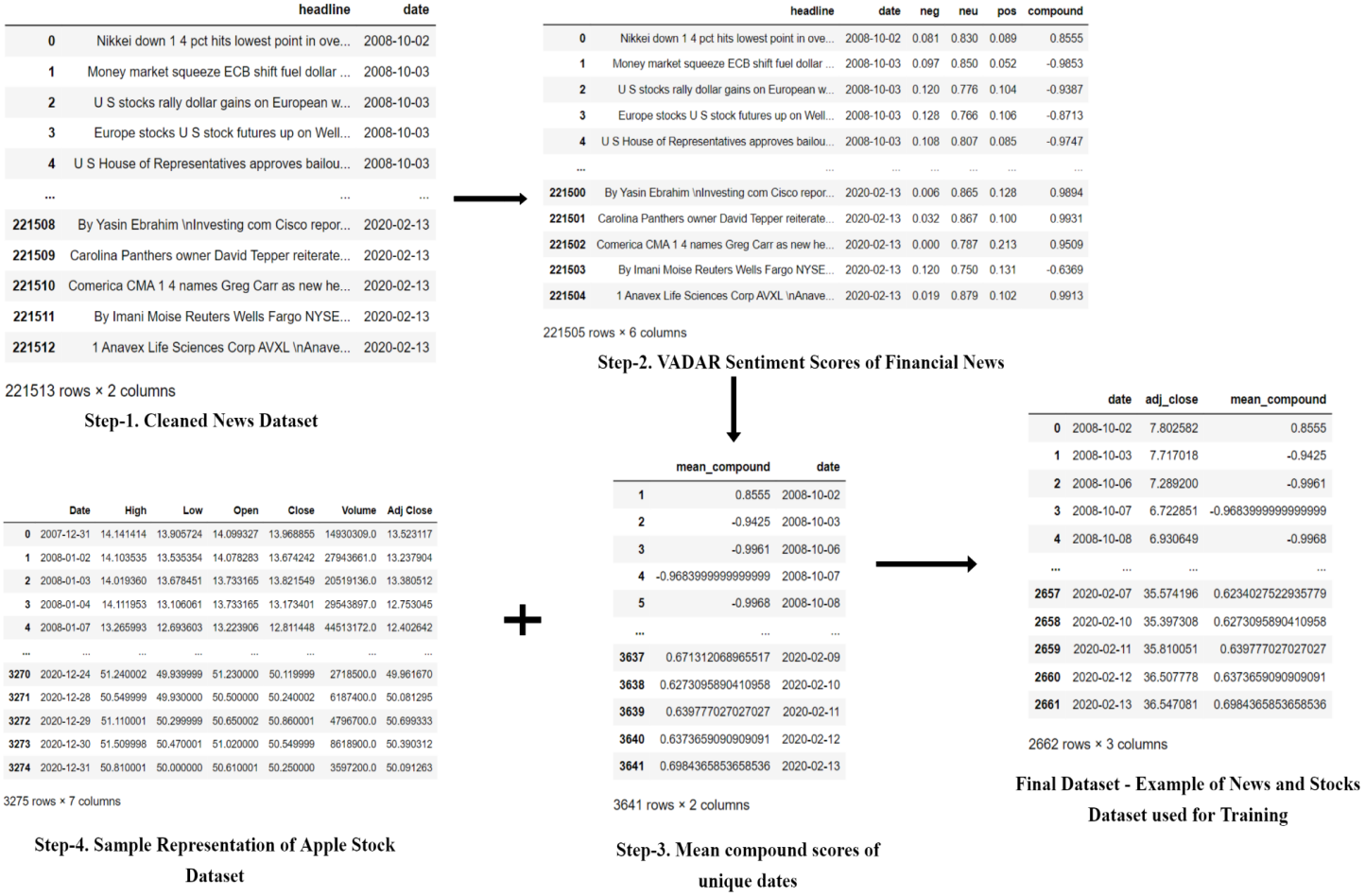


Figure 11. Stages of data preprocessing procedure

6.3 Financial Sentiment Analysis Using VADER

Changes in the news have a huge bearing on the stock market and therefore textual information from such news should show a correlation with stock prices. In this project, the news is considered as the financial news "headline" for any given day. Financial sentiment analysis was used to computationally identify and categorize the market sentiments that are expressed by the author in any single news headline or content. We have performed sentiment analysis using Vader sentiment analyzer to understand the sentiment of the financial news. First of all, the cleaned news dataset was analyzed to check if it is appropriate to conduct this stock prediction study using it. Figure 12. helps in visualizing the news dataset used in this project by showing word clouds formed by sentiment analysis on it. A word cloud is a perceptible representation of word frequency in a dataset wherein the words that occur with a higher frequency appear to be larger. This simple tool can help in understanding and identifying the focus of a dataset. The word cloud representing the whole new dataset demonstrates the financial nature of the article/news headlines used in this project. It can also be seen that positive word clouds show various words have a positive connotation in the news headlines. Examples of such words are gain, optimism, free, one, best, champion, great, impressive, and so on. The negative word cloud on the other hand indicates a negative sentiment in the news headlines detected by VADER. It includes words like war, crude, threat, hurt, fails, worst, scam, dismal, disaster, etc. The neutral cloud shows most frequent words which do not give either positive or negative connotation. The word clouds clearly demonstrate that this dataset is suitable for conducting this study.

