1. INTRODUCTION

Stock market plays a vital role in the economy of company and nation. The stock market refers to the platform where the trading and issuing of equities that is the publicly held stock either through formal exchanges or over the counter markets. A single blow of stock market can ruin ones completely whereas the correct idea about market trend can turn ones into a millionaire. To get this idea stock market prediction shall prove to be a miracle. Know the future market trends with the help of stock market predictions. The proposed system fulfills the above mentioned demands and helps understanding the market trends. The stock market is the primary source for any company to raise funds for business expansions. It is based on the concept of demand and supply. If the demand for a company's stock is higher, then the company share price increases and if the demand for company's stock is low then the company share price decrease.

Different methods been deployed for the stock market prediction, some of them are time's series forecasting, statistical analysis, fundamental analysis and technical analysis. Because of the non-linear nature of stock market it is difficult to predict. Machine learning techniques such as the artificial neural network has ability to map nonlinear nature can be used effectively for time series analysis such as stock market prediction. But to have considerably good prediction ability it is important to train network properly with sufficiently large data so that on exposing it to real world considerable accuracy can be achieved. In the task of training it is important to consider proper set of input variable because input set represents factors that will be used & factors that are going to affect prediction and nonlinear mapping. Due to involvement of many number of industries and companies, it contain very large sets of data from which it is difficult to extract information and analyze their trend of work manually. Stock market analysis and prediction will reveal the market patterns and predict the time to purchase stock. The successful prediction of a stock's future price could yield significant profit. This is done using large historic market data to represent varying conditions and confirming that the time series patterns have statistically significant predictive power for high probability of profitable trades and high profitable returns for the competitive business investment.

The Nepal Stock Exchange Limited (NEPSE) is the only stock exchange of Nepal. It is established under the company act, operating under securities exchange act, 1983.It is located in Singha Durbar Plaza, Kathmandu Nepal. The basic objective of NEPSE is to import free marketability and liquidity to the government and corporate securities by facilitating transactions in its trading floor through member, market intermediaries, such as broker, market makers etc. So stock market can be termed as the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. The successful prediction of a stock's future price could yield significant profit. The efficient-market hypothesis suggests that stock prices reflect all currently available information and any price changes that are not based on newly revealed information thus are inherently unpredictable. Others disagree and those with this viewpoint possess myriad methods and technologies which purportedly allow them to gain future price information.

1.1 Problem Statement

There is unavailability of Prediction Tracking. None of the applications provide regular predictions for the Stock Prices. If we plan to invest on stock “xyz” today, then there is no application to predict its behavior at that instance. Also, there is no guarantee on the accuracy of the predictions. There is no application available today that can provide us a guarantee on their prediction or at least ensure the percentage of accuracy that their system can provide.

2. OBJECTIVE

- The main objective of this project is to predict an approximate value of share price along with the accuracy and recommend users whether they should invest or not.

3. LITERATURE REVIEW

In the last few decades forecasting of stock returns has become an important field of research. In most of the cases the researchers had attempted to establish a linear relationship between the input macroeconomic variables and the stock returns. After the discovery of nonlinearity in the stock market index returns, many literatures have come up in nonlinear statistical modeling of the stock returns, most of them required that the nonlinear model be specified before the estimation is done. But since stock market return is noisy, uncertain, chaotic and nonlinear in nature, ANN has evolved out to be better technique in capturing the structural relationship between a stock's performance and its determinant factors more accurately than many other statistical techniques. In literature, different sets of input variables are used to predict stock returns. In fact, different input variables are used to predict the same set of stock return data. Some researchers used input data from a single time series where others considered the inclusion of heterogeneous market information and macro-economic variables. Some researchers even preprocessed these input data sets before feeding it to the ANN for forecasting.

3.1 Relevant Works

Wilson and Sharda [1] studied prediction firm bankruptcy using neural networks and classical multiple discriminant analysis, where neural networks performed significantly better than multiple discriminant analysis. Min and Lee were doing prediction of bankruptcy using machine learning. They evaluated methods based on Support Vector Machine, multiple discriminant analysis, logistic regression analysis, and three-layer fully connected back-propagation neural networks. Their results indicated that support vector machines outperformed other approaches. Lee was trying to predict credit rating of a company using support vector machines. They used various financial indicator and ratios such as interest coverage ratio, ordinary income to total assets, Net income to stakeholders' equity, current liabilities ratio, etc. and achieved accuracy of around 60%. Predicting credit rating of the companies were also studied using neural networks achieving accuracy between 75% and 80% for the United States and Taiwan markets. Tsai and Wang [2] did a research where they tried to predict stock prices by using ensemble learning, composed of decision trees and artificial neural networks. They created dataset from Taiwanese stock market data, taking into account fundamental indexes, technical indexes, and macroeconomic indexes. The performance of Decision Tree + Artificial Neural Network trained on Taiwan stock exchange data showed F-score performance of 77%. Single algorithms showed F-score performance up to 67%. Kim and Han [3] used a genetic algorithm to transform continuous input values into discrete ones. The genetic algorithm was used to reduce the complexity of the feature space. This paper proposes a novel evolutionary computing method called a genetic quantum algorithm. Genetic Quantum Algorithm is based on the concept and principles of quantum computing such as cubits and superposition of states. Instead of binary, numeric, or symbolic representation, by adopting bit chromosome as a representation Genetic Quantum Algorithm can represent a linear superposition of solutions due to its probabilistic representation. As genetic operators, quantum gates are employed for the search of the best solution.

There are many tools and software available out there that provide forecasting of stock market entities, share quantity and share value for a given financial organization. Most of them claim to predict the stock market with near to 100% accuracy but the opinions from the users vary.

4. METHODOLOGY

4.1 System Description

This system mainly consists of three modules:

- Firstly, data is collected from NEPSE.
- Secondly, analysis is carried out on the collected data by examining the current market direction, tracking the industry group and specific companies after which the data is represented and scored accordingly
- At last, LSTM-RNN is designed to predict the stock value. By this we can also recommend users whether to buy or sell shares.

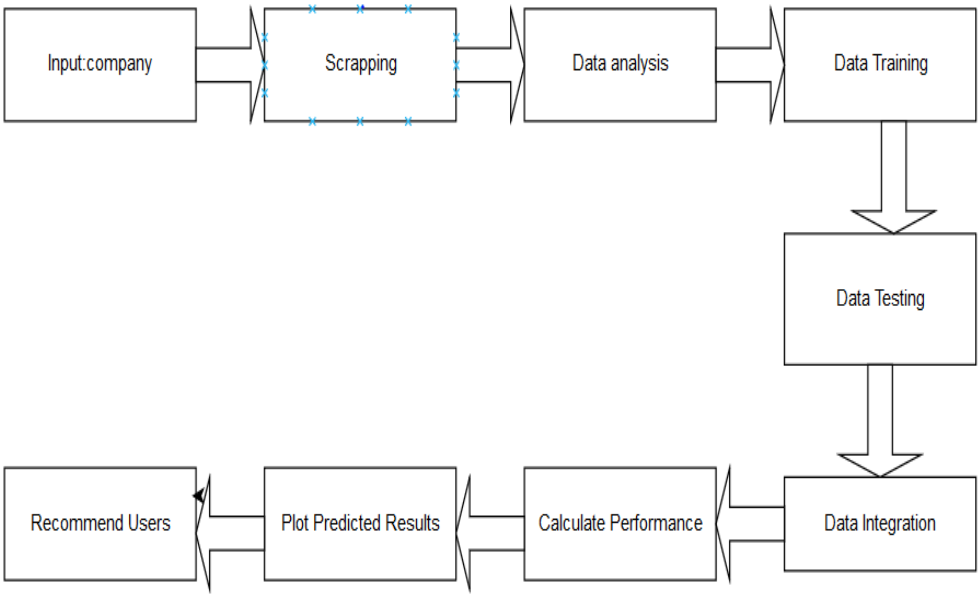


Figure 1:Block Diagram of System

4.2 Backend Development

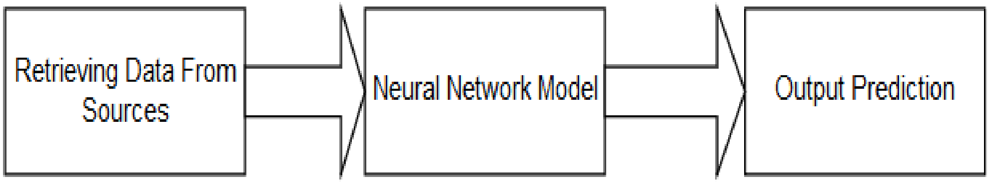


Figure 2:Procedure for Backend Development

4.2.1 Data Collection

This project attempts to predict the stock value with respect to the stock's previous value and trends. It requires historic data of stock market as the project also emphasizes on data mining techniques. So, it is necessary to have a trusted source having relevant and necessary data required for the prediction. We will be using Nepal Stock Exchange website (http://www.nepalstock.com.np) as the primary source of data. This website contains all the details such as: Closing value, highest value, lowest value, number of shares, increase or decrease in stock values for each financial companies. It also provides the overall performance of Nepal Stock Exchange and performance of companies of different categories. The site is updated on daily basis and it is also a repository for years of stock market data for Nepal. There is no API provided by the website for providing data. Web scraping has been done in order to collect the data required for prediction. Other sources of data are banking and financial statistics published by Nepal Rastra Bank, annual report of different sample banks, supervision report of Nepal Rastra Bank and annual reports of concerned banks. In addition to these, different published articles, report, book, journal, and graduate research project will also analyzed.

