SOFTWARE REQUIREMENTS SPECIFICATION

for

Stock Price Prediction Using Machine Learning

Prepared by : 1.Aditya Baghela (19ESKIT004)
2. Anosh Field (19ESKIT010)
3. Atishay Jain (19ESKIT020)
4. Dhruv Raj Naruka (19ESKIT024)

Submitted to : Mrs. Sanju Choudhary
(Associate Prof.)

June 1, 2023

Contents

1 Introduction

1.1 Purpose

The main objective is to predict the stock prices such that we can make more informed and accurate investment decisions. Our proposed stock price prediction system integrates mathematical functions, machine learning and other external factors. This can be used for the purpose of achieving better stock prediction accuracy and issuing profitable trades.

There are two types of stocks. You may know of intraday trading by the commonly used term “day trading”. Intraday traders hold securities positions from at least one day to the next and often for several days to weeks or months. In order to store past information in the sequence prediction problems. LSTM are more powerful. This is most important in our project because the previous price of a stock , we can build a model that will predict whether the price will go up or down.

1.2 Intended Audience and Reading Suggestions

This SRS is for developers, project managers, users and testers. Further the discussion will provide all the internal, external, functional and also non-functional informations about ”Stock Price Prediction Using Machine Learning”.

1.3 Project Scope

SRS	Software Requirement Specifications
HMS	Hotel Management System
DBMS	Database Management System
Blueprint	A design technical plan
JDBC	Java Database Connectivity
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
EJB	Enterprise Java Beans

Table 1.1: Definitions, Acronyms and abbreviations

Due to the high profit of the stock market, it is one of the most popular investments. People investigated for methods and tools that would increase their gains while minimizing the risk, as the level of trading and investing grew. Two stock exchanges namely- the National Stock Exchange (NSE) and the Bombay Stock Exchange (BSE), which are the most of the trading in Indian Stock Market takes place. Sensex and Nifty are the two prominent Indian Market Indexes. Since the prices in the stock market are dynamic, the stock market prediction is complicated. Time-series prediction is a common technique widely used in many real-world applications such as weather forecasting and financial market prediction. It uses the continuous data in a period of time to predict the result in the next time unit. Many time series prediction algorithms have shown their effectiveness in practice. The most common algorithms now are based on Recurrent Neural Networks (RNN), as well as its special type - Long-short Term Memory (LSTM) and Gated Recurrent Unit (GRU). Stock market is a typical area that presents time-series data and many researchers study on it and proposed various models. In this project, LSTM model is used to predict the stock price.

2 Overall Description

2.1 Product Perspective

Businesses primarily run over customer's satisfaction, customer reviews about their products. Shifts in sentiment on social media have been shown to correlate with shifts in stock markets. Identifying customer grievances thereby resolving them leads to customer satisfaction as well as trustworthiness of an organization.