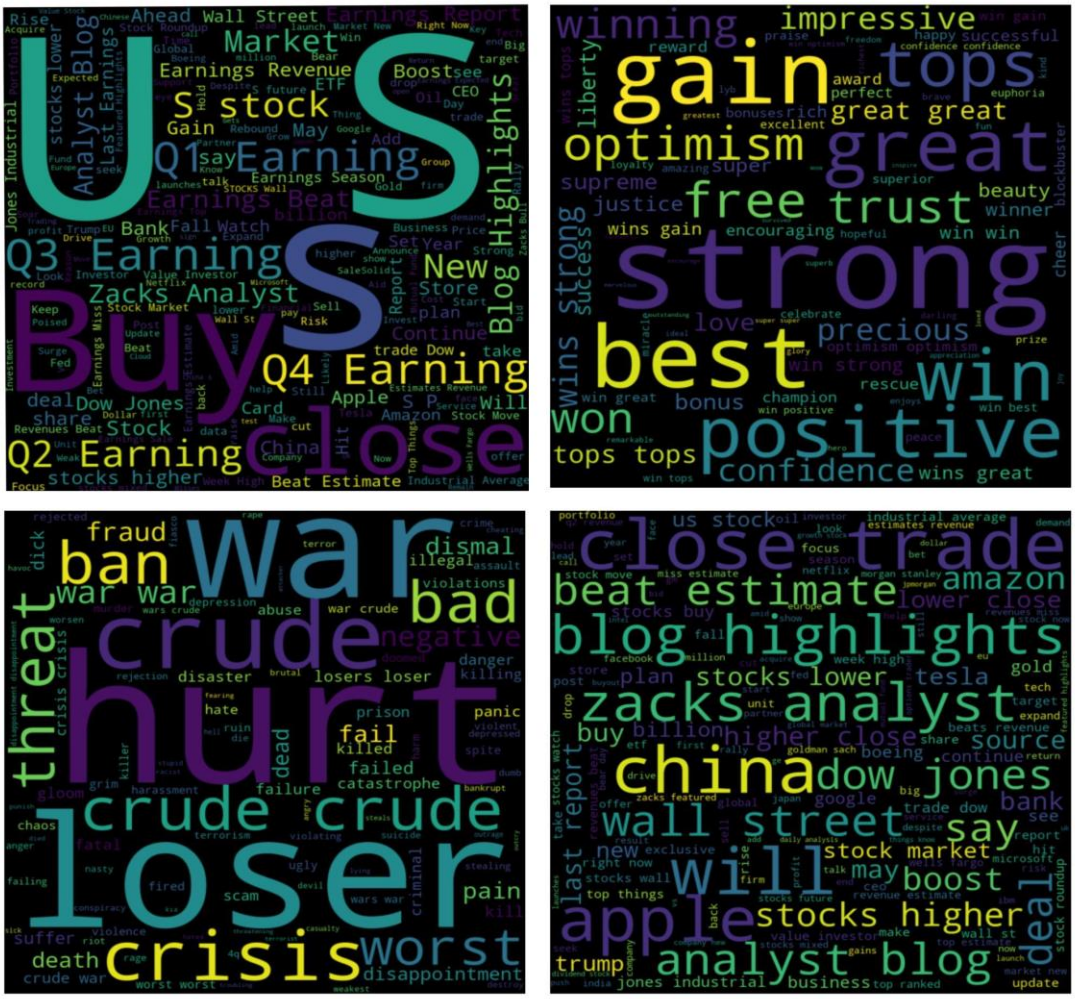


Figure 12. Visualization of Word Cloud formed by application of VADER on our financial news dataset. From top-left clockwise - (a) Complete dataset word cloud, (b) Positive word cloud, (c) Negative word cloud, (d) Neutral word cloud

Example of how VADER helped extract financial sentiment of news in this project -

1. Date: 2009-08-17
   News Headline: *"US STOCKS-Futures point to sharp drop after Japan data"*

   This news clearly indicates that the stock price might see a drop shortly. So, an investment expert or trader would manually rate this headline as a negative indicator for stock price.
   Results for this sentence using VADER –

	headline	date	compound
7	US STOCKS Futures point to sharp drop after Ja...	2009-08-17	-0.2732

Figure 13. VADER compound score for news headline 1

   As we can see in Figure 13, the compound score assigned by VADER is -0.2732, which is negative, but also not too negative as the news does not give any solid surety that stocks will fall, it just gives a future warning.