4.2 Design of Neural Network

4.2.1 Recurrent Neural Network

A recurrent neural network (RNN) is a class of artificial neural network where connections between units form a directed graph along a sequence. This allows it to exhibit dynamic temporal behavior for a time sequence. Unlike feed forward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. Recurrent Neural Network comes into the picture when any model needs context to be able to provide the output based on the input. Sometimes the context is the single most important thing for the model to predict the most appropriate output.

4.2.2 Problem in Typical RNN

One of the appeals of RNNs is the idea that they might be able to connect previous information to the present task, such as using previous video frames might inform the understanding of the present frame. If RNNs could do this, they'd be extremely useful. But can they? It depends.

Sometimes, we only need to look at recent information to perform the present task. For example, consider a language model trying to predict the next word based on the previous ones. If we are trying to predict the last word in "the clouds are in the sky," we don't need any further context – it's pretty obvious the next word is going to be sky. In such cases, where the gap between the relevant information and the place that it's needed is small, RNNs can learn to use the past information.

But there are also cases where we need more context. Consider trying to predict the last word in the text "I grew up in France… I speak fluent French." Recent information suggests that the next word is probably the name of a language, but if we want to narrow down which language, we need the context of France, from further back. It's entirely possible for the gap between the relevant information and the point where it is needed to become very large.

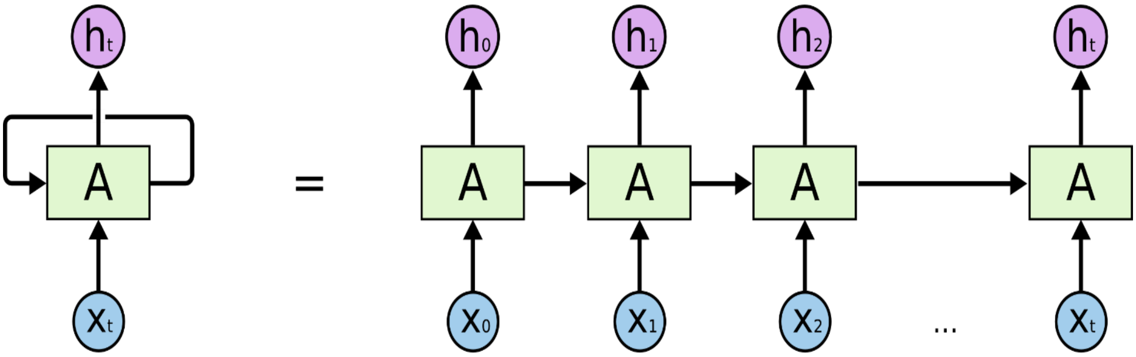


Figure 3:Typical RNN

4.2.3 Long Short Term Memory (LSTM) Networks

Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997), and were refined and popularized by many people in following work. They work tremendously well on a large variety of problems, and are now widely used. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn. All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer. The repeating module in a standard RNN contains a single layer.

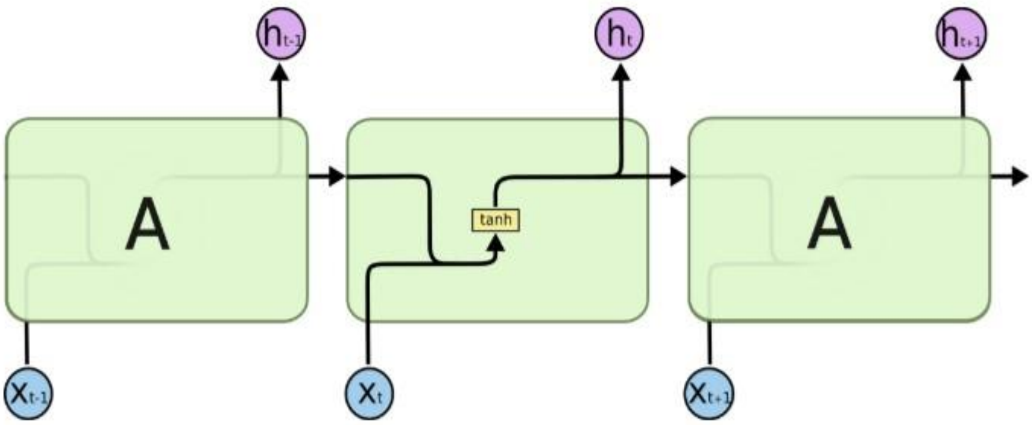


Figure 4:simple architecture of LSTM networks

The repeating module in an LSTM contains four interacting layers. LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.

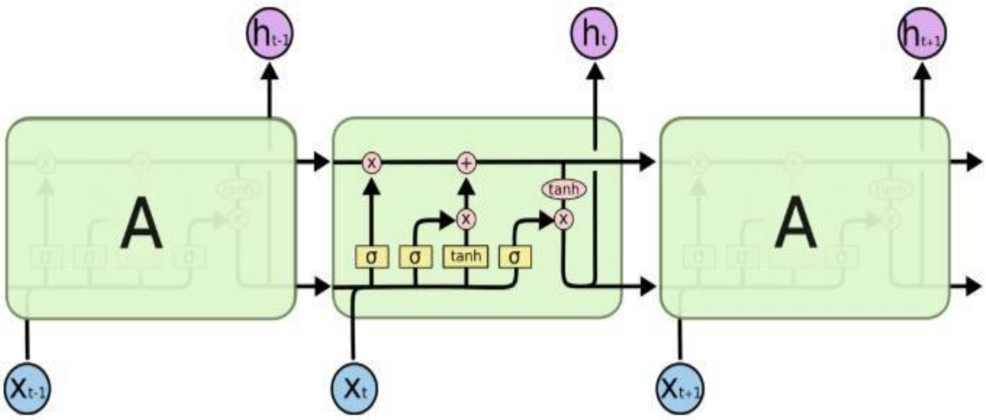


Figure 5:Detail Architecture of LSTM networks

4.2.4 Use of LSTM

Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997), and were refined and popularized by many people in following work. They work tremendously well on a large variety of problems, and are now widely used.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn.

All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

Long Short Term Memory(LSTM) network was built for better accuracy. Preprocessed data was subjected to this model and trained in this model. After training of data, testing and validation were performed and maximum training and testing accuracy were obtained from this model. Extremely high prediction accuracy for real time data were obtained from this model and thus it was concluded that LSTM networks have higher performance than RNN networks for our sort of long time series data.

4.2.5 Components in LSTM Network

1. Memory gate

To construct an architecture that allows for constant error flow through special, self-connected units without the disadvantages of the naive approach, we extend the constant error carrousel CEC embodied by the self-connected, linear unit j by introducing additional features. A multiplicative input gate unit is introduced to protect the memory contents stored in j from perturbation by irrelevant inputs. Likewise, a multiplicative output gate unit is introduced which protects other units from perturbation by currently irrelevant memory contents stored in j. The resulting, more complex unit is called a memory cell.

2.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.

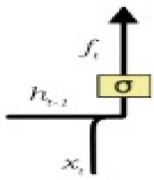


Figure 6:Forget Gate

3. Input Gate

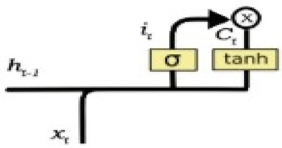


Figure 7:Input Gate

The input gate is responsible for the addition of information to the cell state. This addition of information is basically three-step process as seen from the diagram above.

- 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 h_t-1 and x_t.
- 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.

Once this three-step process is done with, we ensure that only that information is added to the cell state that is important and is not redundant.

4. Output gate

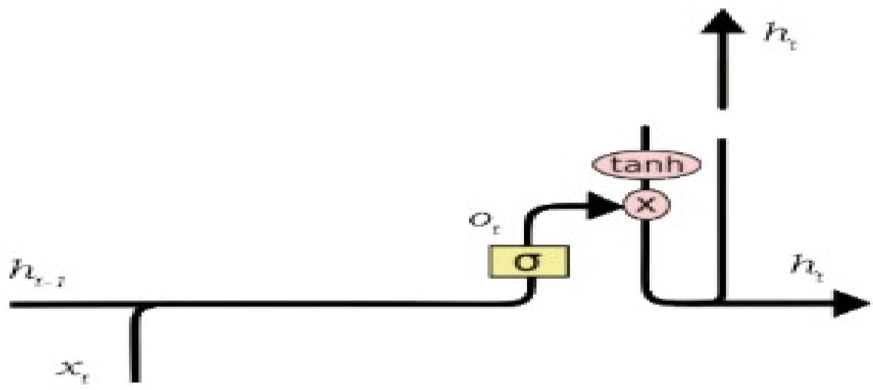


Figure 8: Output Gate

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.

The filter in the above example will make sure that it diminishes all other values but ‘Bob’. Thus the filter needs to be built on the input and hidden state values and be applied on the cell state vector.

4.3 Sigmoid Activation Function

The Sigmoid Function curve looks like an S-shape.

