**An Application of Text Classification to Check a Product Review**

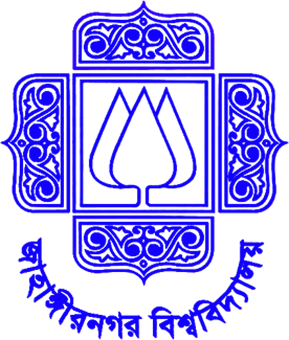
by

Shuvashish Paul Sagar

Roll: 180194

A Thesis Report submitted to the institute of Information Technology

in partial fulfillment of the requirements for the degree of Masters of Science

****

Institute of Information Technology

Jahangirnagar University

Savar, Dhaka-1342

January 2020

**CERTIFICATE**

This is to certify that the thesis entitled **An Application of Text Classification to Check a Product Review** has been prepared and submitted by **Roll: 180194** in partial fulfillment of the requirement for the degree of Master of Science (M.Sc.) in Information Technology on January 16, 2020.

Accepted and approved in partial fulfillment of the requirement for the degree Master of Science (M.Sc.) in Information Technology.

K M Akkas Ali Jesmin Akhter

Chairman Member

Dr. Md. Abu Yousuf Prof. Dr. Md. Mahbub Alam

Member Member (External)

**DECLARATION**

It is hereby declared, that this project or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Shuvashish Paul Sagar

Roll: 180194

**Acknowledgements**

I feel grateful to and wish our profound indebtedness to Md. Fazlul Karim Patwary Professor, Institute of Information Technology (IIT), Jahangirnagar University, Savar, Dhaka. Deep Knowledge & keen interest of my supervisor in the field of Data Science and Big Data influenced me to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project. Heartiest gratefulness to Dr. M, Mesbahuddin Sarker, Director, IIT, JU, who gave me the golden opportunity to do this wonderful project on the topic “An Application of Text Classification to Check a Product Review”, which also helped me in doing a lot of research and I came to know about so many new things I am really thankful to them.

Secondly, I would also like to thank my project-mates who helped me a lot in finalizing this project within the limited time frame.

Finally, I would like to acknowledge my most affectionate and respective parents for being with me always. Many thanks for my beloved wife for being so supportive. Last but not the least, I pay my gratitude to my office colleague for cooperating me while implementing this project.

**Abstract**

In our modern era where internet is ubiquitous, everyone relies on various online resources for shopping. Along with the increase in use of social media platforms like Facebook, Twitter etc. the user review spread rapidly among millions of users within a very short span of time. Consumer reviews on online products plays a vital role in selection of a product. The customer reviews are the measurement of customer satisfaction. This review data in terms of text can be analyzed to identify customer’s sentiment and their demands too.

In this research, we aim to perform four different classification techniques of various reviews available online with the help of concepts pertaining to Artificial Intelligence, Natural Language Processing and Machine Learning. Moreover, a web crawling methodology has been proposed also. Using this web crawling algorithm, we can collect data from any website. We investigate and compare these techniques with the parameter of accuracy using different numbers of training data. Then we find the best classifier method based on accuracy. We also discuss related research areas, open problems, and future research directions for fake news detection.

**Table of Contents**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Title page** | | | |  |  | |  | |
| **Certification** | | | |  |  | | i | |
| **Declaration** | | | |  |  | | ii | |
| **Acknowledgements** | | | |  |  | | iii | |
| **Abstract** | | | |  |  | | iv | |
| **Table of Contents** | | | |  |  | | v | |
| **List of Tables** | | | |  |  | | viii | |
| **List of Figures** | | | |  |  | | ix | |
| **List of Symbols** | | | |  |  | | x | |
| **List of Abbreviations** | | | |  |  | | xi | |
| **Chapter 1** | **Introduction** | | | | | 1 | |
|  | 1.1 Introduction to Log Management | | | | | 1 | |
|  | 1.2 Distributed file system | | | | | 1 | |
|  | 1.3 Access control policy | | | | | 2 | |
|  | 1.4 Limitations of distributed file system | | | | | 2 | |
|  |  | 1.4.1 Limitations of NFS | | | | 2 | |
|  |  | 1.4.2 Limitations of CIFS | | | | 3 | |
|  |  | 1.4.3 Limitations of AFS | | | | 3 | |
|  |  | 1.4.4 Limitations of XFS | | | | 4 | |
|  |  | 1.4.5 Limitations of Samba | | | | 4 | |
|  | 1.5 Motivation of the project | | | | | 4 | |
|  | 1.6 Objectives of the project | | | | | 5 | |
|  | 1.7 Outline of the Project Paper | | | | | 5 | |
|  |  | | | | |  | |
| **Chapter 2** | **Literature Review** | | | | | 7 | |
|  | 2.1 Related works in ISP log and distributed file system | | | | | 7 | |
|  | 2.2 ISP log format and mandate | | | | | 7 | |
|  | 2.3 Hadoop distributed file system (HDFS) | | | | | 10 | |
|  |  | 2.3.1 Salient features of HDFS | | | | 11 | |
|  |  | 2.3.2 Map reduce algorithm | | | | 14 | |
|  |  | 2.3.3 Flow of map reduce algorithm | | | | 14 | |
|  | 2.4 No-sql query processing tool on DFS | | | | | 15 | |
|  | 2.5 Security Solution for Distributed File System | | | | | 16 | |
|  |  | | | | |  | |
| **Chapter 3** | **Proposed System Architecture** | | | | | 17 | |
|  | 3.1 Proposed DFS | | | | | 17 | |
|  | 3.2 Log analysis over DFS | | | | | 19 | |
|  | 3.3 Access control policy implementation | | | | | 20 | |
|  | 3.4 Security features of DFS | | | | | 21 | |
|  | 3.5 Environment setup and optimization | | | | | 22 | |
|  | 3.6 Data protection | | | | | 23 | |
|  | 3.7    Proposed log format for ISP | | | | | 24 | |
|  |  | | | | |  | |
| **Chapter 4** | **Implementation** | | | | | 26 | |
|  | 4.1 Introduction | | | | | 26 | |
|  | 4.2 DFS cluster environment | | | | | 26 | |
|  | 4.3 Configuring master node | | | | | 27 | |
|  |  | | 4.3.1 Host file configuration | | | 27 | |
|  |  | | 4.3.2 Key-based SSH authentication setup | | | 27 | |
|  |  | | 4.3.3 User profile configuration | | | 28 | |
|  |  | | 4.3.4 Set java home | | | 29 | |
|  |  | | 4.3.5 ISP log analytics core site configuration | | | 29 | |
|  |  | | 4.3.6 Synchronization of binary and data directory | | | 29 | |
|  |  | | 4.3.7 Configuration of data block replication parameter | | | 29 | |
|  |  | | 4.3.8 Configuration of yarn as job scheduler | | | 30 | |
|  | 4.4 Configuration of Data Nodes | | | | | 31 | |
|  | 4.5 Apache spark configuration | | | | | 32 | |
|  | 4.6 Implementation of access control using apache ranger | | | | | 33 | |
|  | 4.7 Secured connectivity among Nodes | | | | | 34 | |
|  | 4.8 Setup and configure NFS for benchmarking | | | | | 35 | |
|  |  | | | | |  | |
| **Chapter 5** | **Simulation and Result** | | | | | | 36 |
|  | 5.1 Introduction | | | | | | 36 |
|  | 5.2 Data collection | | | | | | 36 |
|  | 5.3 Monitor HDFS cluster | | | | | | 37 |
|  | 5.4 Experiments with Proposed Method | | | | | | 38 |
|  | 5.5 Comparison between proposed method and map reduce | | | | | | 39 |
|  | 5.6 Comparison with present scenario of ISP log | | | | | | 40 |
|  | 5.7 Monitor access to ISP log analytics using proposed solution | | | | | | 41 |
|  | 5.8 Performance benchmarking of Proposed DFS over NFS and CIFS | | | | | | 42 |
|  | 5.9 Security comparison among NFS,CIFS and HDFS | | | | | | 43 |
|  |  | | | | | |  |
| **Chapter 6** | **Conclusions and Future Works** | | | | | | 44 |
|  | 6.1 Conclusion | | | | | | 44 |
|  | 6.2 Major contribution and recommendation for future works | | | | | | 44 |
|  |  | | | | | |  |
| **References** |  | | | | | | 46 |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| Table 2.1 | ISP user access log | 8 |
| Table 2.2 | Syslog format | 8 |
| Table 2.3 | ISP IPDR log format | 9 |
| Table 3.1 | ISP log analytics Name Node and Data Node information | 18 |
| Table 3.2 | Proposed log format | 24 |
| Table 5.1 | Sample ISP user internet activity log | 36 |
| Table 5.2 | Scala output of clients connected with pop name | 38 |
| Table 5.3 | Security comparison among NFS, CIFS and HDFS | 43 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| Fig. 2.1 | Flow of map reduce algorithm | 15 |
| Fig. 3.1 | Proposed DFS’s network architecture | 17 |
| Fig. 3.2 | Proposed distributed file system | 18 |
| Fig. 3.3 | Proposed security policy management | 21 |
| Fig. 3.4 | Proposed DFS access control policy | 23 |
| Fig. 3.5 | ISP log merge script | 25 |
| Fig. 4.1 | Kernel and Java version of master and slave nodes | 26 |
| Fig. 4.2 | Name and Data Node host file | 27 |
| Fig. 4.3 | User profile configuration | 28 |
| Fig. 4.4 | Spark user profile | 28 |
| Fig. 4.5 | Core site configuration | 29 |
| Fig. 4.6 | HDFS site configuration | 30 |
| Fig. 4.7 | Mapred-site configuration | 30 |
| Fig. 4.8 | Yarn site configuration | 31 |
| Fig. 4.9 | Slave node configuration | 31 |
| Fig. 4.10 | CSV databrick loading configuration | 32 |
| Fig. 4.11 | Apache spark web gui | 32 |
| Fig. 4.12 | Log analytics access control policy’s dash board | 33 |
| Fig. 4.13 | Log analytics user and group information | 33 |
| Fig. 4.14 | IPSec over GRE tunnel configuration between master and ISP | 34 |
| Fig. 5.1 | HDFS cluster monitoring | 37 |
| Fig. 5.2 | Scala output of apache spark framework for MAC based query | 39 |
| Fig. 5.3 | Comparison between proposed method and map reduce (3 Instances) | 40 |
| Fig. 5.4 | User activity monitor | 41 |
| Fig. 5.5 | Read latency of proposed DFS, CIFS and NFS | 42 |
| Fig. 5.5 | Write latency of proposed DFS, CIFS and NFS | 42 |