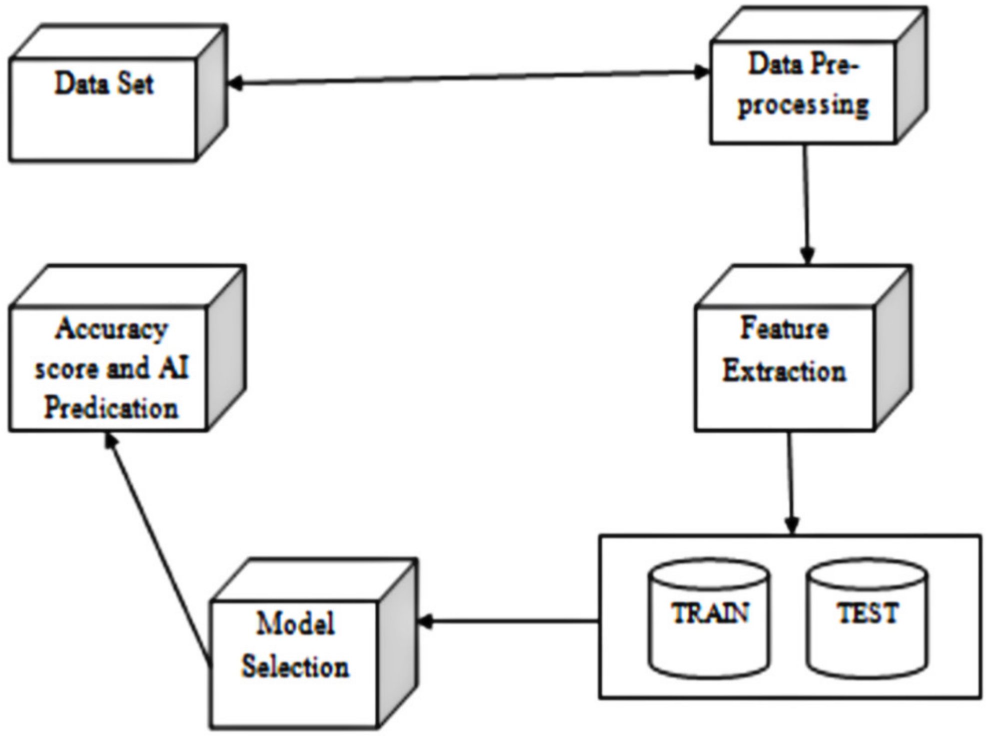


Figure 2.1: Product Perspective

2.2 System Architecture

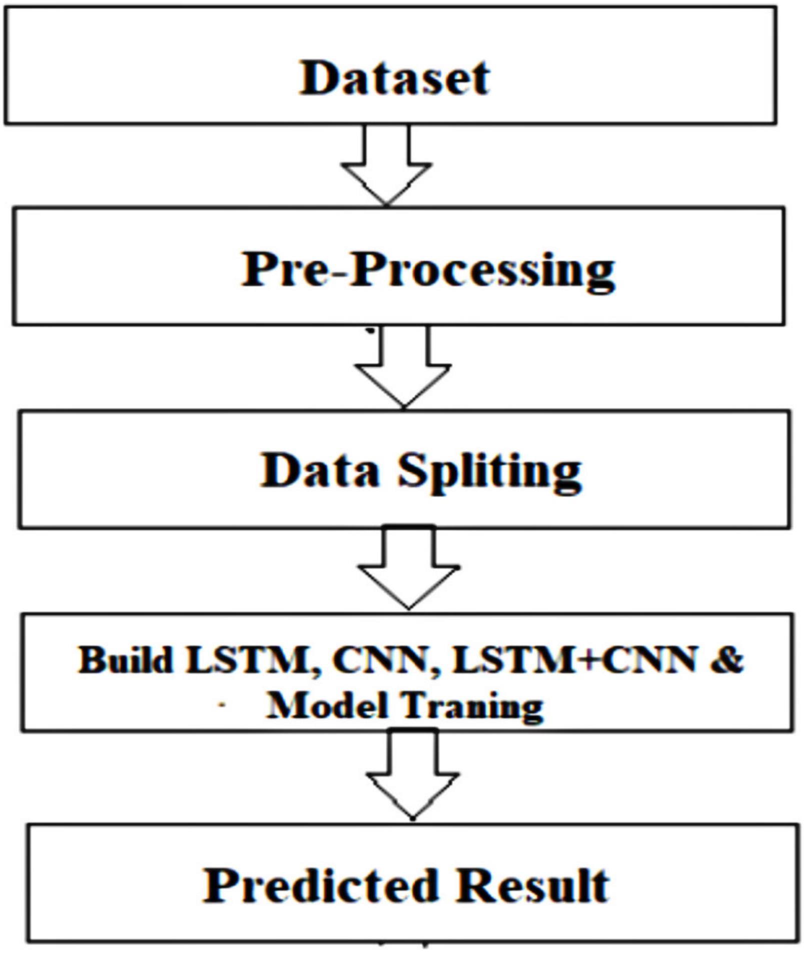


Figure 2.2: System Architecture

2.3 User Classes and Characteristics

There are 3 user Levels in our Stock Price Prediction System:

1. Input as Dataset
2. Pre processing
3. Data splitting

4. Build & Model train Lstm, CNN and Hybrid approach of LSTM+CNN
5. Output as Predicted Result

Attributes such as: price of open, high, low, close, adjusted close price taken from a huge dataset are fed as input to the models for training to pre-process the data techniques like normalization & one hot encoding applied on dataset. After this data is divided into two sets namely training & testing which are ratios of 80:20 respectively. Then, this set is used to train a model using 3 different approaches: LSTM, CNN and Hybrid approach of LSTM+CNNS. Finally, all these modules are evaluated using Root mean square error.