2. Date: 2013-01-24
   News Headline: *"Google Earnings Review Hopeful Signs in A Multi-Screen World"*
   This headline indicates a positive sentiment for Google stock prices.

	headline	date	compound
110	Google Earnings Review Hopeful Signs In A Mul...	2013-01-24	0.5106

Figure 14. VADER compound score for news headline 2

   As shown in Figure 14, the compound score assigned by VADER is 0.5106 which is positive. Thus, VADER correctly expressed the sentiment of this news for google stock prices.

This proves that VADER is quite efficient in analyzing financial news and hence is an ideal choice for this project. Figure 15 further shows the results of natural language processing (sentiment analysis) on the financial news headlines dataset.

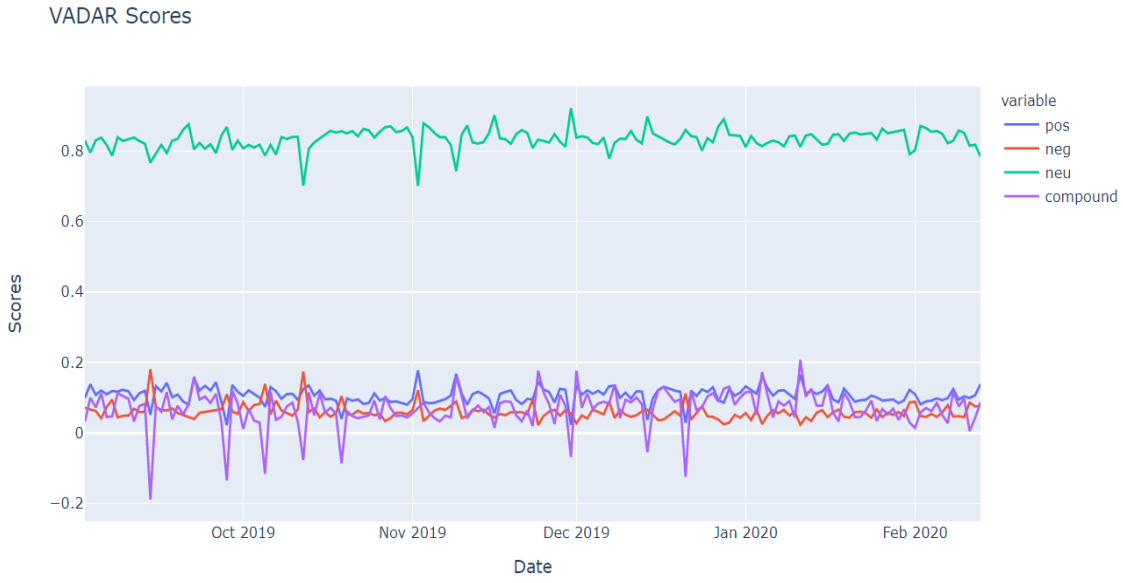


Figure 15. Polarity Scores assigned by sentiment analyzer VADER for the given dataset, showing the variation of news headline sentiments over different dates.