**List of Symbols**

|  |  |
| --- | --- |
|  | Router |
|  | Switch |
|  | Cloud |
|  | Connectivity |
|  | Server |
|  | Hadoop |
|  | Spark framework |
|  | Firewall |

**List of Abbreviations and Acronyms**

ISP Internet Service Provider

DFS Distributed File System

HDFS Hadoop Distributed File System

ACP Access Control Policy

NTP Network Time Protocol

IoT Internet of Things

BTRC Bangladesh Telecommunication Regulatory Commission

BWA Broadband Wireless Access

IPDR IP Address Data Record

SSH Secure Shell

SHA Secure Hash Algorithm

GUI Graphical User Interface

SQL Structured Query Language

POP Point of Presence

Wi-Fi Wireless Fidelity

WIMAX Worldwide Interoperability for Microwave Access

LTE Long Term Evaluation

IP Internet Protocol

MAC Media Access Control

JDK Java Development Kit

XML Extensible Markup Language

RAM Random Access Memory

API Application Programming Interface

CSV Comma Separated Value

# CHAPTER ONE

**INTRODUCTION**

**1.1 Overview**

In recent times, with the pace of faster growing internet technologies, business companies are facing more challenges and opportunities to provide high quality product or services. People are more likely to purchase from online platforms now-a-days. They share their feelings and feedbacks online. Thus, the number of user reviews is increasing day by day [1]. Online businesses are more likely to get user reviews from different online resources and social medias. With the help of these data, online businesses help themselves to provide high quality product and services. Thus, they can compete with the remaining competitors of the market. With this vast amount of data, it is impossible to analyze all data manually. As a result, a need of an automated procedure comes.

Sentiment analysis is a methodology that aims contextual mining of text to identify subjective information to predict some outcome [2]. User’s satisfaction analysis involves in data mining process. Mostly it works with the Natural Language Processing (NLP), text analysis in form of Opinion Mining (OM) [3]. The term Opinion Mining sometimes considered similar to Sentiment Analysis. Various algorithms have been found that work with sentiment analysis. In different cases, they provide different accuracy levels.

**1.2 Motivation**

Data mining is one of the most interesting topics in research. The lots of data can be used for further work with data mining. The more the model is trained the more accurate it can predict. Moreover, collecting data from various internet resources is not an easy task. The researchers and developers will be helped to collect data from internet using this methodology.

Our text mining research work will be able to re-cycle unused data to work with it in supervised learning mechanism. With this research work we can use the unused data of user review section. With this dataset we will enrich our dataset and thus classifier model. This purpose attracts the researchers much.

**1.3 Challenges**

Earlier magazines, newspapers and other offline resources were used to express views and feedbacks of people. However, with the advancements of internet, the uses of social medias and different online portals and blogs has been used to express satisfaction now-a-days [4]. In some context it is a challenge and opportunity too to grab the user reviews from different online things. The huge data set can be used to detect the Aspect Based Sentiment Analysis (ABSA). Moreover, normal English language is not always used to express thoughts. People use their own native languages to comment on something. So here rises another challenge to analyze any text.

**1.4 Our Proposed System and Contributions**

We present in this paper a procedure to extract and crawl data from any online website or repository. In this work we work with a predefined dataset of amazon user review on electronics products [5]. We study, investigate and compare four machine classifier methods namely, Long-Short Term Memory (LSTM), Support Vector Machine (SVM), Naive Bayes (NB) and Decision Tree (DS). We find the best classifier methods using the parameters of accuracy using different numbers of training data. In this paper, our contribution is to find the best classifier methods based on accuracy. Moreover, we would like to present a web crawling procedure.

**1.5 Problem Statement**

While working with online data we always get data from online resources. This is a predefined data set. It is difficult to work with the same data set from online and then let the model work on a real scenario. We need a methodology from where we can collect data and build a dataset.

Some approaches work on the text for detecting user’s satisfaction and predicts as positive, negative or neutral. Not all mechanisms provide the best accuracy for different data sets. Careful features selection is very important for training of machine learning model. All the detection techniques are capable of detecting text up to some extent but if we rely on single method, not all the times the best accuracy level is assured. So, we have to use the adaptive methods of all possible techniques to detect the possible sentiment of a text. We need to specify that which method or technique is suitable for a particular dataset.

**1.6 Objectives**

Considering the challenges to gather dataset and to get the best accuracy level, objectives of the work were set as below:

1. Propose a system that would collect data from various e-commerce sites using a web crawler mechanism.
2. Assemble found dataset in a mannered way to use later.
3. Use four different text classification techniques to a predefined dataset.
4. Compare accuracy of all outcomes and consider the best in this case.

**1.7 Socio Economic Advantages**

With the increased number of online business, a particular business is facing more and more challenges to compete with the other businesses. In this case, sentiment analysis can help a business a lot. They can improve their services and products accordingly the user reviews. That would not only benefit the businesses but also the consumers. Every individual would get a betterment of this practice. The socio-economic advantages are noted below:

1. To ensure better products.
2. To make sure that the companies are providing the best possible services.
3. To avoid unwanted content and also safe time.
4. To gradually enrich dataset.
5. To help a user to purchase the best product according to user reviews.

**1.8 Assumption and Limitation**

Every website or online resource is not built in the same way. They vary in code architecture and design pattern. So, collecting data with the same mechanism from all sites is not possible. We need various code architecture to crawl from different sites. This is the main limitation of our proposed model. Here, we worked with a site to grab user reviews. And finally, we worked with a dataset found from amazon’s electrical products. There are some more limitations that are discussed below:

1. The web crawler does not work for all website. The code structure is required to change to work for different sites.
2. Languages except English can not be considered.
3. Some comments are seen ambiguous.
4. People can post fake reviews.
5. Ethical issues can not be ensured.
6. Machine cannot detect 100% accuracy of fake content; it can give near 100% accuracy of fake content.
7. Sometimes the accuracy may be shown the wired number like 50%, then we cannot take any decision whether it is fake or not.

**1.9 Organization of Thesis**

Rest of the thesis is organized as follows

**Chapter 2: Literature Review** This chapter provides the researches which are the summary of related works.

**Chapter 3: Proposed Approach and Model** This chapter provides which methods and models we use in our simulation and describes short description and why we use these methods and models.

**Chapter 4: Experiment and Evaluation Result** This chapter provides the readers how we simulate our system and the final result.

**Chapter 5: Conclusion & Future Work** The concluding remarks about this research are presented in this chapter.

Finally, we add some references and coding summary in appendix section.

**CHAPTER TWO**

**LITERATURE REVIEW**

**2.1 Introduction**

Researchers have been attracted by Sentiment Analysis (SA) methodology in the form of text classification or opinion mining. Many researches have been done based on sentiment analysis [6][7][8]. Dang Van Thin∗, Vu Duc Nguyen, Kiet Van Nguyen, Ngan Luu Thuy Nguyen in their research ”A Transformation Method for Aspect-based Sentiment Analysis” on Journal of Computer Science and Cybermetrics, V.34, N.4, 2018 states that predicting only positive, negative and neutral are not enough. They decided to get aspect behind user’s comment. They introduced Aspect Based Sentiment Analysis (ABSA) [1].

In some cases, data mining and text mining are considered as two different fields [10]. Data mining finds interest in discovering interesting features from large dataset whereas text mining deals with textual information gathering [11].

A research based on several data mining and text classification techniques has been researched by Harpreet Kaur, Veenu Mangat and Nidhi [9]. Various methodology gives various accuracy levels. The accuracy level varies on different data sets. If we get better accuracy for detecting outcome from any given text, we need to focus on feature extraction methods. Then we need to find the best classifier model that provide the best accuracy result. In this paper, we are focusing on feature extraction and then finding the best classifier for detecting any given review in terms of text.

**2.2 Related Work**

A paper titled as “A Review on Text Mining in Data Mining” by Yogapreethi.N and Maheswari.S stated text mining methodology is based on information extraction, summarization, topic tracking, classification and clustering. They majorly put emphasis on Knowledge Discovery from Text (KDT). To find semantic relations between concepts Natural Language Processing (NLP) is used [13][14].

Much research has been proposed for web crawling. They propose various technology and methodology to extract data from different web resources. The previous works helped us a lot to generate this methodology to extract data. Web crawler is a bot that can move to different links and html architecture automatically and can grab the html from that. Sometimes it is termed as spider bot too [15].

Yuefeng Li et al [14]: A Text mining and classification method has been used term-based approaches. The major issue is the polysemy and synonymy. The pattern-based methods can outperform the term-based ones. They use fclustering algorithm in their work. It can discover relevant feature based on both positive and negative feedback for using text mining models.

Jian ma et al [16]: The author focused towards the problem by classifying text documents. Most of the part is done in English. While working with non-English language texts it leads to the forbiddance. Ontology-based text mining has been used. It can be expanded to assist in searching a better match between proposals and reviewers.

Chien-Liang Lu et al [17]: This paper summarizes the information about the movie-rating based on sentiment-classification. The feature-based summarizations are used to generate condensed descriptions of movie reviews. The author designed a latent semantic analysis (LSA) to establish product features. It is a way to reduce the size of summary from LSA. They account both accuracy of sentiment classification and response time of a system to design the system by using a clustering algorithm. OpenNLP2 tool is used for implementation.

Yue Hu et al [18]: PPSGen is a new system which was proposed to solicitation of the presentation slides been generated can be used as drafts. It helps them to prepare the formal slides in a faster way for the proprietor. PPSGen system can bring out slides with better quality suggested by the author. The system was developed by the Hierarchical agglomeration algorithm. Tools are a Microsoft Power- Point and OpenOffice. A 200 combo of papers and slides are taken as tests set from the web demonstrate for evaluation process. PPSGen is comparably better than the baseline methods that were evident by the user study.

Xiuzhen Zhang et al [19]: The problem faced by all the reputation system is concentrated by the  
author. However, the reputation scores are universally high for sellers. It is a situation requiring great effort for promising buyers to select trustworthy sellers. Author proposed CommTrust for trust evaluation by feedback comments through mining. A multidimensional trust model is used for computation job. Data set are collected from ebay, amazon. In this technique used a Lexical-LDA algorithm. CommTrust can effectively address the good reputation problem issue and rank sellers are finally by showing definitely through the extensive experiments on eBay and Amazon data.