2.3.1 LSTM model

Long Short Term Memory is a kind of recurrent neural network. In RNN output from the last step is fed as input within the present step. It tackled the matter of long-term dependencies of RNN within which the RNN will not predict the word hold on within the long term memory however can offer additional accurate forecasts from the recent info. Because the gap length will increases RNN does not offer an economical performance. LSTM will by default retain the knowledge for a long period of time. It is used for processing, predicting and classifying on the basis of time-series data.

2.3.2 CNN model

Convolution: In the Convolution extract the features from the input image. It gives the output in matrix form. MAX Pooling: In the MAX polling it takes the largest element from a rectified feature map. Dropout: Dropout is randomly selected neurons are ignored during training. Flatten: Flatten feed output into a fully connected layer. It gives data in list form. Dense: A Linear operation in which every input is connected to every output by weight. It is followed by a nonlinear activation function. Activation: It uses sigmoid function and predicts the probability 0 and 1.

2.3.3 Hybrid Approach of LSTM + CNN

In the hybrid approach, the Convolutional Neural Networks (CNNs) offer benefits in choosing sensible options and Long Short-Term Memory (LSTM) networks have proven sensible skills to find out how to learn sequential data. Each approach is reported to produce improved results. CNNs possess to convoluted filters over every input layer so as to get the simple options and CNNs have shown enhancements in computer vision, natural language processing and different tasks [14]. CNN may be a powerful tool to pick out features in order to improve the prediction accuracy [15]. The capabilities of LSTMs in learning data series by considering the previous outputs [16]. The multiple convolutional filters slide over the matrix to produce a new feature map and also the filters have numerous completely different sizes to generate different features. The Max pooling layer is to calculate the most value as a corresponding feature to a particular filter. The output vectors of the Max-pooling layer become inputs to the LSTM networks to measure the long-run dependencies of feature sequences. One in all the benefits of the LSTMs is the ability to capture the sequential data by considering the previous data. This layer takes the output vectors from the dropout layer as inputs. This layer includes a set number of units or cells and also the input of every cell is the output from the dropout layer. The final output of this layer has the same number of units within the network; the outputs from LSTMs are merged and combined in one matrix then passed to a fully connected layer. The array is converted into a single output in the range between 0 and 1 using the fully connected layer, in order to be finally classified using sigmoid function [17].

3 SPECIFIC REQUIREMENTS

3.1 External Interface Requirements

User Interfaces

The user interface for the system shall be compatible with any type of web browser such as Mozilla Firefox, Google Chrome, and Internet Explorer.

Software Interfaces

Web Server

- Yahoo finance, NASDOQ , OS (Windows)

Development End

- Python, Python libraries, HTML, XML, JavaScript, OS(Windows))

Communications Interfaces

The System shall be using HTTP/HTTPS for communication over the Internet and for intranet communications, it shall use TCP/IP protocol.

3.2 Functional Requirement

3.2.1 SCIKIT LEARN

Scikit-learn [21] could be a free machine learning library for Python. It features numerous classification, clustering and regression algorithms like random forests, k-neighbours, support vector machine, and it furthermore supports Python scientific and numerical libraries like SciPy and NumPy.

3.2.2 TENSORFLOW

In the TensorFlow has an open source software library for numerical computation using data flow graphs. Inside the graph nodes represent mathematical formulae, the edges of graph represent the multidimensional knowledge arrays (tensors) communicated between them. The versatile architecture permits to deploy the computation to at least one or many GPUs or CPUs in a desktop, mobile device, servers with a single API. TensorFlow was firstly developed by engineers and researchers acting on the Google Brain Team at intervals Google's Machine Intelligence analysis organization for the needs of conducting deep neural networks research and machine learning, but the system is generally enough to be appropriate in a wide range of alternate domains as well.