6.4 Window rolling method for prediction of test data

Training and Testing data is created using the datasets mentioned in Section 6.2, with a split size of 85% training and 15% testing. A rolling window having a size of 10 is used to separate data and prediction of the stock price is done using historical stock price data of the previous 10 days. This is called a point-by-point prediction. Training is performed on real stock prices. This window is shifted to add the next real stock price to the end of this training window to predict the following day's stock value using the trained model. Figure 12. demonstrates how the test data is divided into windows of size 10 and the first 9 price values of each window is used to predict the 10th stock precise value by feeding it to the trained model as input. Then all the predicted and real stock values are compared to find the accuracy of prediction.

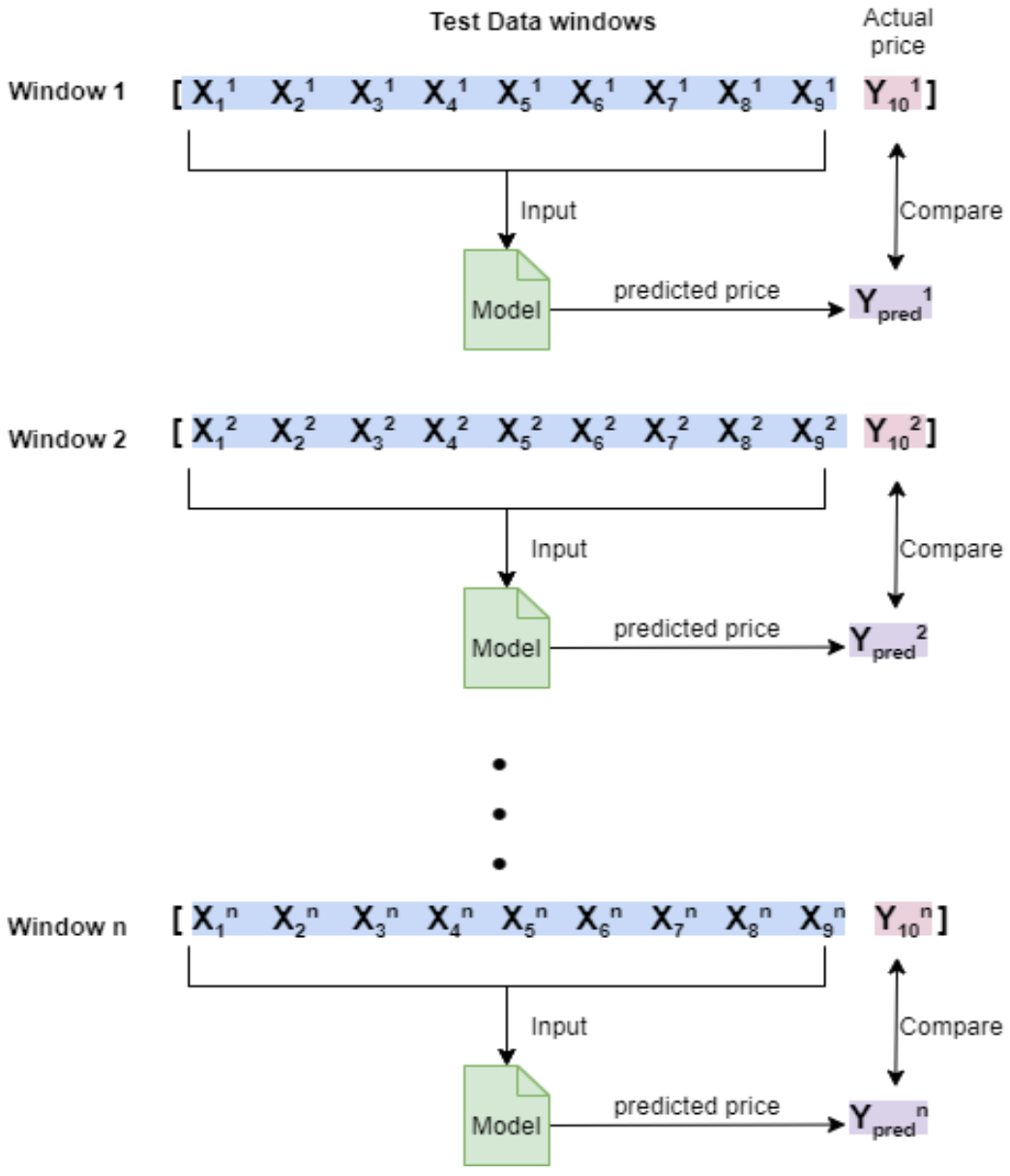


Figure 16. Test data windows for stock prediction

6.5 Data Normalization

When a deep neural network undergoes training using any dataset, it gradually understands and learns to map the given inputs to outputs with the help of training examples. The weights used by such a model are initialized to randomly small values. These weights are updated at each step based on the optimization algorithm using the error estimates obtained on the training dataset. For proper learning, the scale of these inputs and outputs is very important otherwise the learning process can be slowed down or become unstable. Unscaled target variables in regression problems often cause exploding gradient problems, which can eventually cause the training process to fail.[4] This standardization approach helps to map various features in the dataset to a common range and hence becomes necessary.

Therefore, it is important to normalize stock data so that price patterns can be identified, which are a requirement for the LSTM neural network during training. Since compound scores range from -1 to 1, we need not scale them. However, Adjust Close prices must be scaled. Here, "Min-Max" normalization is applied for this purpose, which works in the following way –

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

6.6 Data Denormalization

Normalization of data is for feeding into the LSTM, but for obtaining original predicted Adjust Close price values, we must de-normalize the normalized data using the following equation (reverse of formula for scaling data) –

$$\overline{X_{pred}} = \overline{X_{scaled}} * (X_{max} - X_{min}) + X_{min}$$

Here, $\overline{X_{pred}}$ refers to predicted, denormalized data and $\overline{X_{scaled}}$ refers to predicted, normalized data.

6.7 Adding Noise for Differential Privacy