Dnyanesh G. Rajpathak et al [20]: The challenging task is In-time augmentation of D-matrix  
through the finding of new symptoms and failure modes. Proposed strategy is to construct the fault diagnosis ontology abide with concepts and relationships frequently observed in the fault diagnosis domain. The needed artifacts and their dependencies from the unstructured repair verbatim text were found out by the ontology. Real-life data collected from the automobile domain. Text mining algorithms are used. To establish automatically the D-matrices by the unstructured repair verbatim data that was mined done by the ontology based text mining composed while fault diagnosis. A graph and the graph comparison algorithms have to be generated for each D-matrix.

Jehoshua Eliashberg et al [21]: To forecast the box office performance of a movie at the crenulation point, it’s suitable only if it holds the script and production cost. They extract textual features in three levels particularly genre and content, semantics, and bag-of- words from scripts using domain knowledge of screenwriting, input given by human, and natural language processing techniques. A kernel-based approach is to assess box office performance. Data set are collected from 300 movie shooting scripts. The proposed methodology predicts box office income more exactly 29 percent is reduced mean squared error (MSE) compared to benchmark methods.

Donald E. Brown et al [22]: Rail accidents present image of a valuable safety point for the  
transportation industry in many countries. The Federal Railroad Administration needs the railroads muddled in accidents to submit reports. The report has to be cuddled with default field entries and narratives. A combination of techniques is to automatically discover accident characteristics that can inform a better understanding of the patron to the accidents. Forest algorithm has been used. Text mining looks at ways to extract features from text that takes advantage of language characteristics particular to the rail transport industry.

Luís Filipe da Cruz Nassif et al [23]: In forensic analysis that was computerized with millions of files is usually examined. Unstructured text was found in most of the files performing analyzing process is highly challenging revealed by computer examiners. Document clustering algorithms for the analysis of computers on forensic department seized in police an investigation which was suggested by the author. Variety of mixture of parameters that leads to prompt of 16 different algorithms consider for evaluation. K-means, K-medoids, Single, Complete and Average Link, CSPA are the clustering algorithm are used. Clustering algorithms motivate to induce clusters formed by either relevant or irrelevant document which is used to enhance the expert examiner’s job.

Charu C. Aggarwal et al [24]: Author focused on the Use of Side Information for Mining Text Data. An effective clustering approach was done by the classical partitioning algorithm with probabilistic models which was designed by the author. Dataset used is CORA, DBLP-four-area data set and IMDB. Running time and number of clusters are used as a parameter for analyzing purpose. The results can evident that the usage of side-information can improve the quality of text clustering and classification to sustain a high level of efficiency.

**2.3 Techniques of Text Mining**

There are numerous techniques of text mining. Some of the main techniques are explained here.

**2.3.1 Information Extraction**

Extraction of information is the main step of text mining [25]. The main challenge of here is to gather related and relevant data. Relevant text documents are searched to gather necessary data. These data are then categorized in a simple architecture database in a form of unstructured data. Later the data is served according to the need.

Several data mining techniques are there to extract data. Necessary algorithms and procedures are used to extract the data from database and prepare it to be a structured data.

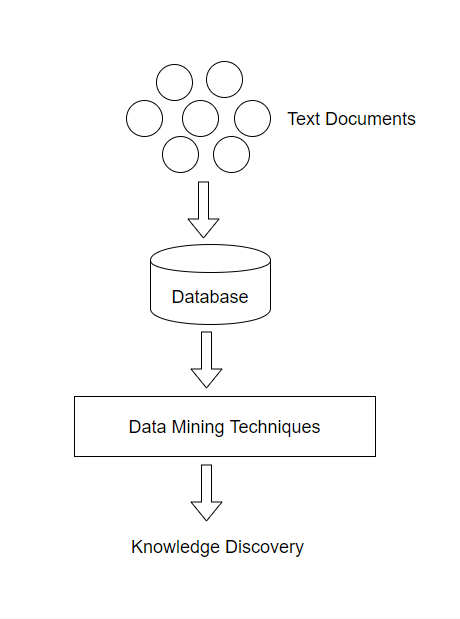


Figure: Information Extraction Process

**2.3.2 Clustering**

Clustering focus towards the similarity measures on different objects and places, it has no predefined class labels. It segregates text into one group and in the same way generates cluster of groups [16]. Words are isolated quickly and weights are assigned to each word. List of classes are generated by using clustering algorithms after calculating similarities.

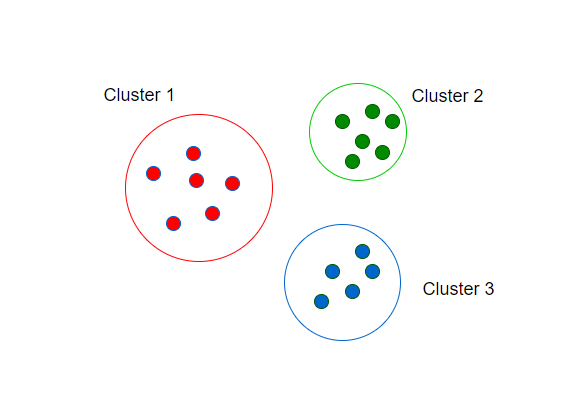


Figure: Data Clustering

**2.3.3 Classification**

Classification is to find the main theme of document by adding Meta and analyzing document. The count of words and from that count decides the topic of the document which was done by the classification technique. It has predefined class label.

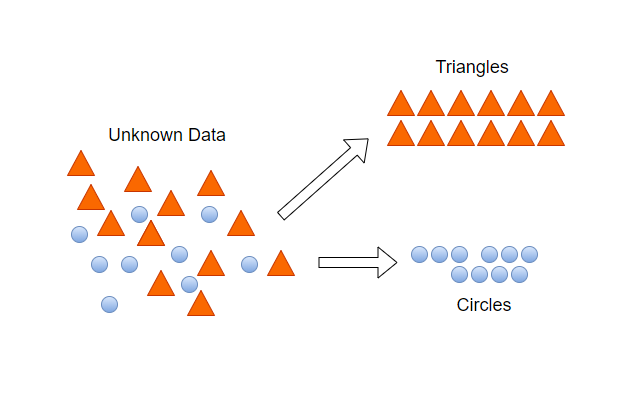


Fig. 2.1: Example of Classification

**2.3.4 Information Visualization**

Instead of searching for extracting the patterns. They provide visual representation for text mining. Text mining used to perform particularly preparation of data, analysis & extraction of data, visualization mapping [24] on Information visualization. Zooming, scaling operations are used for user interaction with the document.

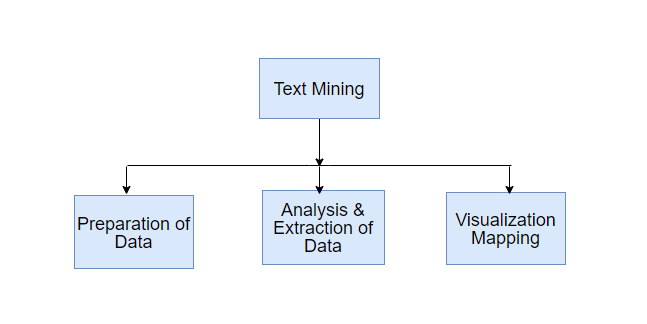


Figure: Visualization

**2.4 Chapter Summary**

Many researchers proposed different models in their research [26] [27] [28] but majority of the researchers use only one or two machine classifier methods in their simulation. But there is a hidden problem that researchers use the classifier method may not the proper method. So, we use four machine classifier methods and find the best method.

**CHAPTER THREE**

**PROPOSED SYSTEM ARCHITECTURE**

**3.1 Introduction**

Data preprocessing, feature selection and classifier are crucial for classification in machine learning. Analyzing the key properties of a dataset with the feature extraction and model best suited for that data to lead to better results. The accuracy is better if we use the feature extraction method properly. Classical machine learning classifier requires numerical values to represent observations of each class. From the data sets in natural language processing tasks (NLP) are usually plain text, as is the case with this research study, there must be a conscientious choice about how to accurately represent the text numerically. Here we use some natural language processing (NLP) techniques to produce vectorized representations of text documents. We use the NLP techniques in the raw data for preparing the data in the numerical representation. In this thesis, we use a method to crawl data from website, sentiment analysis, term frequency-inverse document frequency (TF-IDF). This chapter indicates the preprocessing techniques and feature extraction techniques to refine the textual data, as well as a brief synopsis on the field of sentiment analysis to motivate the idea that sentiment scores can be important features for this study since they provide insight about the motivation and purpose of a piece of text. Finally, this chapter formally introduces the classifier models used in this research: support vector machines, Naive Bayes, Decision Tree, and neural networks.

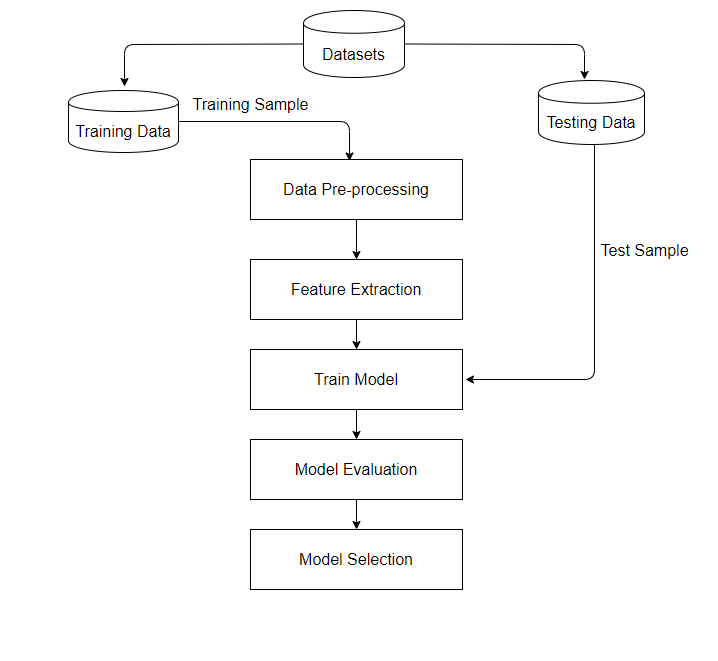


Figure: Work Flow Diagram

**3.2 Proposed Web Crawler**

The Web crawler is an important component of the search engine [29]. It is a program that is used to read the Meta tags specified by the Website creator, find, download, parse content, and store pages in an index file. A crawler usually needs initial Web addresses as a starting point in order to index the information about these Websites. These addresses are the Websites Uniform Resource Locators (URLs), and they are called the seed URLs. After specifying these URLs, the crawler finds the hyperlink text and Meta tags in all pages of the Website until the text finishes [30]. Given a set of seed URLs, the crawler repeatedly removes one URL from the seeds, downloads the corresponding page, extracts all the URLs contained in it, and adds any previously unknown URLs to the seeds [31].It is important to know at this stage that most of the Websites are very large so it can take long time to crawl and index all the data. Furthermore, Websites change their contents frequently, so it is necessary to carefully consider this frequent change as when to revisit the page again in order to keep the index updatability. Further requirements for any crawling system may include: flexibility, high performance, fault tolerance, maintainability, configurability, etc. [30].