3.2.3 KERAS

Keras is [23] a high-level neural networks API, it is written in Python and also capable of running on top of the Theano, CNTK, or. TensorFlow. It was developed with attention on enabling quick experimentation. Having the ability to travel from plan to result with the smallest amount of doable delay is key to doing great research.Keras permits for straightforward and quick prototyping (through user-friendliness, modularity, and extensibility). Supports recurrent networks and convolutional networks, also as combinations of the 2. Runs seamlessly on GPU and CPU.

3.2.4 COMPILER OPTION

Anaconda is [19] free premium open-source distribution of the R and Python programming languages for scientific computing, predictive analytics, and large-scale processes that aim is to modify package management and deployment. Package versions unit managed by the package management system conda.

3.2.5 JUPYTER NOTEBOOK

The Jupyter Notebook is an open-source web application that enables making and sharing documents that contain visualizations, narrative text, live code and equations. Uses include: data , data visualization, data transformation, statistical modeling, machine learning, numerical simulation, data cleaning and much more [24]

4 Other Non-functional Requirements

4.1 Performance Requirements

NF1. Data in the website should be updated according to the stock market.
NF2. Results must return results within 5 seconds.
NF3. Load time of UI Should not take more than 2 seconds.
NF4. Response to customers must be done within 5 minutes.

4.2 Safety Requirements

NF5. Database should be available on the website online.
NF6. Under failure, the system should be able to come back to normal operation when the correct company's name will be available.

4.3 Capacity Requirements

NF7. Systems need to handle at least 1 company's data at one time.
NF8. Not more than 1 customer will be able to see the predicted output.

4.4 Availability Requirements

NF9. Graphs should be generated automatically every time for customers and anytime upon request.

4.5 Software System Attributes

4.5.1 Correctness:

This system should satisfy the normal regular Hotel Management operations precisely to fulfill the end user objectives.

4.5.2 Efficiency:

Enough resources to be implemented to achieve the particular task efficiently without any hassle.

4.5.3 Integrity:

System should focus on securing the customer information and avoid data losses as much as possible.

4.5.4 Flexibility:

System should be flexible enough to provide space to add new features and to handle them conveniently.

4.5.5 Portability:

The system should run in any Microsoft windows environment.

4.5.6 Usability:

The system should provide user manual to every level of users.

4.5.7 Testability:

The system should be able to be tested to confirm the performance and clients specifications.

4.5.8 Maintainability:

The system should be maintainable.

4.6 Requirement Traceability Matrix

The Requirement Traceability Matrix (RTM) reflects the correlation between Non Functional Requirements (NFR) and Functional Requirements (FR). The RTM is a documentation that associates the requirements entirely throughout the validation process. Traceability is regarded to be one of the most important considerations for tracing the requirements.

In the table below we will be tracing the relation between Functional Requirements and Non Functional Requirements.