We used the concept of differential privacy to increase the robustness of the model in predicting stock prices. For this, all the datasets were infused with noise to prevent the biases of authors in affecting the accuracy of prediction. Besides providing privacy to the data being used, differential privacy can help make the model more robust by ensuring that performance is unaffected by any false information that might be present in the news/article headlines. Gaussian noise was added to the news data with different variances in accordance with the variance of the news itself. The sentiment score columns of each dataset were doped with this noise in the following manner-

DP- Algorithm

```
mean_scores = compute_mean(C)
var = variance(mean_scores)
mu = 0
n = 0.1
σ = n * var
C' = C
for x in C' do
        C'[x'] += gaussian_distribution(mu,σ)
end for
```

return C’

C = compound sentiment score column of the dataset
C’ = compound score column after adding gaussian noise
n = noise weighting factor
mu = mean of the scores data
σ = standard deviation of the scores data (noise * sentiment score)
gaussian_distribution = random Gaussian distribution generator
compute_mean = function to calculate the mean of sentiment scores column

The noise was added to the training dataset after splitting the whole data into training and test data. The stock prices for market and individual companies were joined with the news sentiment scores obtained. These DP-applied datasets were then fed to the DP neural network for stock prediction.

Chapter 7 – Experimental Results

7.1 Model Performance Results and Inference

	Model	MAE (degrees)	MSE	RMSE	MPA
Without News	*RNN*	*13.7499*	*287.5568*	*16.9575*	*99.4945*
	LSTM	*11.2062*	*189.7756*	*13.7760*	*99.5875*
With News	*RNN*	*12.8143*	*249.2630*	*15.7880*	*99.5290*
	LSTM	*10.8045*	*179.2558*	*13.3886*	*99.6023*
DP on news	*DP-RNN*	*9.5512*	*128.3715*	*11.3301*	*99.6485*
	DP-LSTM	*10.0766*	*172.9389*	*13.1506*	*99.6295*

Table 4. Performance Results for Test Dataset-A

Table-4 shows the errors and mean performance accuracy of 2 neural network models – RNN and LSTM. Each of the models was performed using three methods i.e. "Without-News" (Compound sentiment score not used), "With-News" (Compound sentiment score was used), and DP-method (Noise added to the dataset for differential privacy) on dataset type-A (described in 6.1.A). It was observed that in all three cases, the LSTM model outperformed simple-RNN in terms of both less error and higher performance accuracy. Moreover, it could be observed that the DP model gave the best performance followed by with news model, which was followed by the "without-news" model. This shows that using compound score sentiment i.e., considering the weight of world news showed significant improvement in the prediction of stock prices.