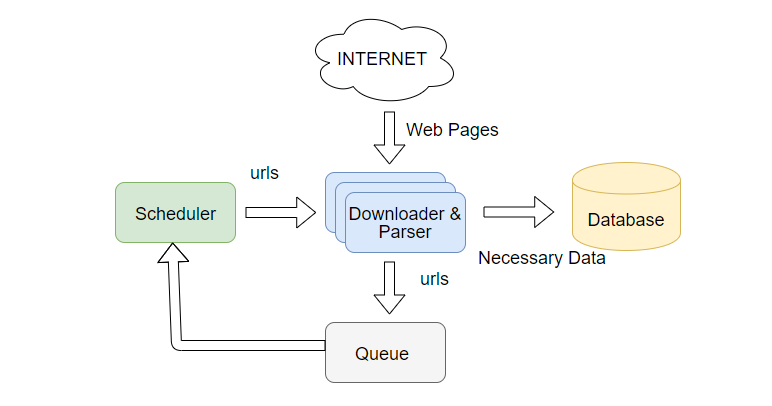


Figure: Web Crawl Procedure

Product reviews from different sites are considered as the dataset in our study. User reviews are mostly found in e-commerce sites. We need to crawl the e-commerce sites basically as we both found the product details and the review there. Most of the e-commerce sites are designed in a similar way. At first, they show the home page. Followed by they show products page. Products page are assembled in a hierarchical way. At first comes the category and subcategories. Then manufacturers are followed by brands, then may be retailers and then the actual product. This practice leads us to a systematic way to crawl the website. The actual flow of data crawling is described as follows:

* Store the main Uniform Resource Locator (URL) address in database.
* Find the main categories and store its URLs to the database.
* Find the subcategories of each main category and assign level to it.
* When all the category pages are stored, we need to find a pattern of product pages URLs.
* Store the product URLs.
* Crawl every product page with cURL request of php. It would fetch the full dom document as a variable.
* Now grab the data you need.

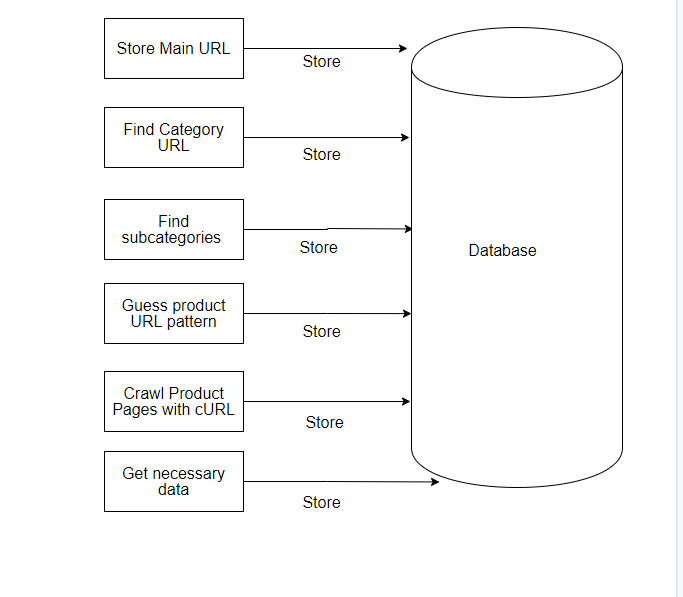


Figure: Web Crawl Flow

**3.3 Data Proprocessing**

Real-world data is usually messy. One of the first steps before performing any data analysis is to clean or refine the data by making the data structured and correct, and removing any discernible noise. The preprocessing steps performed in this study fall into the category because the text is being converted into a convenient and standard form. The very first step in text processing is to tokenize the data, or separate the words. Though common practice is to simply use whitespace and punctuation to delimit words, compound words such as proper nouns (e.g., White House) can lose their meaning when broken up. To overcome this information loss, named entity recognizers can be used to prevent splitting up these tokens.

To standardize the data, the tokens may then be converted to their roots so dier- ent tenses of words can be linked together. However, this task, called lemmatization, requires that the words in each sentence are first tagged with their part-of-speech to determine the root word. Since part-of-speech tagging is sometimes too computation- ally intensive for large documents, a simpler approach called stemming is often used in its place. Stemming aims to remove the suxes of each word to get the root. How- ever, the effectiveness of the stemmer is implementation dependent. For example, the Porter stemmer aims to remove suxes using pattern matching, potentially producing incoherent words or semantically incorrect words (e.g., the stem of ties is ti, and the stem of operator is operate). Finally, to refine the text of the document, articles, pro- nouns, prepositions, and other uninformative words are sometimes filtered out before the core analysis is performed. So these uninformative words are called stop words, generally help in info recovery and document resemblance tasks.

**3.4 Feature Extraction**

**3.4.1 Bag of Words**

A bag-of-words model, or BoW for short, is a method of mining features from text for use in modeling, such as with machine learning algorithms. The method is very simple and flexible, and used in a myriad of ways for extracting features from documents. A bag-of-words is a demonstration of text that describes the occurrence of words within a file. It includes two things:

* 1.A vocabulary of known words
* 2.A measure of the existence of known words

It is named a bag of words, because any data about the order or structure of words in the document is rejected. The model is only concerned with when known words occur in the document, not where in the document.

**3.4.2 N-grams**

An n-gram is simply an order of tokens. In the background of computational linguistics, these tokens are typically words, though they can be characters or subsets of characters. The n simply refers to the number of tokens. N-grams of texts are widely used in text mining and natural language processing tasks. They are fundamentally a set of co-occurring words within a given window and when calculating the n-grams we typically move one word forward (although we can move X words forward in more advanced scenarios). Such as, for the sentence ”I live in dhaka”. If N=2 (known as bigrams), then the ngrams would be:

* I live
* live in
* in Dhaka

So, we have 3 n-grams in this case. Notice that we moved from the->live to live-

>in to in, etc, essentially moving one word forward to generate the next bigram. If N=3, the n-grams would be:

* I live in
* live in dhaka

So, we have 2 n-grams in this case. When N=1, this is stated to as unigrams and this is basically the separate words in a sentence. When N=2, this is named bigrams and when N=3 this is named trigrams. When N>3 this is usually mentioned to as four grams or five grams and so on. In this thesis, we use unigrams, bigrams and trigrams model. Table shows the word level unigrams, bigrams and trigrams for the sentence I live in dhaka.

|  |  |
| --- | --- |
| **Unigrams** | |
| **Token Value** | **Token Sequence** |
| I | 1 |
| live | 2 |
| in | 3 |
| dhaka | 4 |
| **Bigrams** | |
| I live | 1 |
| live in | 2 |
| in dhaka | 3 |
| **Trigrams** | |
| I live in | 1 |
| live in dhaka | 2 |

Table: Word level Unigrams, Bigrams and Trigrams

**3.4.3 TF-IDF**

Tf-idf means frequency-inverse document frequency, and the tf-idf weight is a weight frequently used in information recovery and text mining. This weight is a numerical measure used to estimate how significant a word is to a document in a collection or corpus. The significance increases consistently to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Dissimilarities of the tf-idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document’s relevance given a user query. One of the easiest ranking functions is calculated by summing the tf-idf for each query term; many more sophisticated ranking functions are variants of this simple model. Tf-idf can be effectively used for stop-words filtering in many subject fields including text summarization and classification. Naturally, the tf-idf weight is collected by two terms: the first computes the normalized Term Frequency (TF). The number of times a word seems in a document, divided by the total number of words in that document; the second term is the Inverse Document Frequency (IDF), calculated as the logarithm of the number of the documents in the corpus divided by the number of documents where the exact term appears.

**TF**: Term Frequency, it measures how repeatedly a term occurs in an article. Every document is different in length so, it is possible that a term would look much more times in long documents than smaller ones. So, the word frequency is often divided by the article length (the total number of terms in the article) as a way of normalization:

**IDF**: Inverse Document Frequency, it measures importance of a term. While calculating TF, all terms are measured equally important. Still it is known that certain terms, for example ”is”, ”of”, and ”that”, may appear a lot of times but have less importance. Therefore we need to weigh down the frequent terms while scale up the rare ones, by calculating the following:

As soon as preprocessing is done, the left over text must be transformed into real valued vectors so that the text can be used by a model. One method to produce numerical values for words in an article is to represent each word by its term frequency- inverse document frequency score. The term frequency-inverse document frequency (tf-idf) of a word is used to quantify the importance of a word in a corpus based on how frequently the word shows up in a article and how many other articles also contain the word.

A basic tf-idf scoring function is available in the above Equation. The first term represents the term frequency (tf) of the word t, which is the ratio of the number of times term t appears in a document and total number of terms in the document. The 2nd or another term is the inverse document frequency (idf) which assists to boost rarer, more useful words, and diminish the impact of commonly used non-informative words, like articles and pronouns. Since the idf is computed by taking the logarithm of the total number of documents divided by the number of documents with term t in it with the word oset by 1 to avoid 0 denominators, words that appear in almost all the documents will have a idf close to 0. In contrast, words that appear in only select documents will have higher idf values, thereby growing their tf-idf weights. For example, consider a document containing 100 words wherein the word cat appears 3 times. The term frequency (i.e., tf) for cat is then (3 / 100) = 0.03. Now, assume we have 10 million documents and the word cat appears in one thousands of these. Then, the inverse document frequency (i.e., idf) is calculated as log (10,000,000 / 1,000) =4. Thus, the Tf-idf weight is the product of these quantities: 0.03 \* 4 = 0.12.