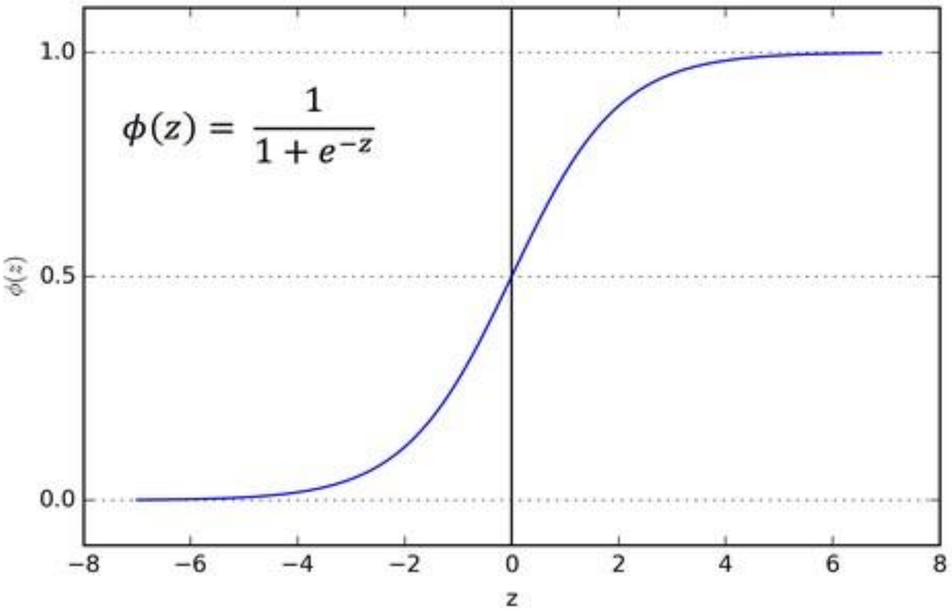


Figure 9:Sigmoid Activation Function

The main reason why we use sigmoid function is because it exists between (0 to 1). Therefore, it is especially used for models where we have to predict the probability as an output. Since probability of anything exists only between the range of 0 and 1, sigmoid is the right choice. The function is differentiable. That means, we can find the slope of the sigmoid curve at any two points. The function is monotonic but function’s derivative is not. The logistic sigmoid function can cause a neural network to get stuck at the training time. The softmax function is a more generalized logistic activation function which is used for multiclass classification.

It is required in Neural Networks that the output changes very slowly with input. Deep neural networks are what is in use presently and observing how a small change in the bias value or the weights associated with the artificial neurons affects the overall value of the output of the the neuron is very important. A perceptron may have the outputs flipped suddenly with a small change in the input value, thus leaving the machine learning engineer baffled. To observe the tiny changes in the output to come at the correct value of input, we need a function to be applied on the dot product of weights and bias value so that overall output is smooth. But now, the function could have been any function f() that is smooth in nature like quadratic function, cubic function. The reason sigmoid function is chosen is that, exponential functions generally are similar to handle mathematically and since learning algorithms involve lots of differentiation, thus choosing a function that is computationally cheaper to handle is quite good.

4.4 Global Variable Initializer

Global Variable Initializer is a shortcut to initialize all global variables. It is not required, and you can use other ways to initialize your variables or in case of easy scripts sometimes you do not need to initialize them at all. All variables can be initialized at once using the tf.global_variables_intializers. This op must be run after the model being fully constructed.

4.5 Technical Indicators

Technical indicators are fundamental part of technical analysis and are typically plotted as a chart pattern to try to predict the market trend. Indicators generally overlay on price chart data to indicate where the price is going, or whether the price is in an "overbought" condition or an "oversold" condition. A technical indicator is a mathematical calculation based on historic price, volume, or open interest information that aims to forecast financial market direction. Here are some types of technical indicators used in our project.

4.5.1 Simple Moving Average - SMA

A simple moving average (SMA) is an arithmetic moving average calculated by adding recent closing prices and then dividing that by the number of time periods in the calculation average. A simple or arithmetic, moving average that is calculated by adding the closing price of the security for a number of time periods and then dividing this total by that same number of periods. Short-term averages respond quickly to changes in the price of the underlying, while long-term averages are slow to react.

4.5.2 Exponential Moving Average - EMA

An exponential moving average - EMA is a type of moving average that places a greater weight and significance on the most recent data points. The exponential moving average -EMA is also referred to as the exponentially weighted moving average. Exponentially weighted moving averages react more significantly to recent price changes than a simple moving average, which applies an equal weight to all observations in the period.

4.5.3 Moving Average Convergence Divergence -MACD

The Moving Average Convergence-Divergence indicator, commonly known as MACD, is a technical indicator consisting of 2 lines the MACD line and the signal line as well as a bar chart. It is used to generate buy-and-sell signals, and to determine whether an investment or index may be overbought i.e. potentially expensive or oversold i.e. potentially cheap. MACD can be approximated by subtracting the value of a longer exponential moving average EMA from a shorter one. The shorter EMA is constantly converging toward, and diverging away from, the longer EMA. This causes MACD to oscillate around the zero level. A signal line is created with an EMA of the MACD line.

4.5.3.1 MACD Moving Average Crossovers

The primary method of interpreting the MACD is with moving average crossovers. When the shorter-term 12-period exponential moving average EMA crosses over the longer-term 26-period EMA a potential buy signal is generated; this is seen on the chart below:

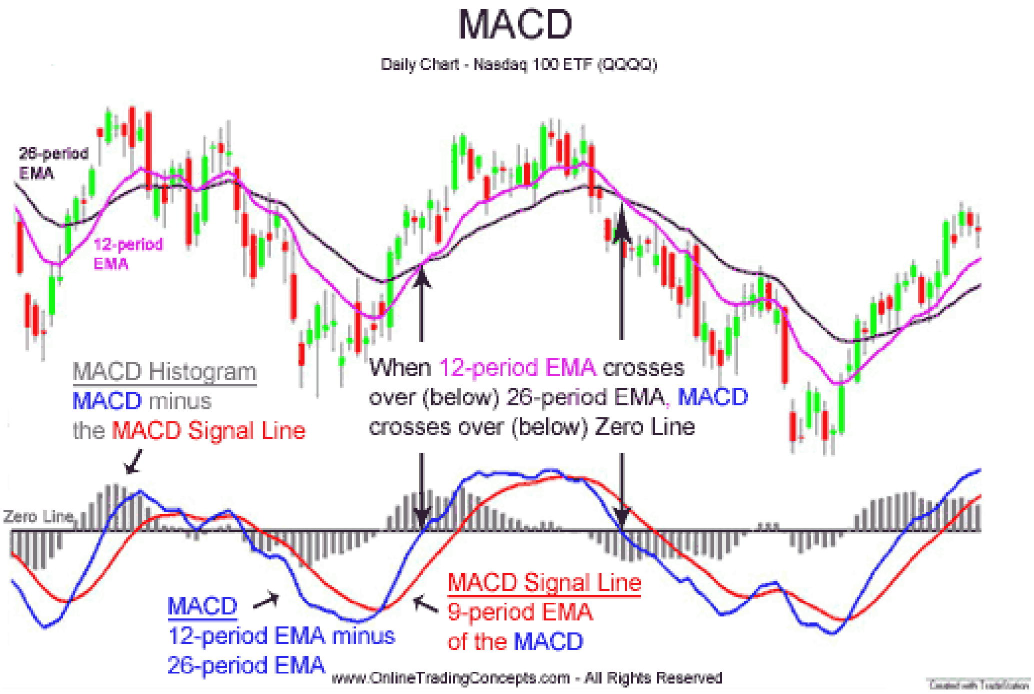


Figure 10:Most Common MACD Potential Buy and Sell Signals

Remember that the MACD line (the blue line)
is created from the 12-period and 26-period
EMA. Consequently:

• When the shorter-term 12-period EMA crosses above the longer-term 26-period EMA, the MACD line crosses above the Zero line.

• When the 12-period EMA crosses below the 26-period EMA, the MACD line crosses below the Zero line.

4.5.3.2 Moving Average Crossover Potential Buy Signal

A possible buy signal is generated when the MACD (blue line) crosses above the zero line.

4.5.3.3 Average Crossover Potential Sell Signal

When the MACD crosses below the zero line, then a possible sell signal is generated. The prior potential buy and sell signals might get a person into a trade later in the move of a stock or future. Another potential buy and sell signal is shown in the graph below.

4.5.3.4 MACD Potential Buy Signal

A potential buy signal is generated when the MACD (blue line) crosses above the MACD Signal Line (red line).

4.5.3.5 MACD Potential Sell Signal

Similarly, when the MACD crosses below the MACD Signal Line a possible sell signal is generated. The MACD moving average crossover is one of many ways to interpret the MACD technical indicator. Using the MACD histogram and MACD divergence warnings are two other methods of using the MACD.

4.5.4 Relative Strength Index - RSI

The Relative Strength Index - RSI is a momentum indicator that measures the magnitude of recent price changes to analyze overbought or oversold conditions. It is primarily used to attempt to identify overbought or oversold conditions in the trading of an asset.