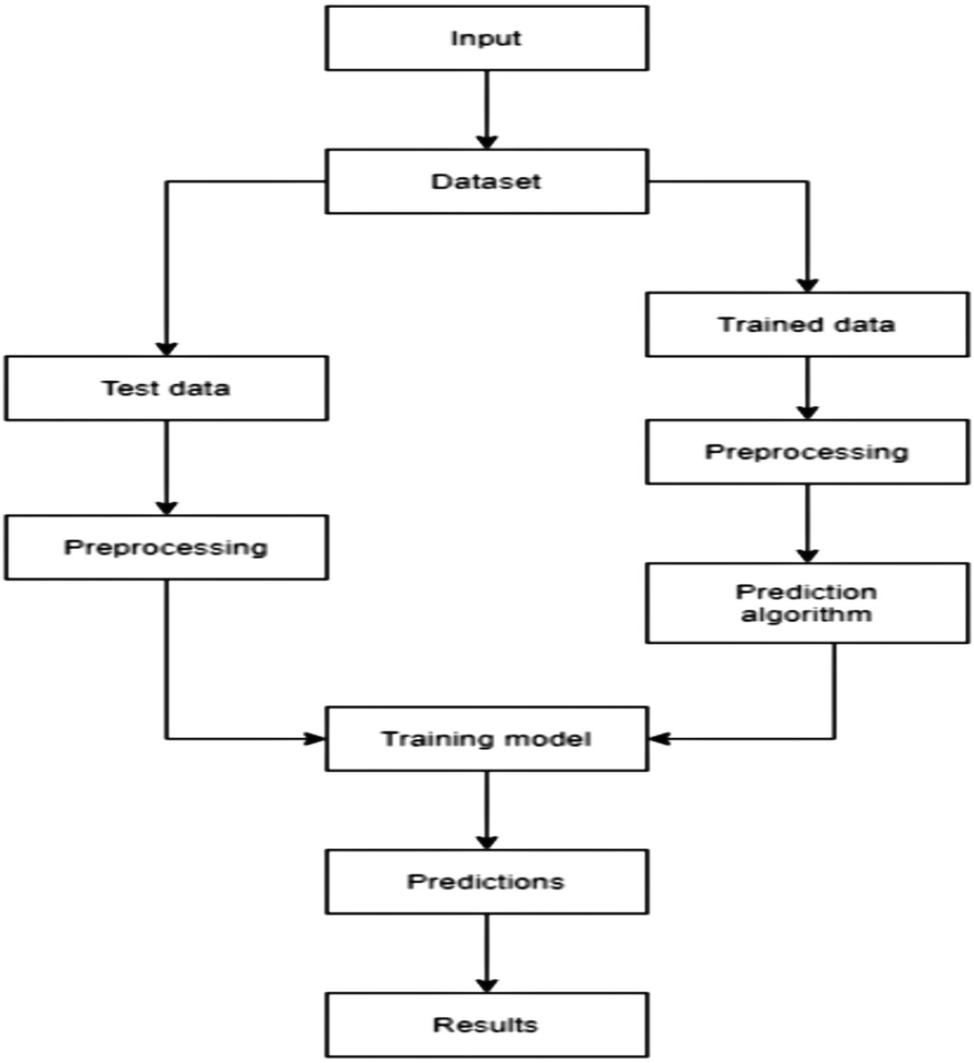
RTM	NF1	NF2	NF3	NF4	NF5	NF6	NF7	NF8	NF9	NF10	NF11	NF12	NF13
FR1			X								X		
FR2	X						X		X				
FR3						X							
FR4				X									

FR5			X										
FR6		X											
FR7		X											
FR8						X							
FR9									X				
FR10	X						X						
FR11					X								
FR12													X
FR13	X								X				
FR14								X				X	