**3.5 Classification Models**

**3.5.1 Support Vector Machine (SVM)**

Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification and regression challenges. However, it is mainly used in classification problems. SVM is a discriminative classifier formally defined by a separating hyperplane. In other words, given the labeled training data (supervised learning), the algorithm generates an optimal hyperplane that categorizes new examples. In the two-dimensional space, this hyperplane is a line that divides a plane into two parts, where in each class it lies on each side. In this algorithm, we plot each data element as a point in n-dimensional space (where n is the number of entities we have) with the value of each entity as the value of a particular coordinate. Then, we perform the classification finding the hypo plain that differentiates the two classes very well.

**How does it work**: Suppose we give us a graph of two kinds of labels in the graph as shown in the Figure 3.4. Can we decide on a separation line for the classes? We could have found something similar to the following image Figure 3.5. Separate the two classes enough. Any point that is to the left of the line falls in the orange circle class and, to the right, falls in the blue square class. Class separation, that’s what SVM does. Find a line / hyperplane (in the multidimensional space that separates the outs classes). In short, we will discuss why we write the multidimensional space.

**Handle complex datasets**: The support vector machine (SVM) classifier is a high-performance machine learning algorithm that is based on the relatively simple concept of dividing data into different regions. For example, in the binary classification, the SVM seek to maximize the distance between the data points of the opposite classes and the dividing decision limit. The decision limit, also known as the hyperplane, is formulated as a linear combination of weights in each dimension of the

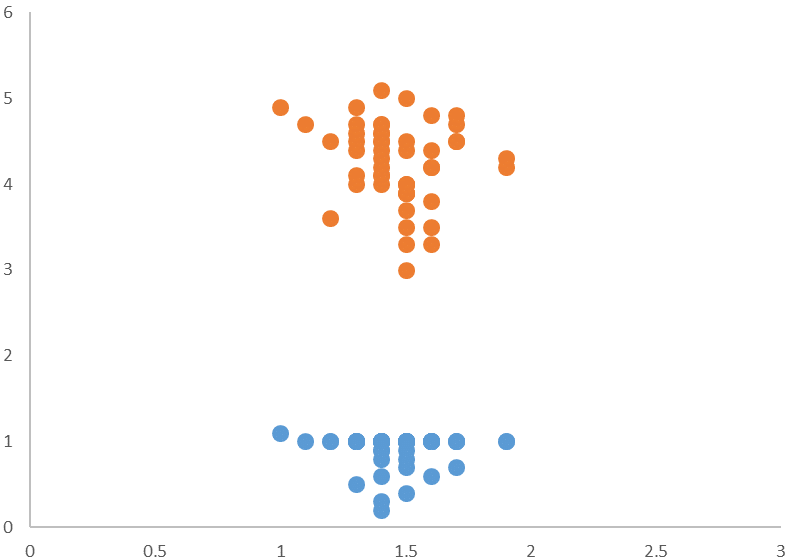


Figure: We can separate the classes by drawing a line between the orange circles and blue circles

input, as shown in the equation 3.4. w is the vector of weights applied to the input x, a support vector, whose length corresponds to the dimensionality of x, and b is the bias, a constant offset.

*wT x* + *b* = 0

The data points closest to the hyperplane are called support vectors, and the shortest distance between these points and the limit is called margin. By maximizing the margin, the probability of classification errors due to noise is reduced, assuming that the test data points come from the same distribution as the training data. One clear advantage of using SVM is the low memory cost: only these support vectors must be kept in memory; the other data points, which are furthest from the hyperplane,

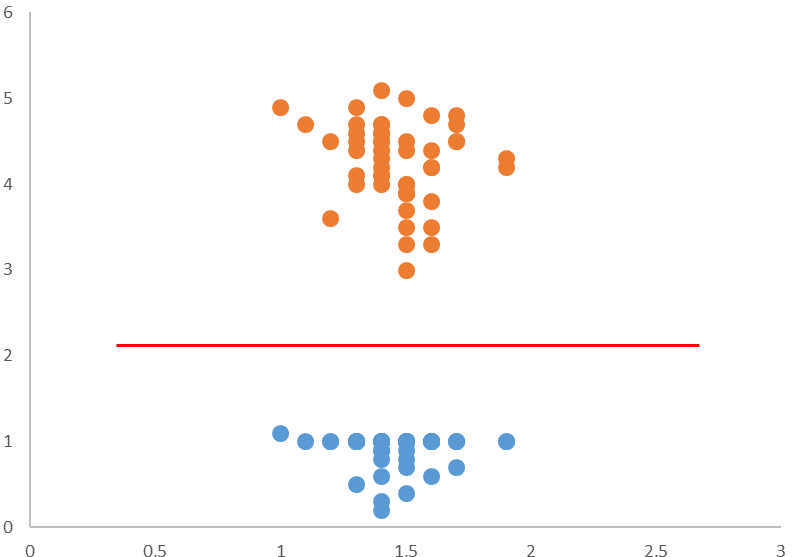


Figure: Sample cut to divide into two classes

should no longer be considered (remember that the KNN algorithm requires that each input be compared with each training observation to determine the most likely class) [32]. However, in most practical applications, the data cannot be separated directly into 2 well-defined regions, especially for complex data sets with much overlap in the space of the original feature. In these cases, it may be possible to allocate the space of the characteristic in a higher dimension where the classes are more easily separable. However, this assignment may be too complex and computationally intensive to apply to large training sets. This computational complexity can be completely avoided by using symmetric kernel functions in the pair of vectors, resulting in a similarity score that is exactly equivalent to the product of points of their inputs when assigned to a higher dimensional space without having to map explicitly the vectors calculating their point product, as shown in equation [33].

K(x, x0) = G(x)TG(x0) = G(x0)TG(x) = K(x0, x)

One example use-case for this higher-order mapping is when the classes data points are radially dependent. As shown in Figure 3.6, the data can be mapped to R3 using the mapping function in equation 3.7.

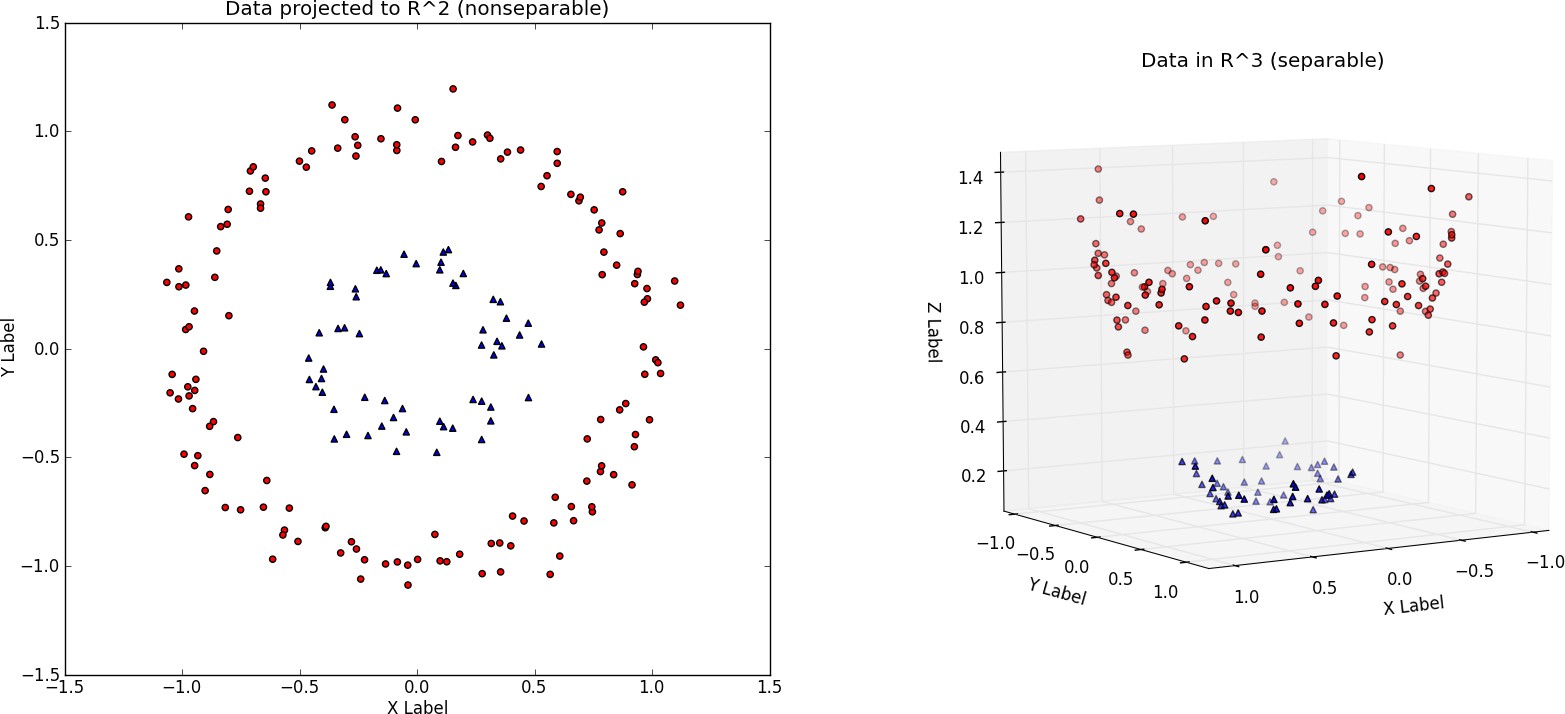


Figure: Visualizing 2-dimensional observations from two radially dependent distributions [3] Left: 2-D plot of the dataset where x is the X-Label and y is the Y-Label Right: Dataset is mapped into R3 where z is the radius from the origin

**3.5.2 Naïve Bayes**

The Naive Bayesian classifier is based on Bayes’ theorem with assumptions of independence among the predictors. A naive Bayes model is easy to build, without an estimation of complicated iterative parameters, which makes it particularly useful for very large data sets. Despite its simplicity, the naive Bayes classifier often works surprisingly well and is widely used because it often exceeds the most sophisticated classification methods. A naive Bayes classifier uses probability theory to classify the data. Bayes ’naive classification algorithms make use of Bayes’ theorem. The key idea of Bayes’ theorem is that the probability of an event can be adjusted as new data is introduced. What makes a naive Bayes classifier naive is its assumption that all the attributes of a data point under consideration are independent of each other. A classifier that classifies fruits in apples and oranges would know that the apples are red, round and have a certain size, but they would not assume all these things at once. Oranges are also round, after all.