The relative strength index (RSI) is calculated using the following formula:

RSI = 100 - 100 / (1 + RS)

Where RS = Average gain of up periods during the specified time frame / Average loss of down periods during the specified time frame.The RSI provides a relative evaluation of the strength of a security's recent price performance, thus making it a momentum indicator. RSI values range from 0 to 100. The default time frame for comparing up periods to down periods is 14, as in 14 trading days. Traditional interpretation and usage of the RSI is that RSI values of 70 or above indicate that a security is becoming overbought or overvalued, and therefore may be primed for a trend reversal or corrective pullback in price. On the other side of RSI values, an RSI reading of 30 or below is commonly interpreted as indicating an oversold or undervalued condition that may signal a trend change or corrective price reversal to the upside.

4.5.5 Signal Line

Signal lines are used in technical indicators, especially oscillators, to generate buy and sell signals or suggest a change in a trend. Often times, signal lines are moving averages of a technical indicator, such as the moving average convergence-divergence (MACD) and stochastics oscillator. A signal line is also commonly known as a "trigger line."

4.6 OHLC

An OHLC chart is a type of bar chart that shows open, high, low, and closing prices for each period. OHLC charts are useful since they show the four major data points over a period, with the closing price being considered the most important by many traders. The chart type is useful because it can show increasing or decreasing momentum. When the open and close are far apart it shows strong momentum, and when the open and close are close together it shows indecision or weak momentum. The high and low show the full price range of the period, useful in assessing volatility. There several patterns traders watch for on OHLC charts.

4.7 Component Used

4.7.1 Python

To create and train neural network with the dataset, python is used as backend tool. For the development of the web application, “django” is used as backend tool. As, “django” is a high-level Python framework and also its free and open source retains all the advantages that Python language offers like rapid development and clean, pragmatic design. As it is built by the experienced developers, it takes care of much of the hassle of Web development, so we can focus on writing our app without needing to reinvent the wheel.

4.7.2 TensorFlow

TensorFlow is an open source software library for numerical computation using data flow graphs. The graph nodes represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) that flow between them. This flexible architecture lets us deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device without rewriting code. TensorFlow also includes TensorBoard, a data visualization toolkit.

4.7.3 NumPy

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions

- tools for integrating C/C++ and Fortran code useful linear algebra, Fourier transform, and random number capabilities

4.7.4 SciPy

SciPy (pronounced “Sigh Pie”) is open-source software for mathematics, science, and engineering. The SciPy library depends on NumPy, which provides convenient and fast N-dimensional array manipulation. The SciPy library is built to work with NumPy arrays, and provides many user-friendly and efficient numerical routines such as routines for numerical integration and optimization. Together, they run on all popular operating systems, are quick to install, and are free of charge. NumPy and SciPy are easy to use, but powerful enough to be depended upon by some of the world’s leading scientists and engineers.

4.7.5 MatPlotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and Python shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code.

4.7.6 Pandas

Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language. It is already well on its way toward this goal.

After collection of data the future share price will be predicted using neural network. The value is then compared the next day with the actual value. This project involves building a web application that will be used for predicting the stock market. The application will be used for predicting the stock market based on the preprocessed datasets and recommend the user based on the user preferences.

With all the accumulated effort invested in this project, there are reasons to believe that at the end of the academic year this project will find itself in a much better shape and quite closer to actual acceptance. The proposed system will be capable of performing all the features it was previously pointed at the phase of requirement analysis. The project is aimed to develop a web application that is mainly focused on providing the better and more accurate prediction value for the future stock market.

5. SYSTEM DESIGN AND ARCHITECTURE

5.1 Software Development Life cycle

We are going to use agile software development life cycle for the development of this web application. Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like

- Planning
- Requirements
- Analysis
- Design
- Coding
- Unit Testing
- Acceptance testing

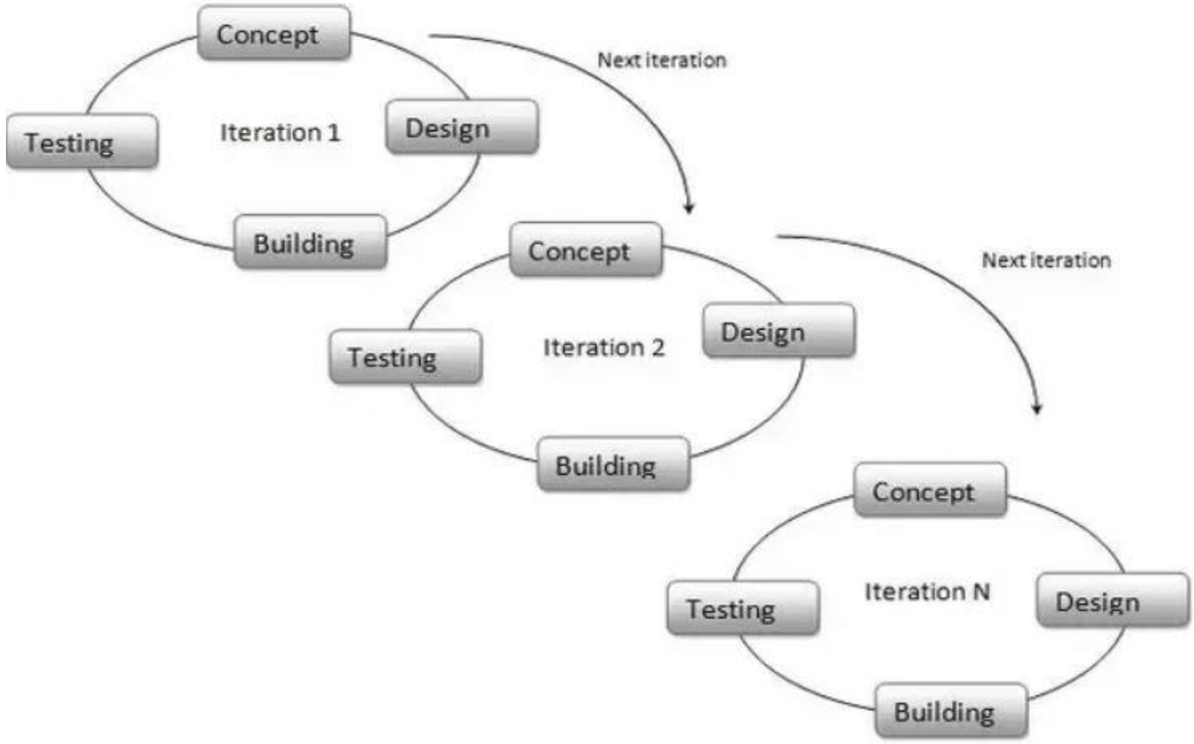


Figure 11:Agile SDLC

At the end of the iteration, a working product is displayed to the customer and important stakeholders. The proposed method for developing the system mainly consists of three major steps. Firstly data is collected and sorted for relevancy for various sources. Secondly analysis is carried out on the collected data by examining the current market direction, after which data is sorted and represented accordingly. At last, artificial neural network is designed and a suitable algorithm yielding best accuracy is chosen to predict the stock value.

5.2 Use Case diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

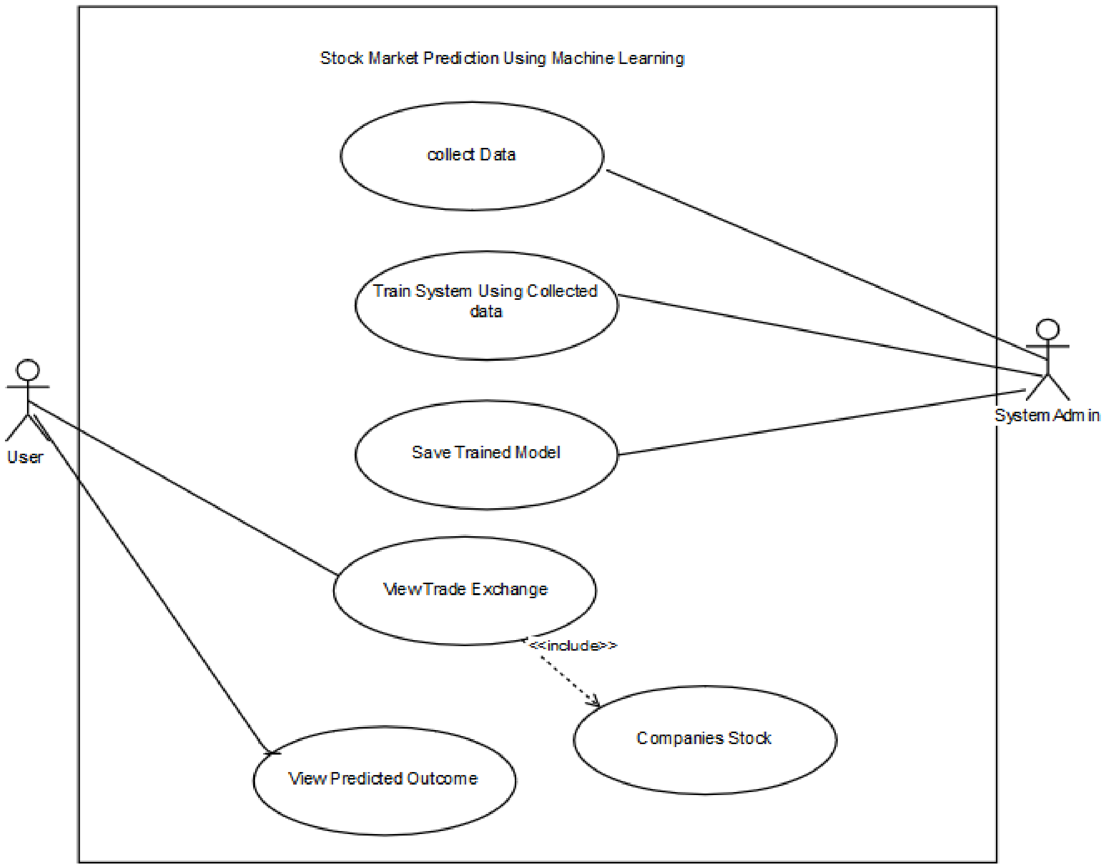


Figure 12:Use Case Diagram of the system

The following steps are performed:

- Data is initially collected from online sources or the stock exchange, either by the system or the admin
- The data is then used to train the system
- Trained model is saved

- User views the trade exchange and stock of a company
- Using the model, closing prices are predicted

5.3 Sequence Diagram

A sequence diagram in a UML is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams typically are associated with use case realizations in the Logical View of the system under development.