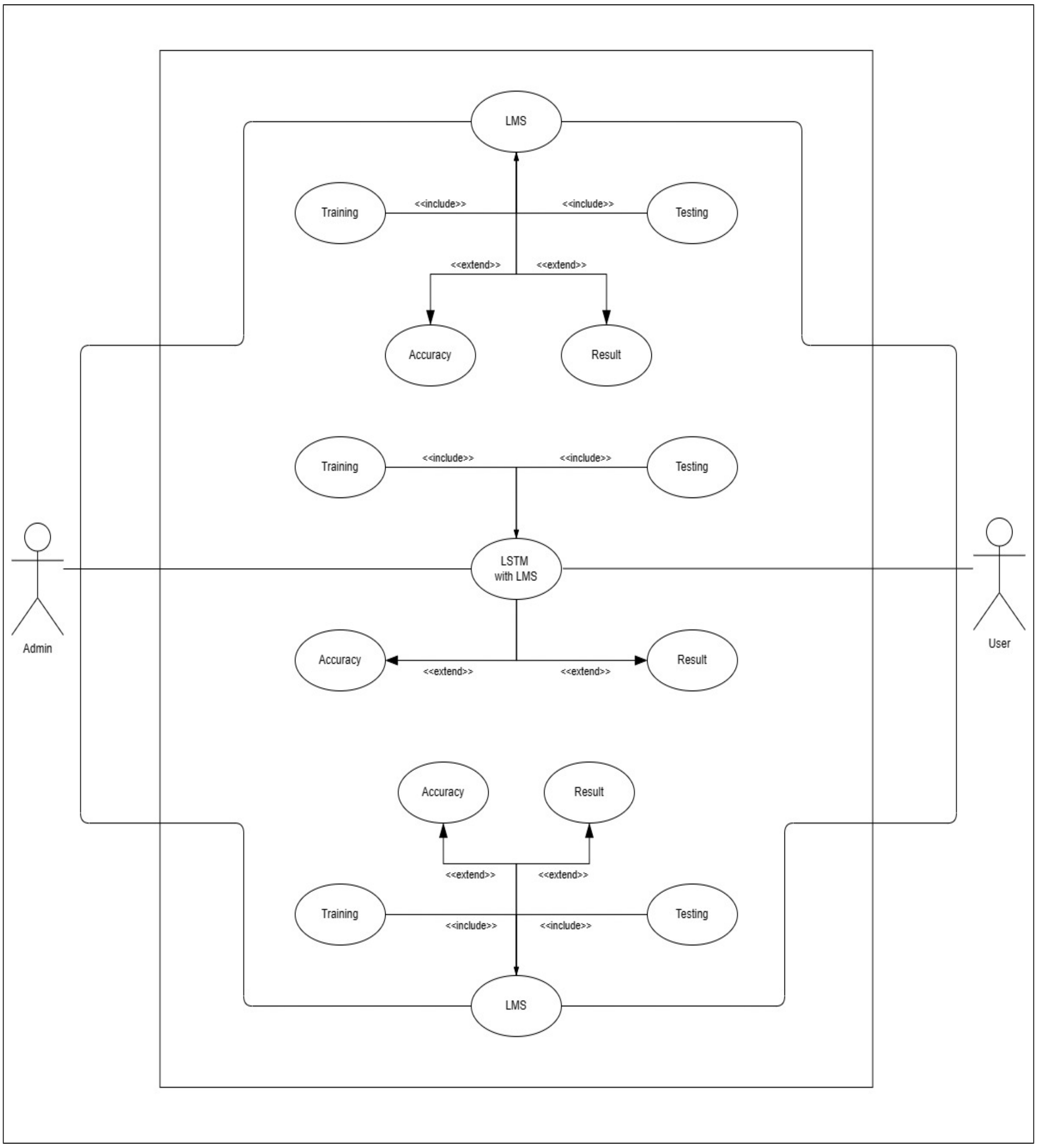
Figure 4.1: Relation b/w Functional and Non-Functional Requirements