A Naive Bayes classifier is not a single algorithm, but a family of machine learning algorithms that make use of statistical independence. These algorithms are relatively easy to write and execute more efficiently than the more complex Bayes algorithms. The most popular application is spam filters. A spam filter examines the email messages for certain keywords and places them in a spam folder if they match. Despite the name, the more data we get, the more accurate a naive Bayes classifier becomes, like when a user marks emails in an inbox for spam. Naive Bayes is a simple, yet effective and commonly-used, machine learning classifier. It is a probabilistic classifier that makes classifications using the Maximum A Posteriori decision rule in a Bayesian environment. It can also be denoted using a very simple Bayesian network. Naive Bayes classifiers have been especially popular for text classification, and are a traditional solution for problems such as spam detection.

**Algorithm**: The Bayes theorem provides a way to calculate the posterior prob- ability, P (A|B), from P(A), P(B) and P (B|A). The Naive Bayes classifier assumes that the effect of the value of a predictor (B) on a given class (A) is independent of the values of other predictors. This assumption is called conditional class independence.

* + P (B|A) is the posterior probability of class (objective) given predictor (at- tribute).
  + P(B) is the previous class probability.
  + P (A|B) is the probability that is the predictive probability of a given class.
  + P(A) is the previous probability of a predictor.

In the ZeroR model there is no predictor, in the OneR model we try to find the best predictor, naive Bayesian includes all the predictors that use the Bayes rule and the assumptions of independence among the predictors.

**Classification**: Now that we have a way of estimating the probability that a given data point falls into a certain class, we need to be able to use this to produce classifications. Naive Bayes handles this in a very simple way; simply choose the ci that has the highest probability given the characteristics of the data point. This is known as the Maximum A Posteriori decision rule. This is because, referring to our formulation of the Bayes rule, we only use the terms P (B|A) and P(A), which are the probability and the previous terms, respectively. If we only used P (B|A), the probability, we would be using a maximum Probability decision rule.

**Multinomial**: We use here the basic multinomial model using Naive bayes. It is used for discrete counts. Such as, we have a text classification problem. Here we can consider Bernoulli’s essays, which is one step further and instead of ”word that appears in the document”, we have ”count how often the word appears in the document”, and it can be considered as number of times that the result number xi on the n tests is observed.

**3.5.3. Decision Tree**

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

A decision tree is a flowchart-like structure in which each internal node represents a “test” on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.

Tree based learning algorithms are considered to be one of the best and mostly used supervised learning methods. Tree based methods empower predictive models with high accuracy, stability and ease of interpretation. Unlike linear models, they map non-linear relationships quite well. They are adaptable at solving any kind of problem at hand (classification or regression). Decision Tree algorithms are referred to as CART (Classification and Regression Trees).

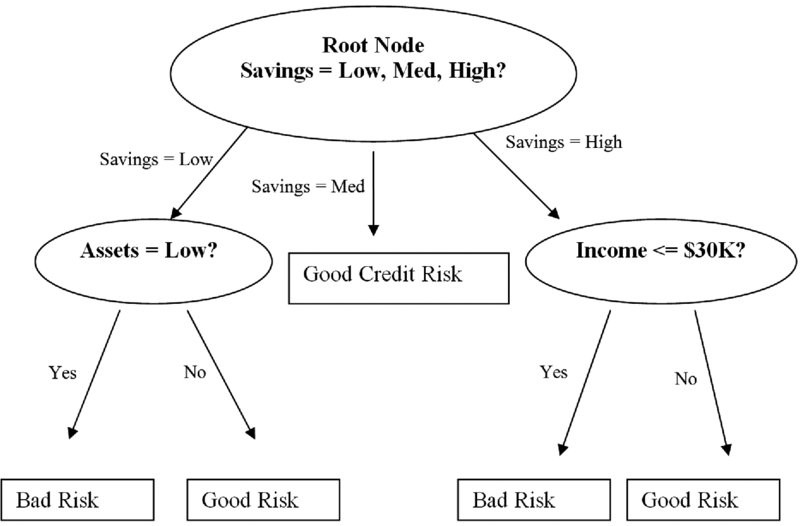


Figure: Decision Tree Classifier

3.5.4 Long Short Term Memory (LSTM)

LSTM neural networks, which represent long-term long-term memory, are a par- ticular type of recurrent neural networks that received much attention recently within the machine learning community. In a simple way, LSTM networks have some internal contextual status cells that act as long-term or short-term memory cells. The output of the LSTM network is modulated by the state of these cells. This is a very important property when we need the prediction of the neural network to depend on the historical context of the inputs, instead of just the last input.

As a simple example, consider that we want to predict the next number of the following sequence: 6 ->7 ->8 ->?. We would like the next exit to be 9 (x + 1). However, if we provide this sequence: 2 ->4 ->8 ->?, We would like to obtain 16 (2x). Although in both cases, the last current entry was number 8, the result of the prediction must be different (when we take into account the contextual information of the previous values and not only the last one).

LSTMs are RNNs with special memory units (also known as cells) that can main- tain information selectively over a long period of time [37]. Instead of consisting of a single repeating layer, as in the case of an RNN, the vanilla LSTM has a memory unit composed of four special layers: three sigmoid layers and one tanh layer [38]. All the sigmoidal layers produce values between 0 and 1, and the tanh layer puts values between -1 and 1. Together, these layers help the cell forget, remember, update and produce information.

As shown in the LSTM in Figure 3.11 and Figure 3.12, the first layer is a sigmoid layer whose input is the concatenation of the output from the previous state (ht − 1) and the current input (xt). In LSTMs, the sigmoidal layers that bind with dot multiplication operators act as gates that let information pass when their outputs are non-zero values. The first sigmoid layer uses (ht − 1) and (xt) and forms a gate in the incoming value stream of the previous cell state (the upper left corner of each cell

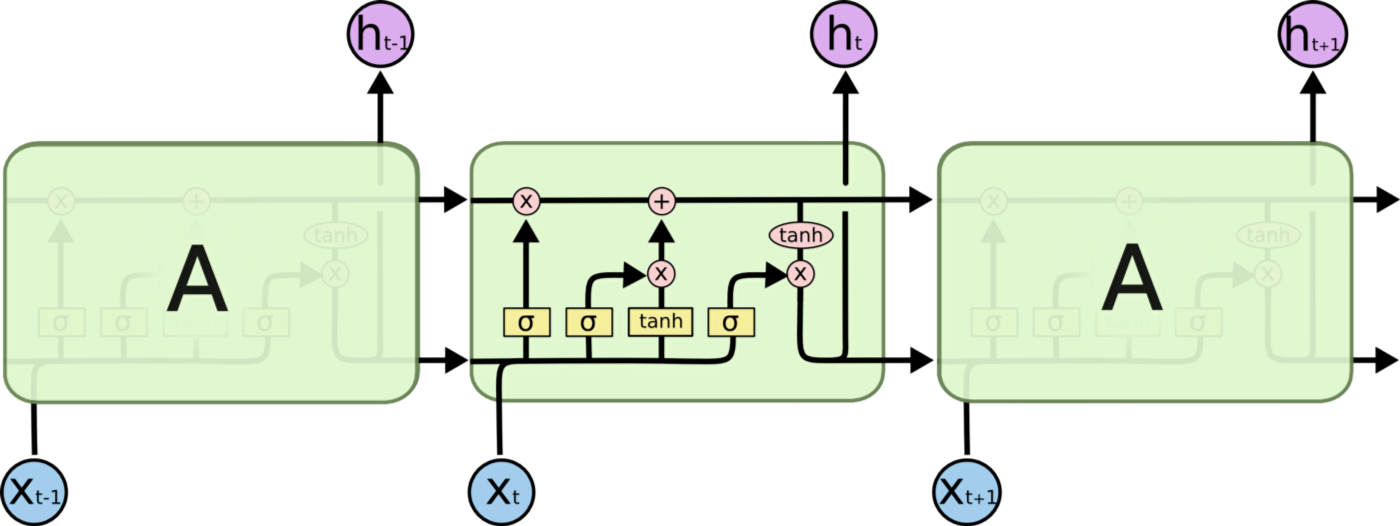


Figure: Unrolled RNN and LSTM chains: RNN with a single layer in each state

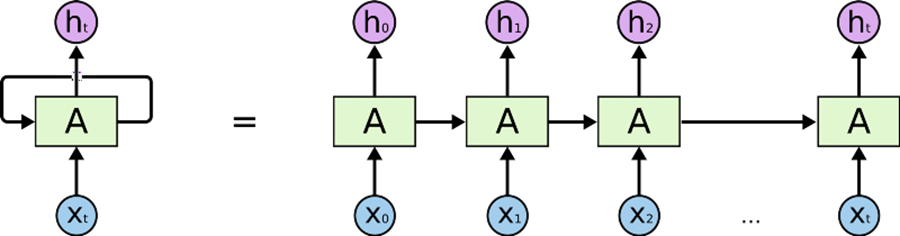


Figure: Unrolled RNN and LSTM chains: LSTM with one memory unit in each state

in Figure 3.11 and Figure 3.12. This gate, often called the ”forgotten gate”, generates a number between 0 and 1 for each value in the previous state. Therefore, if the door generates 0 for a particular value, that value is completely forgotten. The next stage of the cell decides what part of the new information (ht−1 and xt) will be retained in the state. First, a tan(h) layer is used to calculate an updated value for each new information bit. Subsequently, these updated values are filtered through another door to extract what is considered relevant by the LSTM. Finally, the filtered information is added with any information from the previous state that has not been forgotten to form the new cell state.