The steps performed are as follows:

- User visits the application
- Previously saved model is loaded
- User requests for a company's stock data
- He requests for prediction to be made
- The Stock Market Prediction System trains a model using the data from the database
- The model is saved for further use and closing price is predicted
- Result is displayed along with graph
- Also recommendation Engine recommends what to do.

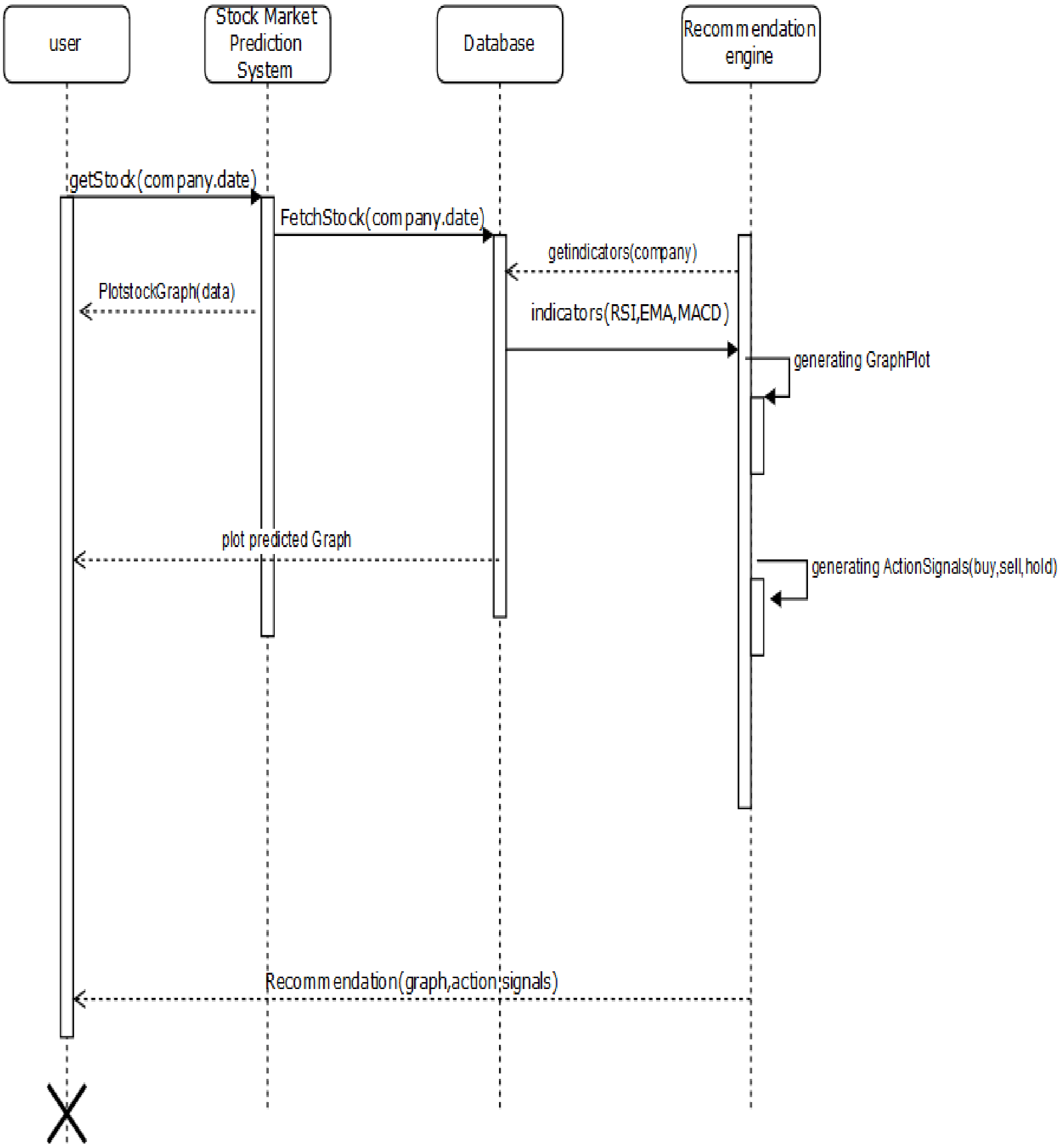


Figure 13:sequence diagram of the system

5.4 Flow Chart

A flow chart is a graphical or symbolic representation of a process. Each step in the process is represented by a different symbol and contains a short description of the process step. The flow chart symbols are linked together with arrows showing the process flow direction.

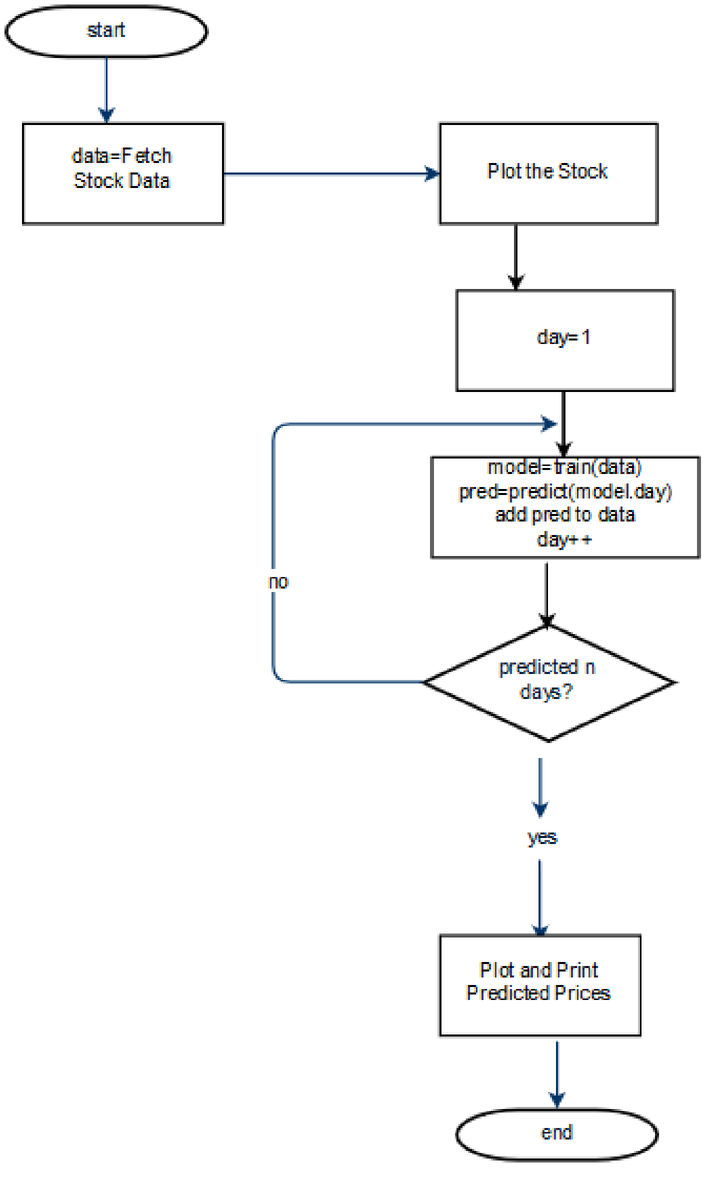


Figure 14: Flowchart Diagram of the system

5.5 Data Sources

This project attempts to predict the stock value with respect to the stock's previous value and trends. It requires historic data of stock market as the project also emphasizes on data mining techniques. So, it is necessary to have a trusted source having relevant and necessary data required for the prediction. We will be using Nepal Stock Exchange website (http://www.nepalstock.com.np) as the primary source of data. This website contains all the details such as: Closing value, highest value, and lowest value, number of shares, increase or decrease in stock values for each financial companies. It also provides the overall performance of Nepal Stock Exchange and performance of companies of different categories. The site is updated on daily basis and it is also a repository for years of stock market data for Nepal.