	Model	MAE (degrees)	MSE	RMSE	MPA
Without News	*RNN*	*26.5429*	*1163.7146*	*34.1132*	*99.0602*
	LSTM	*17.9928*	*611.8498*	*24.7355*	*99.3630*
With News	*RNN*	*25.4552*	*1146.5602*	*33.8609*	*99.0962*
	LSTM	*12.5090*	*384.2966*	*19.6034*	*99.5541*
DP on news	*DP-RNN*	*25.2731*	*1129.5184*	*33.6083*	*99.1026*
	DP-LSTM	*10.1678*	*261.2841*	*16.1642*	*99.6411*

Table 5. Performance Results for Test Dataset-B, SP500 Market index prediction

Table-5 shows the same two models- RNN and LSTM performed in three modes – without news, with news, and DP-model, on a dataset containing SP500 market index, which shows the overall trend of stock prices of 500 companies. Similar results as before were observed as LSTM outperformed RNN and DP-method gave the minimum regression error.

	Model	MAE (degrees)	MSE	RMSE	MPA
Without News	*RNN*	*7.4106*	*312.899*	*9.6069*	*98.5655*
	LSTM	*4.6056*	*132.9515*	*6.3289*	*99.1114*
With News	*RNN*	*7.3510*	*309.1240*	*9.5425*	*98.5699*
	LSTM	*3.9924*	*103.0776*	*5.6163*	*99.2199*
DP on news	*DP-RNN*	*7.3113*	*302.449*	*9.4097*	*98.5732*
	DP-LSTM	*3.4284*	*84.6797*	*5.1299*	*99.3104*

Table 6. Performance Results for Test Dataset-B containing global news headlines sentiment, an average of the results for seven companies- Apple, Amazon, Google, eBay, ADM, ABT, and XOM.

To compile Table-6, the average of errors and MPA of 7 companies - Apple, Amazon, Google, eBay, ADM, ABT, XOM was taken. This table signifies the effect of global financial news on the stock prices of various companies and shows the important relationship between stock trends and financial news. Here also, DP-LSTM was the winner while with news LSTM gave significantly good results

.

	Model	MAE (degrees)	MSE	RMSE	MPA
Without News	*RNN*	*6.1839*	*194.596*	*7.8896*	*98.7077*
	LSTM	*6.3238*	*201.527*	*8.0281*	*98.6932*
With News	*RNN*	*6.1187*	*189.256*	*7.8121*	*98.7239*
	LSTM	*4.2318*	*109.7657*	*6.0394*	*99.1294*
DP on news	*DP-RNN*	*6.0256*	*188.7849*	*7.7962*	*98.9432*
	DP-LSTM	*4.0805*	*95.9178*	*5.5871*	*99.2233*

Table 7. Performance Results for Test Dataset-B containing company-specific news headlines sentiment, an average of the results for seven companies- Apple, Amazon, Google, eBay, ADM, ABT, and XOM.

Table-7 contains the results obtained when the aforementioned methods were applied on Dataset-B, wherein for each company, only company-specific news headlines were taken into consideration for building compound sentiment score columns. This was done to observe if the effect on stock prices is limited to the news concerning the company being observed or are the stock prices more affected by global news headlines. It was observed that the results of Table-6 were much better than those given by Table-7 proving that company-specific headlines for stock price prediction might not be always sufficient, as multiple factors affect them, including the news concerning competitors and general market trends.

Table-8 is compiled by taking the average of all the values obtained from Table-4 to 7, to visualize how RNN models with various methods performed on different datasets. When compared with the “without-news” method, it could be observed that DP-RNN gave **0.11%** better accuracy and a decrease of about **3.35%** in Mean squared error. While “with-news” outperformed “without news” RNN by a **3.29%** decrease in mean

squared error.

