**PCTE Institute of Engineering**

**& Technology**

B.Tech (CSE) VIth Semester



SOFTWARE REQUIREMENTS SPECIFICATION

*For*

FINANCIAL MODELLING USING MACHINE LEARNING

Academic Year:- 2019-2023

NAME :- Radhika

ROLL NO. :- 1917631

E-MAIL :- radhikapctebtech19cse@gmail.com

# INTRODUCTION

## 1.1 PURPOSE

The financial market is a dynamic and composite system where people can buy and sell currencies, stocks, equities and derivatives over virtual platforms supported by brokers. The stock market allows investors to own shares of public companies through trading either by exchange or over the counter markets. This market has given investors the chance of gaining money and having a prosperous life through investing small initial amounts of money, low risk compared to the risk of opening new business or the need of high salary career. Machine learning models can likely give us the insight we need to learn about the future of Cryptocurrency. It will not tell us the future but it might tell us the general trend and direction to expect the prices to move. 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).

* Price estimation of Bitcoin
* Price estimation of Ethereum
* Price estimation of gold
* Stock Market evaluation

## 1.2 Document Conventions

Entire document should be justified.

* Convention for Main title
  + Font face: Times New Roman
  + Font style: Bold
  + Font Size: 14
* Convention for Sub title
* Font face: Times New Roman
* Font style: Bold
* Font Size: 12
* Convention for body
* Font face: Times New Roman
* Font Size: 12

## 1.3 SCOPE OF DEVELOPMENT PROJECT

The prediction methods can be roughly divided into two categories, statistical methods and artificial intelligence methods. Statistical methods include logistic regression model, ARCH model, etc. Artificial intelligence methods include multi-layer perceptron, convolutional neural network, naive Bayes network, back propagation network, single-layer LSTM, support vector machine, recurrent neural network, etc. They used Long short-term memory network (LSTM). Statistical methods including Logistic Regression and Linear Discriminant Analysis for Bitcoin, Ethereum & Gold daily price prediction with high-dimensional features achieve an accuracy of 96%, outperforming more complicated machine learning algorithms.

1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

* LSTM 🡪 Long Short-Term Memory
* ATS 🡪 Automated Trading System
* GRU 🡪 Gated Recurrent Unit
* ML 🡪 Machine Learning
* SVM 🡪 Support Vector Machine
* EMH 🡪 Efficient Market hypothesis
* AI 🡪 Artificial Intelligence
* NN 🡪 Neural Networks
* ARMA 🡪 Autoregressive Moving Average
* DRL 🡪 Deep Reinforcement Learning
* LMS 🡪 Least Mean Square
* UML 🡪 Unified modelling Language
* MSE 🡪 Mean Squared Error
* RMSE 🡪 Root Mean Squared Error
* IDE 🡪 Integrated Development Environment
* SRS 🡪 Software Requirement Specification