6. RESULT AND ANALYSIS

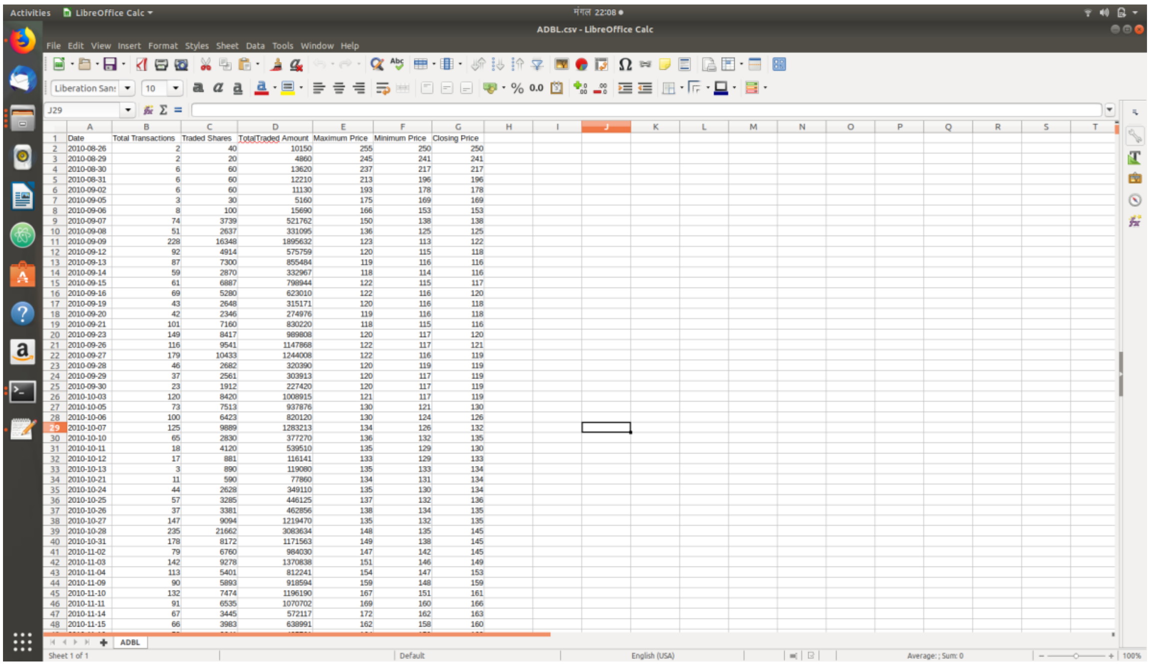


Figure 15:CSV file of scrapped data from NEPSE website

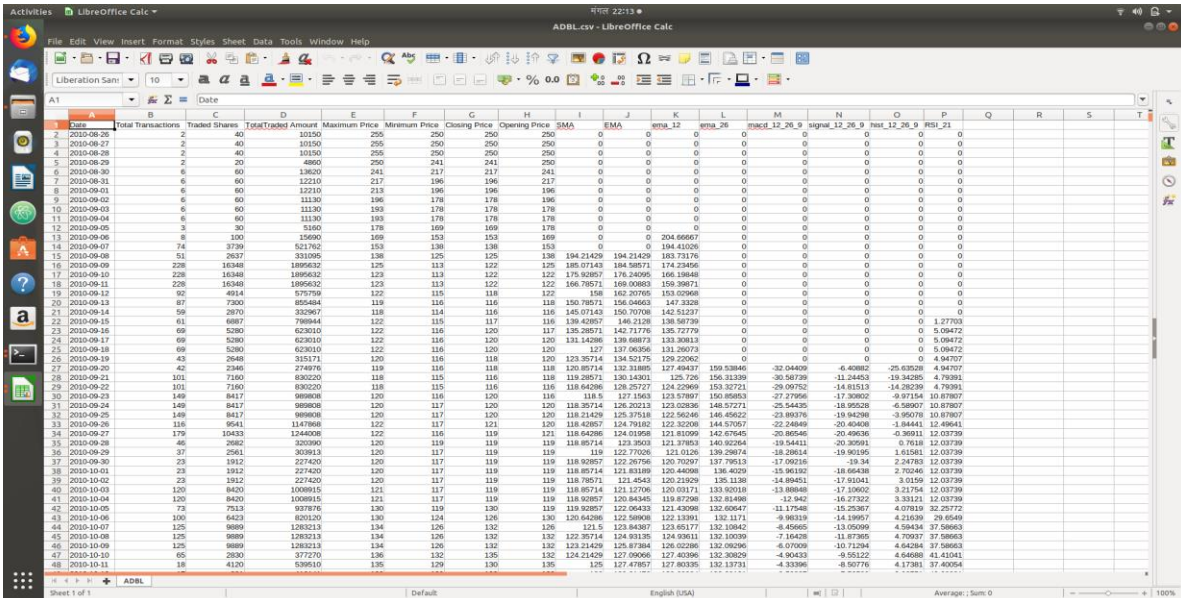
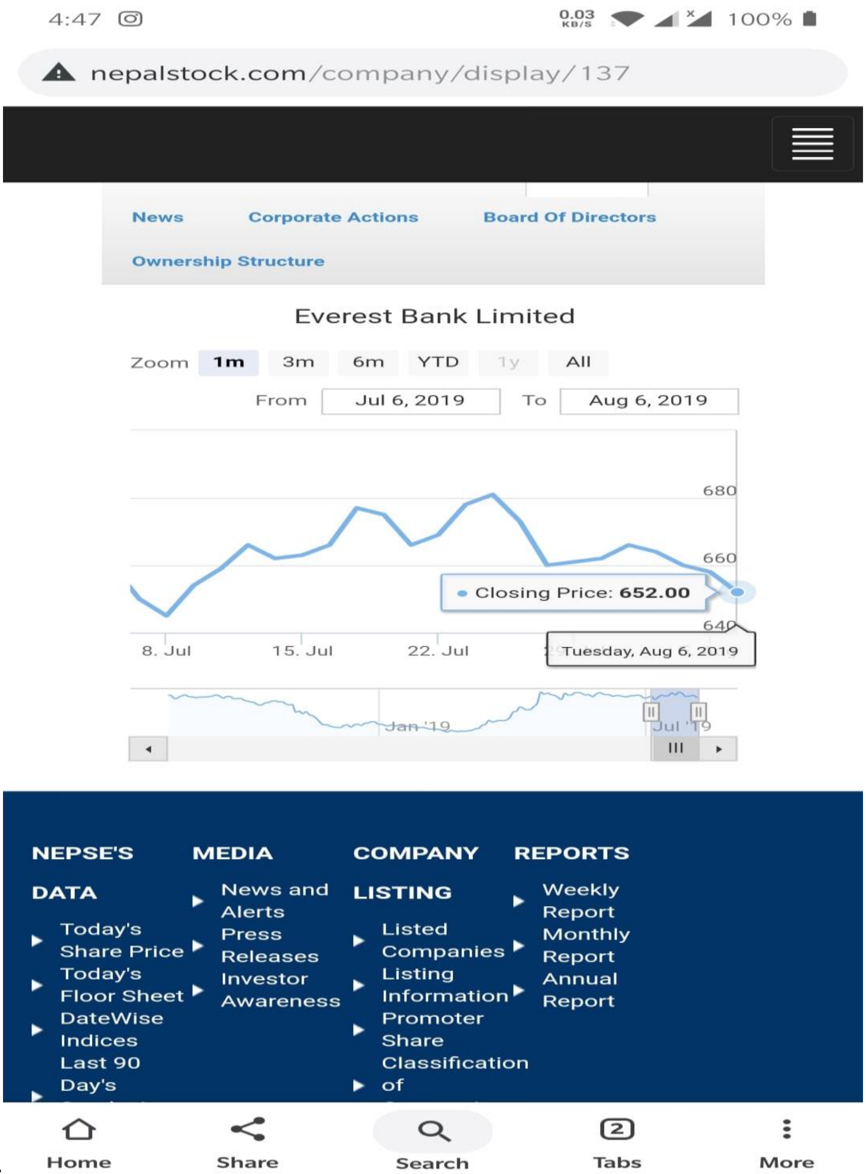


Figure 16:CSV file of scrapped data after being processed for recommendation system

```
4      0.0      0.0          0.0             0.0           0.0      0.0
Epoch 0/90  Current loss: 0.5489546656608582  Testing Accuracy: 98.36528978962467
Epoch 30/90  Current loss: 0.0016501599457114935  Testing Accuracy: 98.1372972952401
Epoch 60/90  Current loss: 0.00033803153201006353  Testing Accuracy: 98.52064905798156
The predicted value obtained from the model is
[[[[653.11896]]]]
Accuracy obtained from the model is
[[[[98.50964]]]]
```

Figure 17:Predicted value of closing price of Everest Bank Ltd. and accuracy of the model as on 5th of August 2019



:

Figure 18:Actual closing price of Everest Bank Ltd. as On 5th of august 2019

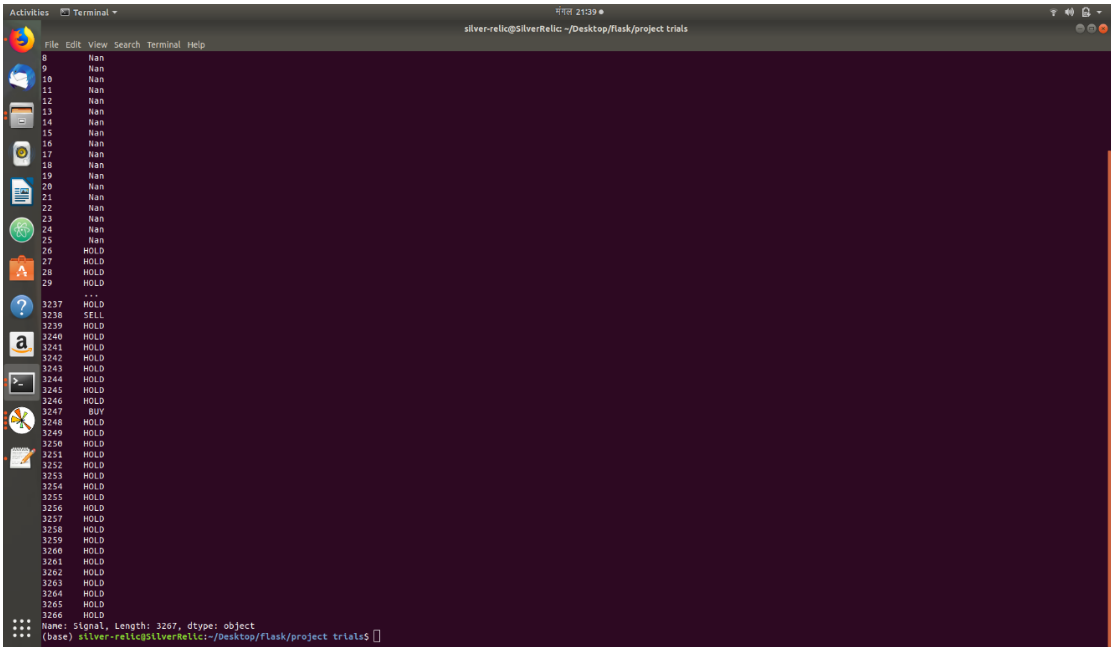


Figure 19:Recommendation System generating Hold, Buy and Sell signals of selected company using MACD

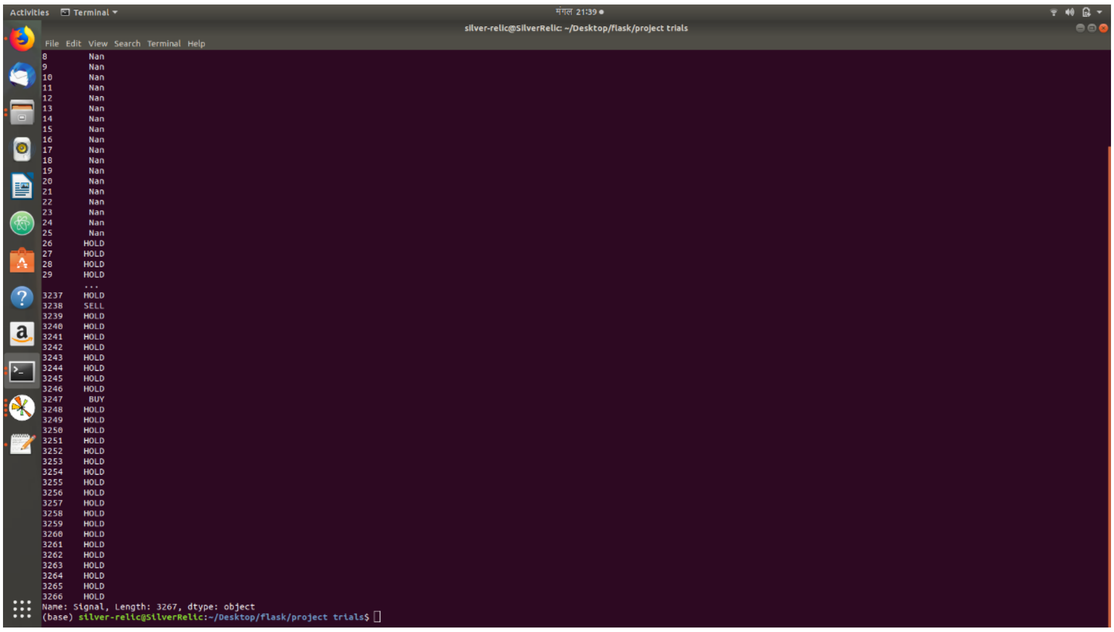


Figure 20: Recommendation System generating Hold, Buy and Sell signals of selected company using RSI

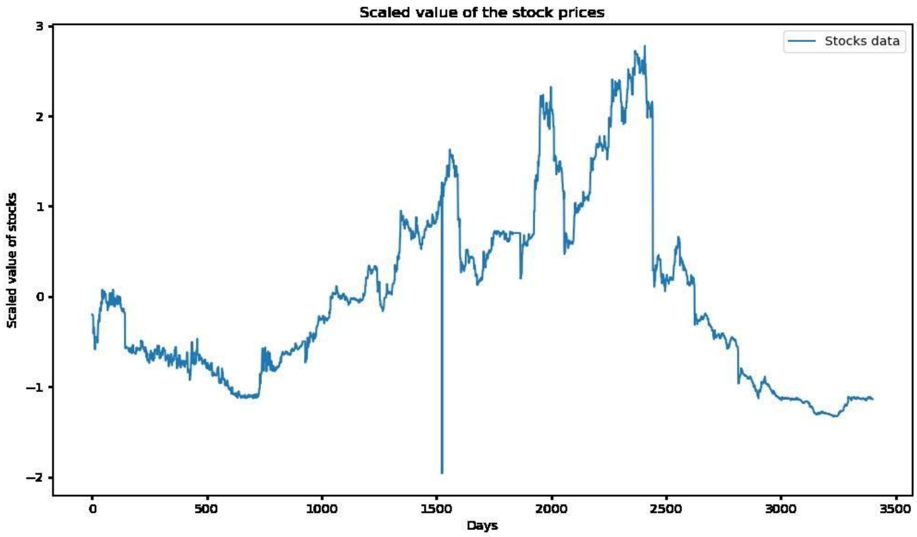


Figure 21:Graph of Everest bank plotted with the help of Scrapped data from NEPSE

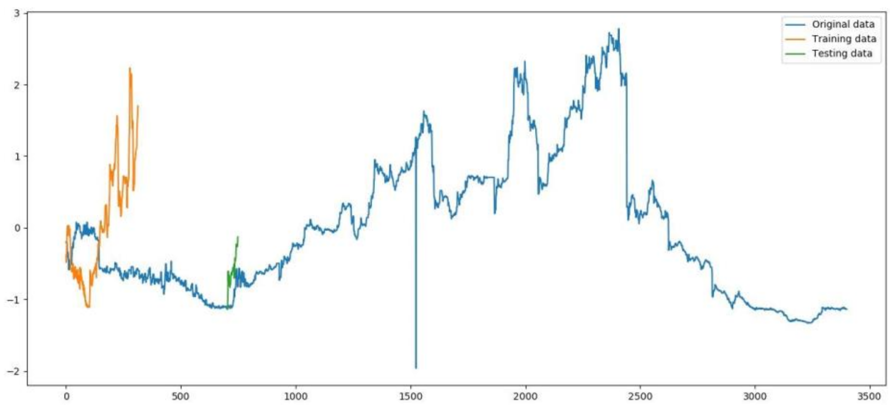


Figure 22:graph of Everest Bank showing the model output during training and testing along with actual value available from scrapped data

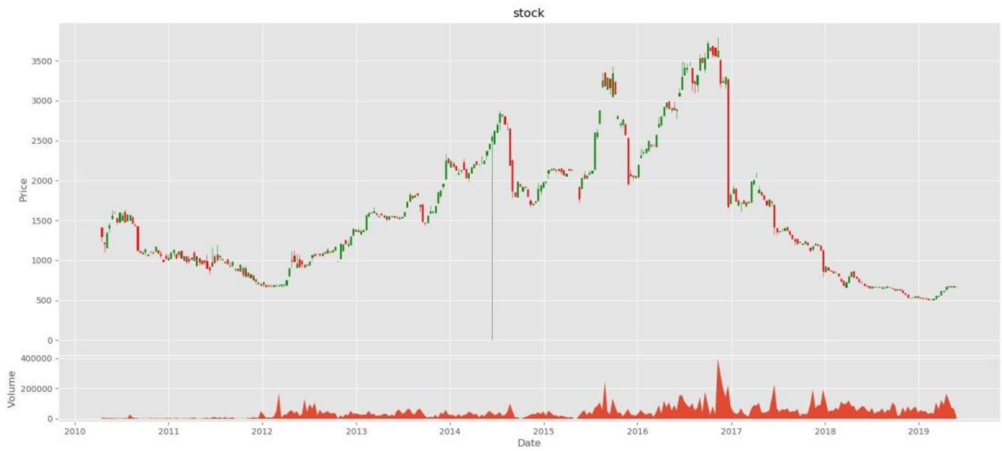


Figure 23:OHLC graph of Everest Bank Ltd.