	MAE (degrees)	MSE	RMSE	MPA
Without News	*13.471*	*489.691*	*17.141*	*98.956*
With News	*12.934*	*473.550*	*16.750*	*98.979*
DP on news	*12.043*	*473.280*	*15.536*	*99.066*

Table 8. Average Performance Results of RNN model on all datasets

	MAE (degrees)	MSE	RMSE	MPA
Without News	*10.032*	*284.026*	*13.217*	*99.188*
With News	*7.884*	*194.098*	*11.162*	*99.376*
DP on news	*6.938*	*153.705*	*10.008*	*99.451*

Table 9. Average Performance Results of LSTM model for all datasets

Table-9 is compiled by taking the average of all the values obtained from Table-4 to 7 for the LSTM model to check its overall performance for the three methods used. When compared to the "without-news" method, the "with-news" method outperformed by a **0.19%** increase in accuracy, **31.66%** decrease in MSE, and **21.41%** decrease in MAE. The DP-LSTM method outperformed all the methods with an MPA of **99.45** and an MAE of about **6.94** degrees. Compared to the traditional "without-news method", DP-LSTM achieved an improvement of **0.26%** in MPA, **45.88%** decrease in MSE, and **30.84%** decrease in MAE.

On comparing the results of RNN and LSTM of Tables-5 and 6 respectively, it can be observed that LSTM performed better while predicting future stock prices as compared to RNN. The increase in accuracy of LSTM was about 0.388% and the decrease in MSE and MAE was about **67.52%** and **42.39%** respectively.

The above tables show that LSTM performs significantly better than RNN for stock prediction. When LSTM is coupled with news sentiment and Differential privacy features are added to it, it performs much better than traditional LSTM approaches. It may be noted that all the results are obtained by running multiple trials. Sections 7.2 and 7.3 shows results of stock prediction in the form of plots for some companies to easily visualize the performance results for some companies.

7.2 Stock Prediction Plots

Figure 17. Stock Test Data Prediction plot results for DP-LSTM model from Sep 2018 to Jan 2020. The real stock Adjust close price curve is represented by blue color while the red curve represents prices predicted by the model. From top to bottom, results for the following companies are given - (a) Amazon, (b) Google, (c) Apple

7.3 Stock Forecasting Plots

Figure 18. Stock Forecasting Curves showing the forecasted stock close prices for days beyond the testing data. The blue curve denotes real stock prices from17 Dec 2019 to 13 Feb 2020 while the red curve represents the forecasted future stock prices using the DP-LSTM model for the next 10 days from 13 Feb 2020 to 23 Feb 2020. From top to bottom, results for the following companies are given - (a) Google, (b) Apple, (c) ExxonMobil

Chapter-8 Conclusion

Over the years, investors and traders around the world have been dependent on the news related to relevant markets and instruments to make important trade decisions. This is called manual trading. This was prone to many risks arising from personal biases and emotional responses of the trader to the variety of news floating around. The emergence of algorithmic trading has remarkably reduced such risks. With the exponential increase in competition, traders came up with many new ways and methods for predicting the stock market and taking calculated and smart calls, to have an edge over other traders. One of the emerging trends is a well-defined combination of algorithmic trading models and sentiment analysis. In this project, we created a stock prediction model that incorporated sentiment scores generated by analyzing different news headlines and articles available on the internet to refine trading signals obtained from conventional technical indicators.[20] We created a variety of different types of datasets to suit the needs of stock prediction and proposed a deep neural network to increase prediction performance of adjusted close prices and for observing the future movement of stocks in the form of forecasting curves.