The last stage of the cell calculates the output for the current state (ht). The first step in this stage is to map the state of the cell in the acceptable output space. In Figure 3.11 and Figure 3.12, the activation function of tan(h) is applied to the state of the cell to push each value between -1 and 1. The transformed state then passes through the final sigmoid gate; therefore, the output consists of only the parts of the cell state that LSTM considers appropriate. For example, in a machine translation configuration, the LSTM can show if the subject is singular or plural, so the LSTM knows how to conjugate the next verb if in fact it is the next entry. There are other LSTM configurations, and each has its own advantages in certain configurations. However, a recent study shows that most of these variations do not work significantly better than vanilla LSTM. The LSTM model used in this thesis is a variant of the vanilla LSTM model that has a configurable number of memory units [39]. Note that the LSTM presented in this section had only one memory unit and, therefore, produced a single output. Adding additional memory units increases the dimensionality of the output in each state. Therefore, if the LSTM contains 50 memory units in each layer, the LSTM will eventually produce a 50-dimensional output vector after processing the entire input sequence. Since the challenge presented in this thesis is a binary classification task, the outputs of each LSTM are passed through another layer of sigmoid activation to produce a prediction.

**CHAPTER FOUR**

**IMPLEMENTATION**

**4.1 Introduction**

This chapter will cover detail explanations of configuration that is being used to make this project complete and working well. Section 4.2 gives an overview of master and Data Node environment. Section 4.3 describes necessary configuration for master node. Section 4.4 shows configuration for Data Nodes. Necessary configuration for apache spark is discussed in section 4.5. To ensure authorized access apache ranger is used and necessary configuration discussed in section 4.6. Section 4.7 describes configuration used to secure connectivity between nodes.

**4.2 DFS Cluster Environment**

Master and Data Nodes must have similar version of debian linux kernel and java or jre version for ideal hadoop’s distributed file system. For verification of debian linux and java version for master node and all Data Nodes, following commands can be executed as shown in the Fig. 4.1

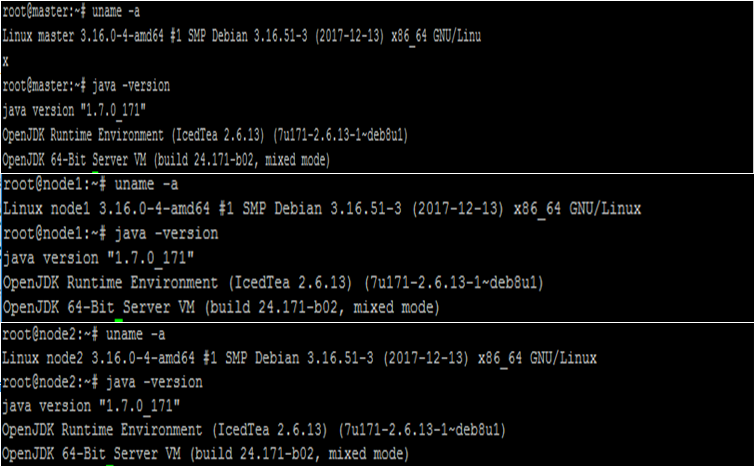


Fig. 4.1: Kernel and Java version of master and Data Nodes

**4.3 Configuring Master Node**

The master nodes in distributed Hadoop clusters host the various storage and processing management services. This section shows configuration for master node to make it fully functional.

**4.3.1 Host File Configuration**

For each node to communicate with its names, we have configured ‘/etc/hosts’ file at all three nodes to add the IP address of the three servers as Fig. 4.2

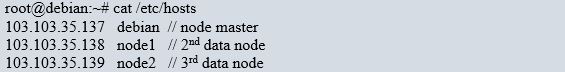


Fig. 4.2: Name and Data Node host file

**4.3.2 Key-Based SSH Authentication Setup**

The master node will use an ssh-connection to connect to other nodes with key-pair authentication, to manage the cluster without password. The key based password less SSH connectivity [22] among three nodes will be configured using below steps from master node as illustrated as below:

1. Login to **node-master** as the hadoop user, and generate an ssh-key.
2. ssh-keygen -b 4096.
3. Copy the key to the other nodes. Its good practice to also copy the key to the **node-master** itself, so that you can also use it as a Data Node if needed. Type the following commands, and enter the hadoop user’s password when asked. If you are prompted whether or not to add the key to known hosts, enter yes.
4. ssh-copy-id -i $HOME/.ssh/id\_rsa.pub debian@node-master.
5. ssh-copy-id -i $HOME/.ssh/id\_rsa.pub debian @node1.
6. ssh-copy-id -i $HOME/.ssh/id\_rsa.pub debian @node2.

**4.3.3 User Profile Configuration**

Add Hadoop binaries. Edit /home/hadoop/.bashrc file and add the following line

as Fig. 4.3:

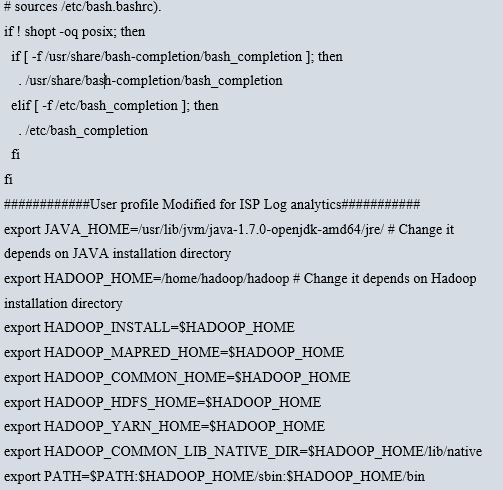


Fig. 4.3: User profile configuration

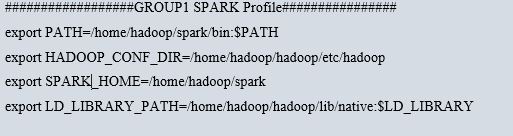
For spark running on hadoop deployment, additional configuration will be appended to user profile at **node-master as provided in Fig. 4.4**:

Fig. 4.4: Spark user profile

**4.3.4 Set Java Home**

We have installed Open JDK version 1.7 with default Java runtime environment. The java path must be defined for user profile as below:

‘/usr/lib/jvm/java-1.7.0-openjdk-amd64/jre/’

**4.3.5 ISP Log Analytics Core Site Configuration**

On master node update ~/hadoop/etc/hadoop/core-site.xml and set the Name Node location to **node-master** on port 9000 as Fig. 4.5:

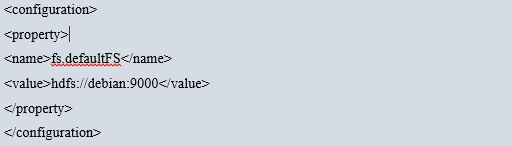


Fig. 4.5: Core site configuration

**4.3.6 Synchronization of Binary and Data Directory**

From master node ‘debian’ we shall execute below commands to synchronize node 1 and 2 hadoop binary installation directory:

# rsync -avzh /home/hadoop/ hadoop@node1:/home/hadoop/

# rsync -avzh /home/hadoop/ hadoop@node2:/home/hadoop/

**4.3.7 Configuration of Data Block Replication Parameter**

For configuration of default directory for Name and Data node to store meta data and user information, hdfs-site file will be edited. The file is located under etc. directory. After modification of the file, formatting the DFS file system is required. Edit ‘/home/hadoop/hadoop/etc/hadoop/hdfs-site.xml’ as Fig. 4.6 at master node to configure Name Node, Data Node and replication of data blocks among Data Nodes:

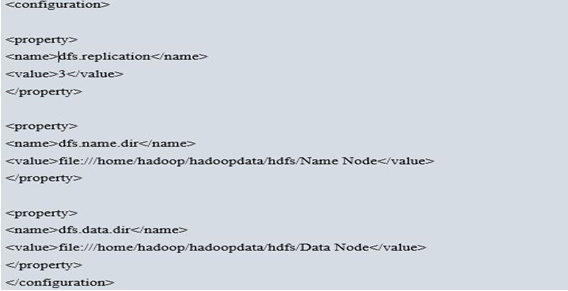


Fig. 4.6: HDFS site configuration

**4.3.8 Configuration of Yarn as Job Scheduler**

Under ~/hadoop/etc/hadoop/ directory, rename mapred-site.xml.template to mapred-site.xml. Edit the file as Fig. 4.7, setting yarn as the default framework for Map Reduce operations:

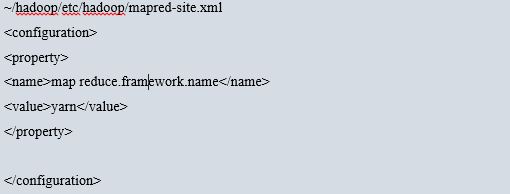


Fig. 4.7: Mapred-site configuration

Edit yarn-site.xml as illustrated in Fig. 4.8:

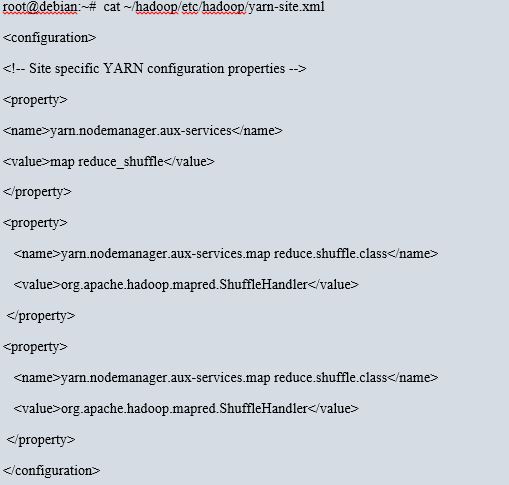


Fig. 4.8: Yarn site configuration

**4.4 Configuration of Data Nodes**

The file slave is used by startup scripts to start required daemons on all nodes. Edit ~/hadoop/etc/hadoop/slaves as Fig. 4.9:

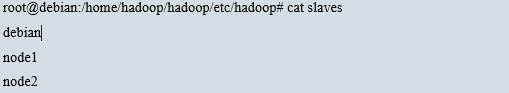
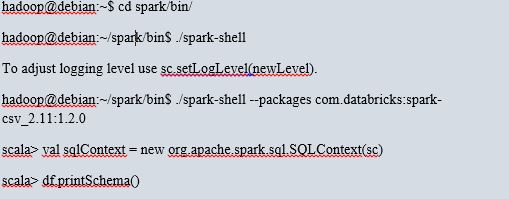