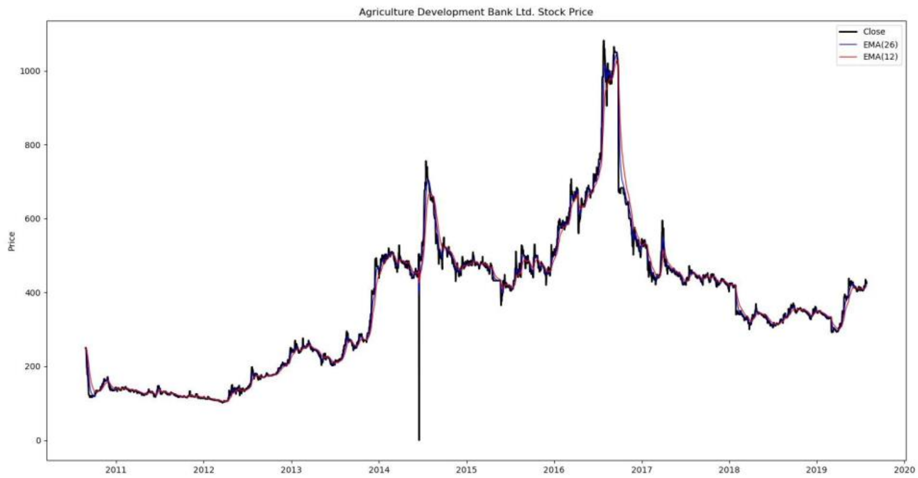


Figure 24:EMA of Agriculture Development Bank Ltd.

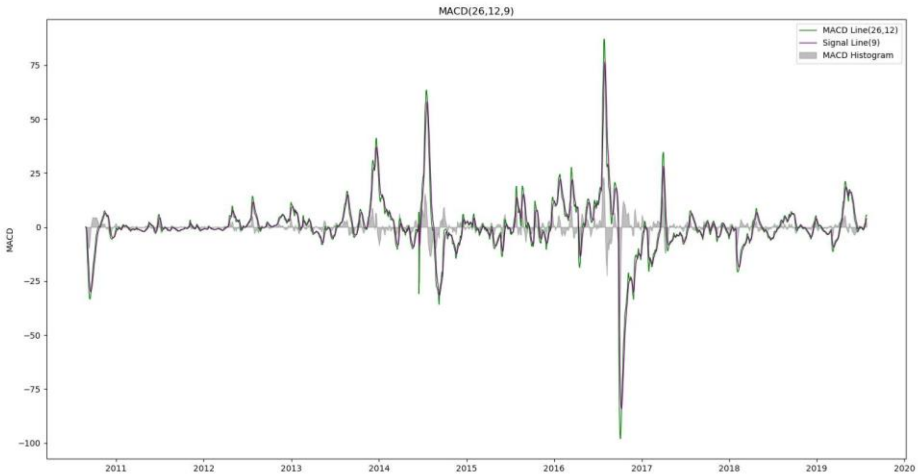


Figure 25:MACD of Agriculture Development Bank Ltd.

6.1 ANALYSIS OF OUR PROJECT

Apart from the previous projects done in the stock market prediction, our project is based on developing a recurrent neural network on keras python machine learning library with tensorflow used as the backend for the built LSTM model. This LSTM model built will give the result of the stock prices of the upcoming days and will represent the prediction accuracy using a graphical method by plotting actual value against the predicted value of our model. From this user will be able to analyze and evaluate their investments done on various banks or organizations. SWOT analysis of our project is shown below:

Strength	It predicts the widely fluctuating stock market of Nepal with the help of the records on how the market has changed from past to the present date.
Weakness	This project is completely based on previous market results i.e. physical factors and doesn’t include socio-political factors or sentiment analysis.
Opportunities	Our project provides a foundation for new prediction models, data analysis and visualization of Nepalese Stock market.
Threats	Our project is that this project has inability to cooperate with sudden fluctuations of stock market due to various reasons such as economic collapse, overthrowing of government, banning of company, terrorism etc.

7. CONCLUSION AND FUTURE WORK

Hence, from this project it can be concluded that stock companies play a vital role in the field of business. Thus, most of the investors can invest money and perform trading with the help of the predicted value. With all the accumulated effort invested in the project, there are reasons to believe that at the end of the semester this project will find itself in a much better shape and quite closer to acceptance than it was.

The main obstacle for the project is the accuracy. We have been constantly using and testing many different algorithms for the prediction process. However we have been able to get satisfactory results using the RNN. We are improving the results by LSTM. In near future we are planning to link the model to a database and fully automate the model that can train, test and update itself which we will then interface with a user friendly web based UI.

8. REFERENCES

[1] R. Wilson and R. Sharda, "Bankruptcy prediction using neural networks", Decision Support Systems, vol. 11, no. 5, pp. 545-557, 1994.

[2] Atsidakos, G. S., Dimitrakakis, E. M., & Zopounidis, C. D. (2011). Elliott wave theory and neuro-fuzzy systems, in stock market prediction: The WASP system. Expert Systems with Applications, 38, 9196–9206

[3] K. Tsai and J. Wang, "External technology sourcing and innovation performance in LMT sectors", Research Policy, vol. 38, no. 3, pp. 518-526, 2009.

[4] K. Han and J. Kim, "Genetic quantum algorithm and its application to combinatorial optimization problem", Evolutionary Computation, 2000, vol. 2, pp. 1354-1360, 2000. [5]"Market Capitalisation Definition from Financial Times Lexicon", Lexicon.ft.com, 2016. [Online]. Available: http://lexicon.ft.com/Term?term=market-capitalisation. [Accessed: 10- Nov-2016

[5]Prashant S Chavan, “Parameters for Stock Market Prediction” et al, Int.J.Computer Technology & Applications, Vol 4 (2) Prototyping Model (2005, August). Retrieved from http://searchcio.techtarget.com/
[6]slideshare.net/anilsth91/stock-market-analysis-and-prediction

DECLARATION

We hereby declare that the Report of the Project Work entitled “STOCK MARKET PREDICTION AND RECOMMENDATION USING MACHINE LEARNING” which is being submitted to the Advanced College of Engineering and Management, Tribhuvan University, in COMPUTER ENGINEERING in the Department of Computer and Electronics Engineering is a bonafide report of the work carried out by us, The material contained in this Report has not been submitted to any University or Institution for the award of any degree.

Sachin Subedi

Saugat Dahal

Smriti Tiwari

Susmita Adhikari

APPROVAL CERTIFICATE

The undersigned certify that the final year project report "STOCK MARKET PREDICTION AND RECOMMENDATION USING MACHINE LEARNING" submitted by Sachin Subedi, Saugat Dahal, Smriti Tiwari, Susmita Adhikari to the Department of Electronics and Computer Engineering in partial fulfilment of the requirements for the Bachelor's degree in Computer Engineering. The project was carried out under special supervision and within the timeframe prescribed by the syllabus. We found the students to be hardworking, skilled, bonafide and ready to undertake any commercial and industrial work related to their field of study and hence we recommend the award of Bachelor of Computer engineering degree.

Er. Dinesh Man Gothe

(Project Supervisor)

DEPARTMENT OF COMPUTER AND ELECTRONICS ENGINEERING

Er. Ajaya Shrestha

Head of Department

DEPARTMENT OF COMPUTER AND ELECTRONICS ENGINEERING, ACEM

COPYRIGHT

ABSTRACT

Lots of Researchers these days are showing interest in the stock market because of its dynamic and unpredictable nature. There are lots of market predicting tools based on traditional predicting techniques but they don't provide the satisfactory result. Machine learning techniques are better than the other techniques because of its ability of nonlinear mapping. That's the main reason that we are trying to develop the application using LSTM for obtaining better and more accurate result. The project is aimed to create an application for the stock market analysis and prediction. A web application is proposed to develop, that analyzes past stock data from certain companies with the help of affecting parameters. The proposed system will also help the stock market enthusiast to determine the values that particular stock will have in upcoming future. The project as a backend development involves programming in Python. In order to train the dataset, we use numpy module of python for the proper implementation of the algorithm. The project is a descriptive model for a particular stock price prediction and predict stock market index. It will be a user friendly system with comfortable and effective user interface.

Keywords:

LSTM, RNN, MACD, RSI

ACKNOWLEDGEMENT

We are grateful to Computer Department of Advanced College of Engineering and Management for providing us with an opportunity of completing an immensely useful project, which helped us to develop creativity and knowledge about the topic of Stock Market Prediction and the importance of Team work in the project. Similarly, we are thankful to HOD Er. Ajaya Shrestha sir and DHOD Er.Raam Sapkota sir for their moral support throughout the project. Special thanks to our project supervisor Er.Dinesh Gothe sir for valuable time, advice, feedback and cooperation throughout this project. We are also very grateful to our families for their help, support, feedback and cooperation during overall completion of our project. We must express our gratitude to all our classmates for their love, support and suggestions in completion of our project.

Last but not the least, we would like to thank everyone who helped in our project.

Sachin Subedi

Saugat Dahal

Smriti Tiwari

Susmita Adhikari

LIST OF FIGURES

LIST OF ABBREVIATIONS

NEPSE	Nepal Stock Exchange Limited
IDE	Integrated Development Environment
LSE	London Stock Exchange
MAPE	Mean Absolute Percentage Error
MSE	Mean Squared Error
SQL	Structured Query Language
SVM	Support Vector Machine
WSJ	Wall Street Journal
LSTM	Long Short Term Memory

Table of Contents