Deep neural networks like RNN and LSTM were used as primary technical indicators for stock price prediction of companies as well as the SP500 market index. The performance results were refined using sentiment scores obtained by the Vader sentiment analyzer. The experimental results were obtained using 16 datasets. We combined the strength of a deep neural network with the NLP model to give accurate stock price values and reduce the investment risk. For sentiment analysis of financial news, VADER Sentiment analyzer was used, and opinion was extracted from text to indicate whether the stock price might increase or decrease. These sentiments were extracted in the form of a compound sentiment score and used as input to the deep neural network. All the results were first obtained using a traditional approach without the use of news for prediction. Thereafter the same neural network was integrated with the compound sentiment scores. The results showed that this led to significant improvement in prediction results. The two deep learning models used were a Simple RNN and LSTM which were tested using three different methods. The methods used for testing these were “without news”, “with-news” and “Differential privacy-based” methods. Since news is not always objective, the differential privacy method allowed us to decrease the bias of some non-objective news and yielded much better performance. Therefore, this method was proven to be useful for enhancing the robustness of the model. It was observed that LSTM on average when performed on the same datasets as a simple RNN, performed **0.388%** better than the latter. Moreover, a decrease of **67.52%** and **42.39%** in MSE and MAE respectively was observed indicating the LSTM is better suited for the prediction of stock prices. We also observed that datasets having news headlines concerning the global financial market gave better results in terms of **11.71%** decrease in MSE and **0.087%** increase in MPA as compared to those that consisted of company-specific news headlines. This indicates that a company’s future stock trend is not solely dependent on the news concerning the company itself, but a variety of other factors like news relating to the general stock market as well as headlines relating to other giant companies can be influential in the drop or rise of stock prices. Amongst the three methods, it was proven that the combined use of differential privacy and news sentiment is much better, giving an improvement of **0.265%** in accuracy and **30.841%** in MSE. This indicates that besides adding a privacy factor to the use of data, differential privacy mechanisms can also help in increasing the accuracy of prediction by masking misleading biases that might be present in the news being used for aiding prediction. Hence overall, DP-LSTM emerged as the winner.

References

[1] Xinyi Li, Yinchuan Li, Hongyang Yang, Liuqing Yang, Xiao-Yang Liu: Differential Privacy-inspired LSTM for Stock Prediction Using Financial News, 20th December 2019.

[2] A Deep Learning Approach for Stock Market Prediction Yan Miao, Stanford University.

[3] Machine Learning in Intraday Stock Trading: Art Paspanthong, Nick Tantivasadakarn, Will Vithayapalert, Stanford University.

[4] https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling

[5] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[6] https://www.infoworld.com/article/3397142/what-is-deep-learning-algorithms-that-mimic-the-human-brain.html

[7] https://medium.com/deeplearningbrasilia/deep-learning-recurrent-neural-networks-f9482a24d010

[8] https://medium.com/analytics-vidhya/rnn-vs-gru-vs-lstm-863b0b7b1573

[9] https://medium.com/@saurabh.rathor092/simple-rnn-vs-gru-vs-lstm-difference-lies-in-more-flexible-control-5f33e07b1e57

[10] https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d

[11] https://www.savjee.be/videos/simply-explained/differential-privacy/

[12] https://www.youtube.com/watch?v=NRf6sUk1bv0

[13] https://towardsdatascience.com/understanding-differential-privacy-85ce191e198a

[14] McMahan, H. Brendan, and Galen Andrew. “A general approach to adding differential privacy to iterative training procedures.” arXiv preprint arXiv:1812.06210 (2018).

[15] https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664

[16] https://www.geeksforgeeks.org/python-sentiment-analysis-using-vader/

[17] https://www.imperva.com/learn/data-security/anonymization/

[18] https://en.wikipedia.org/wiki/Differential_privacy

[19] https://builtin.com/data-science/recurrent-neural-networks-and-lstm

[20] https://blog.quantinsti.com/vader-sentiment/

[21] https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/

[22] https://deepai.org/machine-learning-glossary-and-terms/sigmoid-function

[23] R. Gupta and M. Chen, "Sentiment Analysis for Stock Price Prediction," 2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), 2020, pp. 213-218, DOI: 10.1109/MIPR49039.2020.00051

[24] Stock Price Volatility Prediction with Long Short-Term Memory Neural Networks by Jason C. Sullivan Department of Computer Science Stanford University Stanford, CA

[25] https://t-redactyl.io/blog/2017/04/using-vader-to-handle-sentiment-analysis-with-social-media-text.html

[26] https://towardsdatascience.com/machine-learning-part-20-dropout-keras-layers-explained-8c9f6dc4c9ab

[27] https://stackoverflow.com/questions/44273249/in-keras-what-exactly-am-i-configuring-when-i-create-a-stateful-lstm-layer-wi