**1.5 REFERENCES**

Books :-

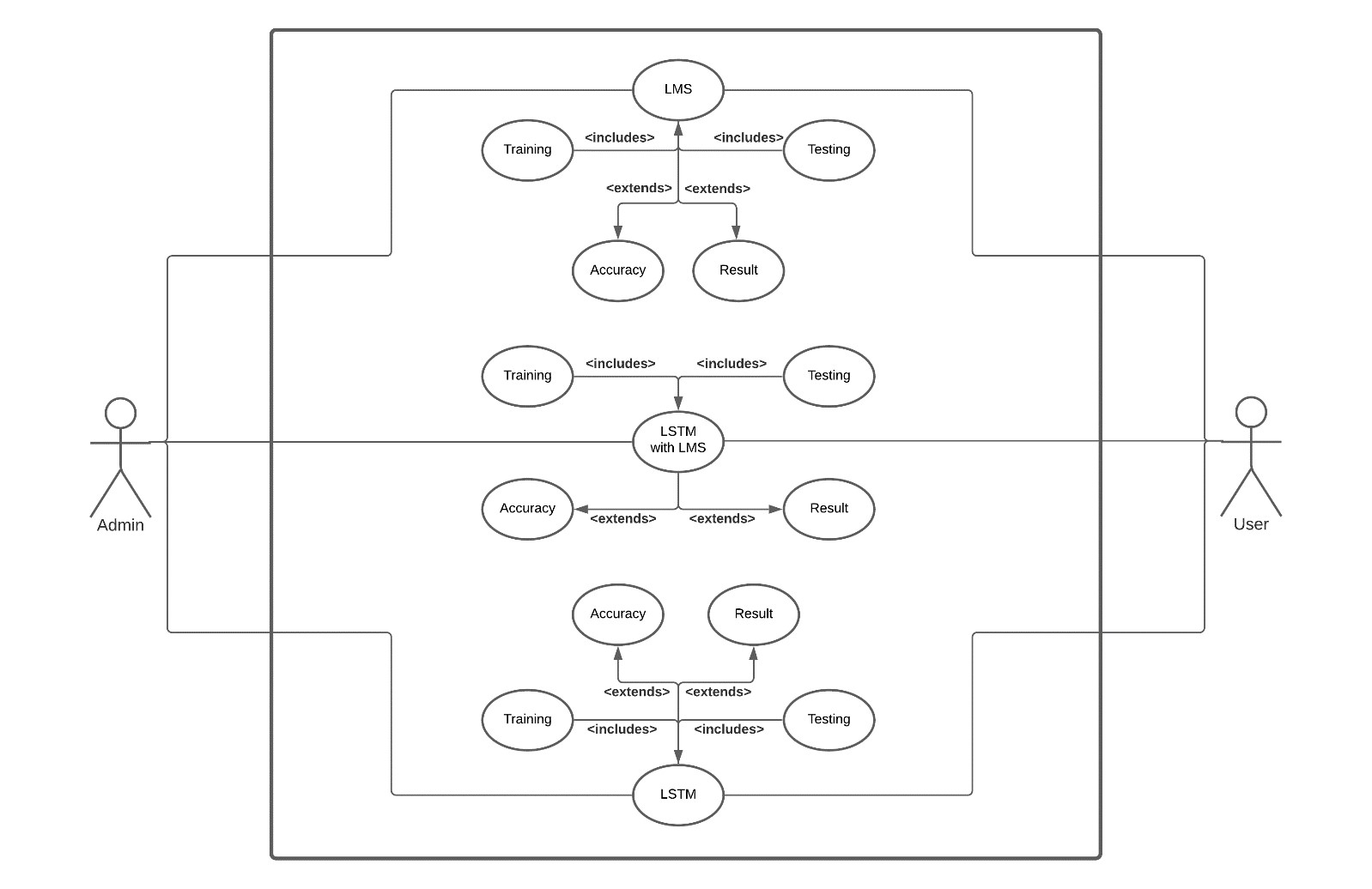
* Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices (ACM Press) by Michael Jackson
* Software Requirements (Microsoft) Second EditionBy Karl E. Wiegers
* Software Engineering: A Practitioner’s Approach Fifth Edition By Roger S. Pressman
* Stock Price Prediction Using LSTM on Indian Share Market by Achyut Ghosh, Soumik Bose1, GiridharMaji, Narayan C. Debnath, Soumya Sen
* James C. Van Horne and George G. C. Parker – The Random-walk Theory: An Empirical Test
* Xiongwen Pang, Yanqiang Zhou, Pan Wang, Weiwei Lin, “An innovative neural network approach for stock market prediction”, 2018

# OVERALL DESCRIPTION

## 2.1 USE CASE DIAGRAM

In the Unified Modelling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

* Scenarios in which your system or application interacts with people, organizations, or external systems
* Goals that your system or application helps those entities (known as actors) achieve.
* The scope of your system

Fig. 1: Using LMS, LSTM and LSTM with LMS in the system

##### **2.2 SEQUENCE DIAGRAM**

##### A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios. Sequence diagrams can be useful references for businesses and other organizations. Try drawing a sequence diagram to:

* Represent the details of a UML use case.
* Model the logic of a sophisticated procedure, function, or operation.
* See how objects and components interact with each other to complete a process.
* Plan and understand the detailed functionality of an existing or future scenario.