5 Diagrams

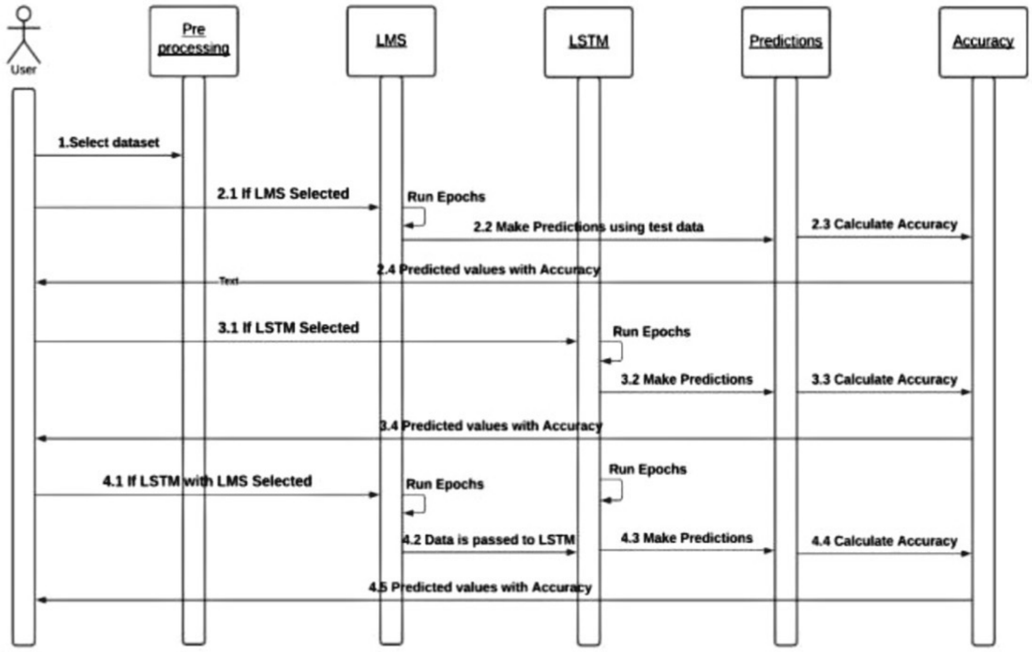
5.1 Structure Chart



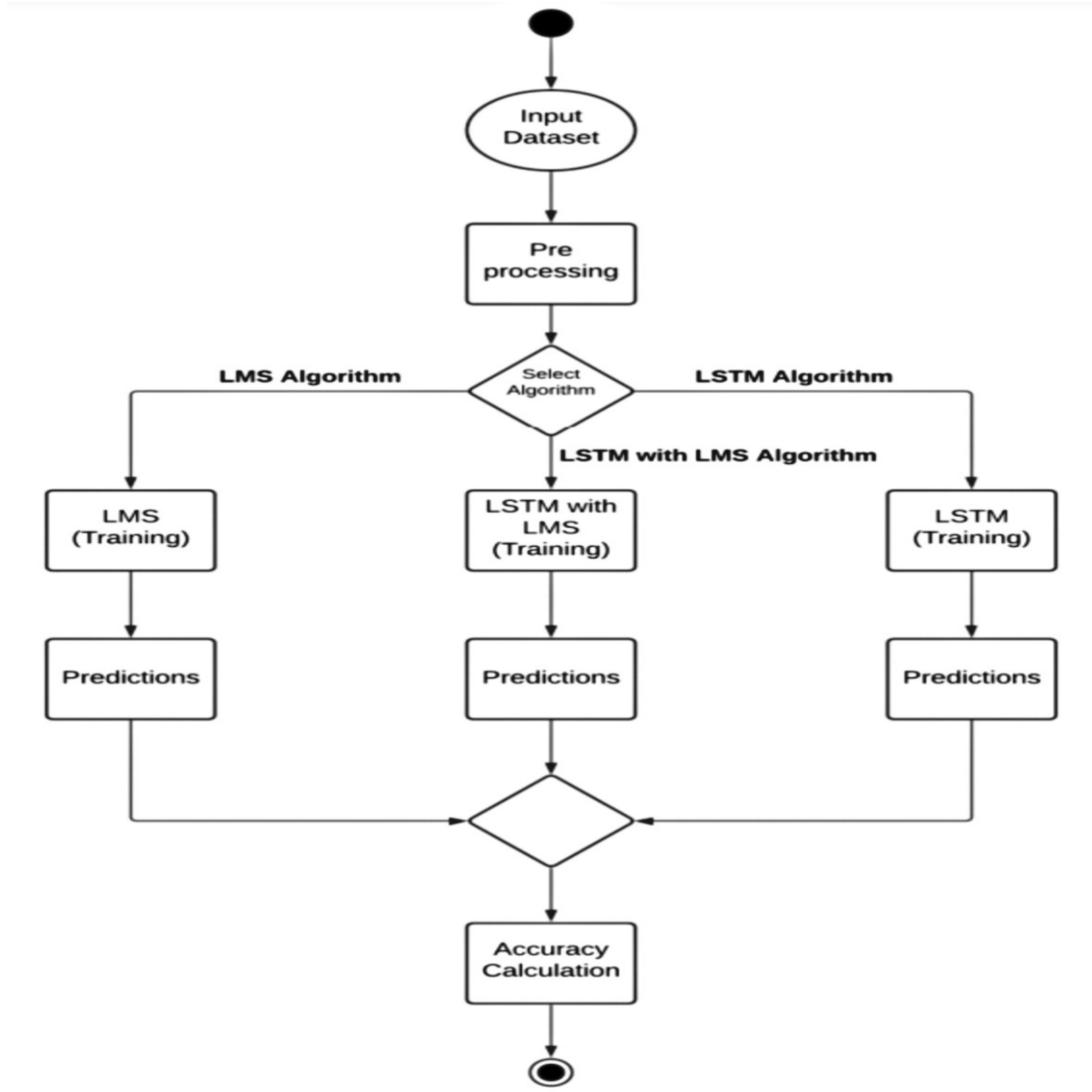
5.2 UML Diagram



5.3 Sequence Diagram



5.4 Activity Diagram



5.5 Component Diagram