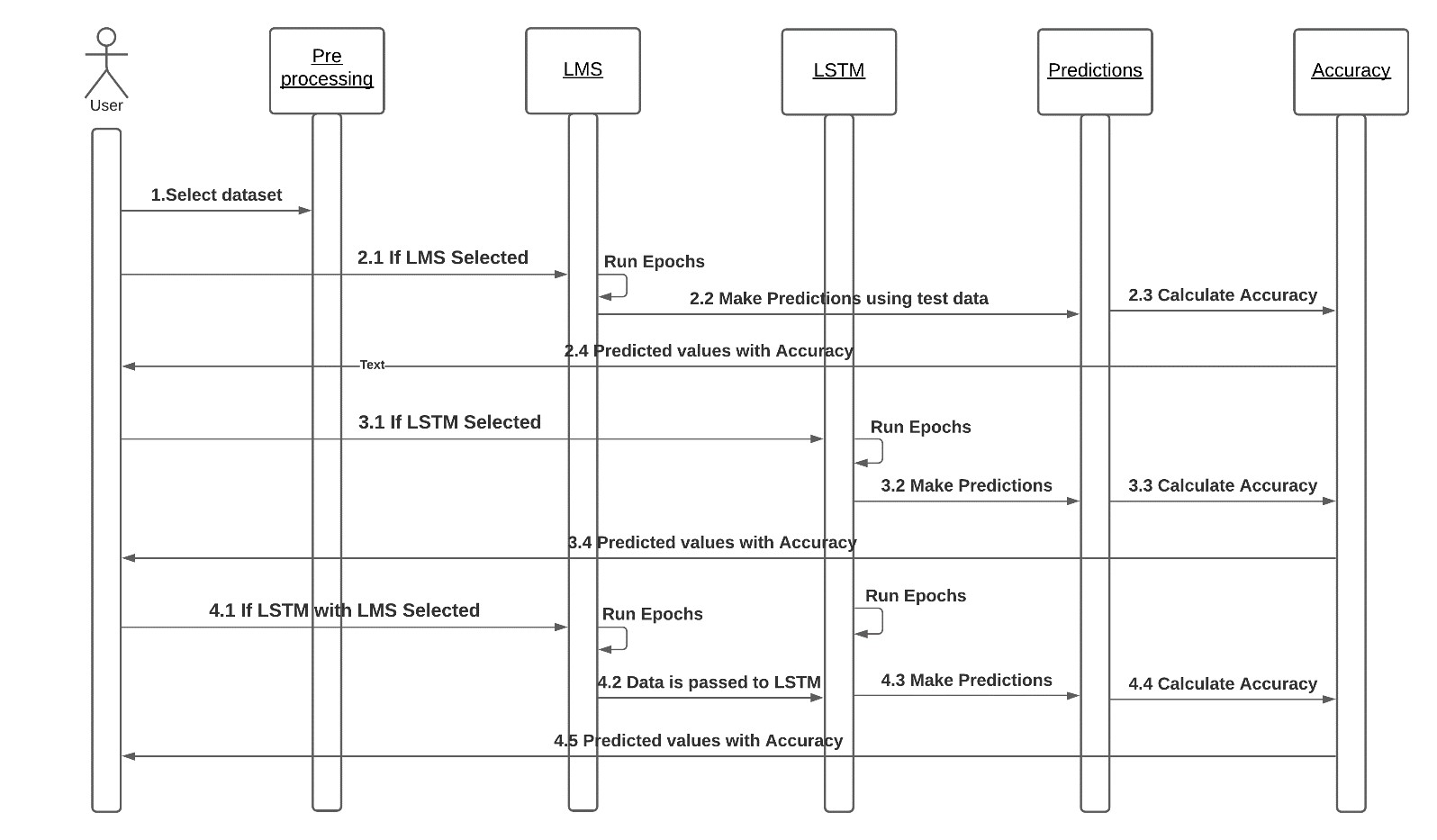


Fig. 2: Execution based on model selection

## 2.3 COLLABORATION DIAGRAM

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

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

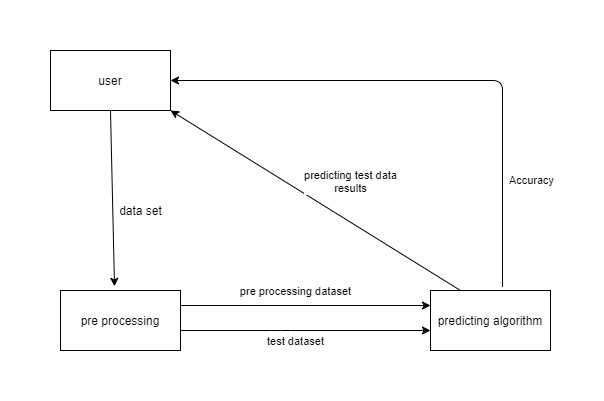


Fig. 3: Data transfer between modules

## COMPONENT DIAGRAM

* Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.
* Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering.
* Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.

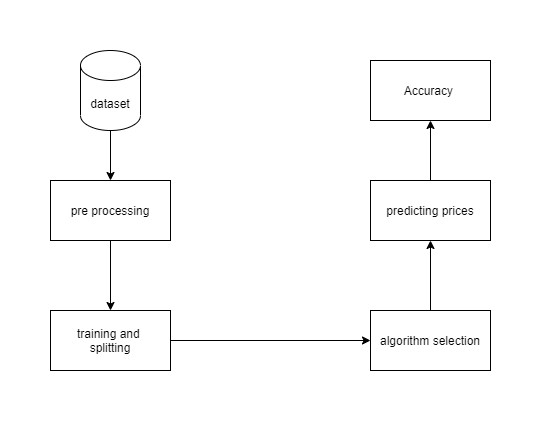


Fig. 4: Components present in the system

# FUNCTIONAL REQUIREMENT:-

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. In this system following are the functional requirements.

Following are the functional requirements on the system:

* The entire control model set must be translated to C output Code.
* Inputs must be models designed using CLAW design components along with standard design components,
* Multiple design models must be processed and the result must be combined to obtain a single output file.

# NON-FUNCTIONAL REQUIREMENT: -

Nonfunctional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviors. They may relate to emergent system properties such as reliability, response time and store occupancy. Nonfunctional requirements arise through the user needs, because of budget constraints, organizational policies, the need for interoperability with other software and hardware systems or because of external factors such as: -

➢ Product Requirements

➢ Organizational Requirements

➢ User Requirements

➢ Basic Operational Requirements

## 4.1 PRODUCT REQUIREMENTS

* **Platform Independency:** Standalone executables for embedded systems can be created so the algorithm developed using available products could be downloaded on the actual hardware and executed without any dependency to the development and modeling platform.
* **Correctness:** It followed a well-defined set of procedures and rules to compute and also rigorous testing is performed to confirm the correctness of the data.
* **Ease of Use:** Model Coder provides an interface which allows the user to interact in an easy manner.
* **Modularity:** The complete product is broken up into many modules and well- defined interfaces are developed to explore the benefit of flexibility of the product. **Robustness:** This software is being developed in such a way that the overall performance is optimized and the user can expect the results within a limited time with utmost relevance and correctness Nonfunctional requirements are also called the qualities of a system. These qualities can be divided into execution quality & evolution quality. Execution qualities are security & usability of the system which are observed during run time, whereas evolution quality involves testability, maintainability, extensibility or scalability.

## 4.2 ORGANIZATIONAL REQUIREMENTS

* **Process Standards**: The standards defined by DRDO are used to develop the application which is the standard used by the developers inside the defense organization.
* **Design Methods**: Design is one of the important stages in the software engineering process. This stage is the first step in moving from problem to the solution domain. In other words, starting with what is needed design takes us to work how to satisfy the needs.

## 4.3 USER REQUIREMENTS

* The coder must request the name of the model file to be processed
* In case of multiple files, the coder must ask the names of the files sequentially.
* The output file must be a C code translated from the model.
* Only a single output file must be created even if multiple input files are provided.
  1. **BASIC OPERATIONAL REQUIREMENTS**

The customers are those that perform the eight primary functions of systems engineering, with special emphasis on the operator as the key customer. Operational requirements will define the basic need and, at a minimum, will be related to these following points: -

* **Mission profile or scenario:** It describes the procedures used to accomplish mission objectives. It also finds out the effectiveness or efficiency of the system.
* **Performance and related parameters:** It points out the critical system parameters to accomplish the mission
* **Utilization environments:** It gives a brief outline of system usage. Finds out appropriate environments for effective system operation.
* **Operational life cycle:** It defines the system lifetime

# SYSTEM REQUIREMENTS

**H/W SYSTEM CONFIGURATION:**

* Processor - Pentium –IV
* Speed - 1.1 Ghz
* RAM - 4GB RAM
* Hard Disk - 20 GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor – SVGA

**S/W SYSTEM CONFIGURATION:**

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