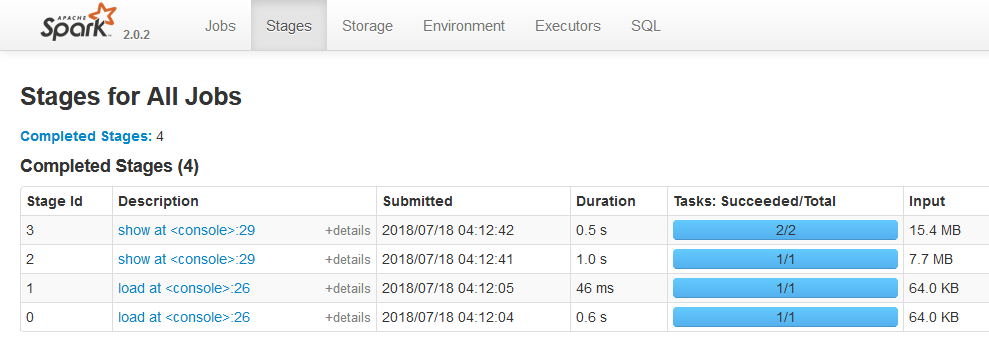
Fig. 4.9: Slave node configuration

**4.5 Apache Spark Configuration**

Apache Spark has been downloaded and extracted under hadoop user’s home directory using common hadoop user at Master node. Under the spark/bin directory run the ./spark-shell and load databricks for csv file processing as Fig. 4.10:

 Fig. 4.10: CSV databrick loading configuration

The resulting output of df.printSchema() will provide all column’s information for any log file in CSV format. Spark’s job execution, completed SQL query’s duration can be monitored from web gui using url http://103.103.35.137:4040 as Fig. 4.11:

Fig. 4.11: Apache Spark Web Gui

**4.6 Implementation of Access Control using Apache Ranger**

Apache Ranger has been installed and configured in third node of distributed file system and integrated with Log analytics. The Ranger administration portal can be accessed using url: 103.103.35.139:6080 as Fig. 4.12:

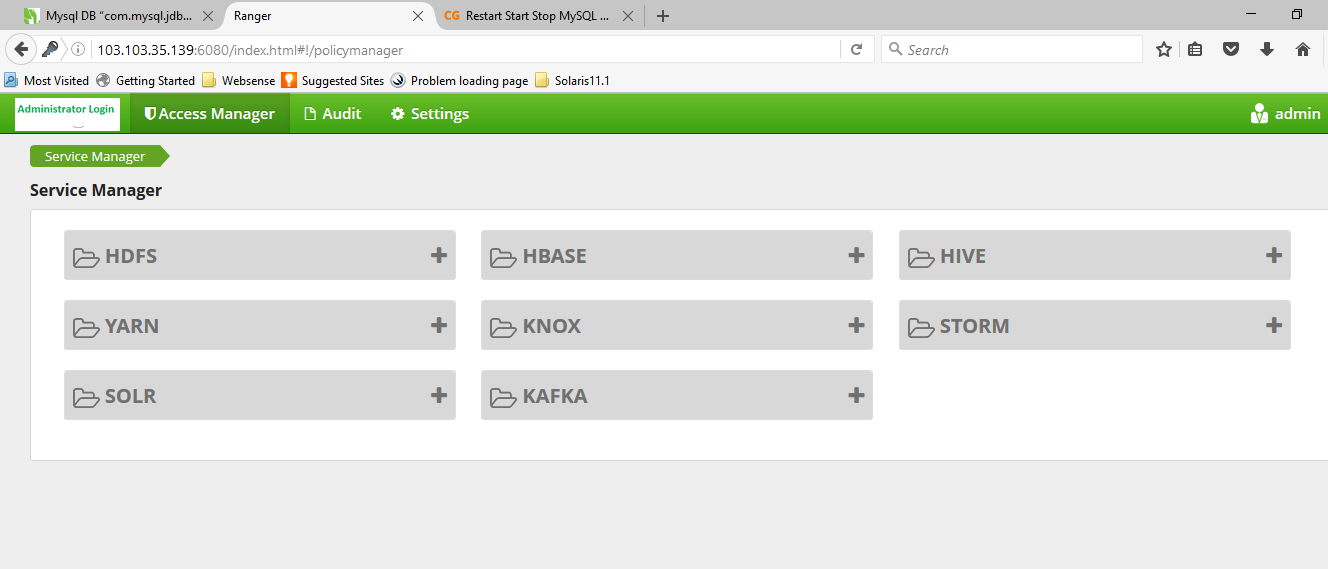


Fig. 4.12: Log analytics Access Control Policy’s Dash board

User named Shazzad and Mahfuz has been created with admin and user role as Fig. 4.13:

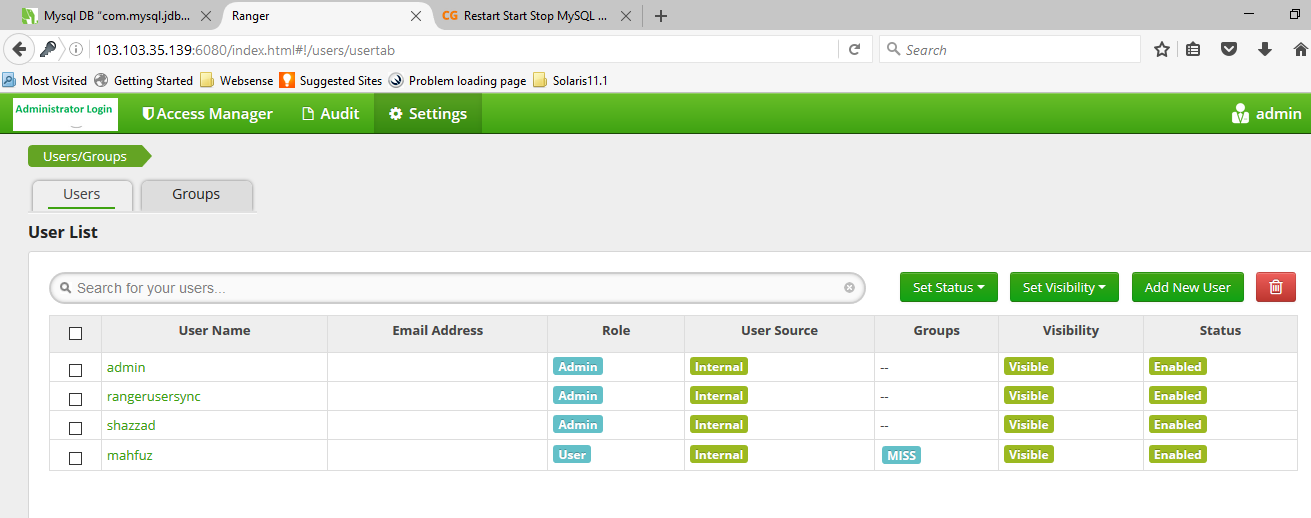


Fig. 4.13: Log analytics User and Group information

From above Fig. 4.13, user name, role and status can be verified and add, delete, modification of privileges can be done. Audit of administrator and user activity can be accessed from the mentioned web page.

**4.7 Secured Connectivity among Nodes**

In order to ensure secured communication among name node and all Data Nodes, we shall apply following best practices for configuration of VPN and IPTABLE [23] in participating hosts:

1. Enable Host Based Firewall.
2. Allowed Required Ports i.e. 50070, 8088, 6080 and Restrict or deny access to all other ports,

firewall-cmd --permanent --add-port=50070/tcp

firewall-cmd --permanent --add-port=8088/tcp

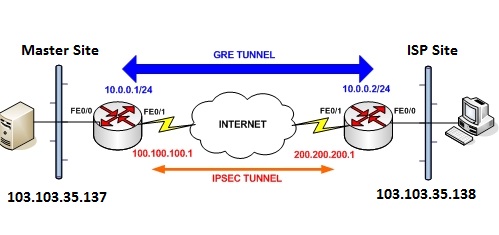
iptables -A INPUT -p tcp --dport 50070 ACCEPT

iptables -A INPUT -p tcp --dport 8088 ACCEPT

iptables -A INPUT -p tcp --dport 6080 ACCEPT

firewall-cmd --reload

1. Configure IPSec tunnel between Master and Slave Nodes as Fig. 4.14:

Fig. 4.14: IPSec over GRE Tunnel conFig.uration between Master and one ISP site

1. Define Access Control List (ACL) to allow only master and slave nodes.
2. Use Strong Cryptography key and Passphrase for tunnel configuration.
3. Allow Authorized user or administrator to modify or update network configuration and log activities.
4. Arrange for secondary or backup link connectivity between Name node and each Data Node.

**4.8 Setup and Configure NFS for benchmarking**

To setup NFS mounts for performance comparison with proposed DFS, 1 NFS server with 3 clients will be configured:

NFS Server: master node with ip address: 103.103.35.137

NFS Client: Data Node 1 with ip address: 103.103.35.137

NFS Client: Data Node 2 with ip address: 103.103.35.138

NFS Client: Data Node 3 with ip address: 103.103.35.139

For sharing a directory with NFS, an entry is required in “/etc/exports” configuration file.

A new directory named “isplog” in “/” partition will be created to share with client server using below commands in unix vi editor:

# vi /etc/exports

/isplog 103.103.35.137 (rw,sync,no\_root\_squash)

At the NFS client end i.e, from Data Node 2, check the nfsshare using below commands:

# showmount -e 103.103.35.137

Export list for 103.103.35.137:

/isplog 103.103.35.138

Above command shows that a directory named “isplog” is available at server having IP address “103.103.35.137” to share with Data Node 2.

To mount that shared NFS directory at the client’s node, following command will be used:

# mount -t nfs 103.103.35.137:/isplog /mnt/isplog

CIFS mount point will be configured in the same way for performance comparison in next chapter.

**CHAPTER FIVE**

**SIMULATION AND RESULT**

**5.1 Introduction**

In this chapter, the architecture and relevant configuration of the proposed DFS will be used to simulate and evaluate the performance of the file systems, query processing and security features. To evaluate the performance of log processing, a comparison between proposed query processing method and map reduce will be performed. ISP log will be used by Spark utility to perform search based on specific attribute and the latency of read, write operations shell be recorded in proposed DFS, NFS and CIFS shares respectively to draw performance comparison.

**5.2 Data Collection**

Our proposed solution will process user’s internet activity log real-time and for that purpose we have collected ISP user’s log from 2 ISP. User access log contains required information about Client’s name, contact address, phone no., Source IP Address, Destination IP Address, MAC Address, Source Port, Destination Port, Logging time. Sample of user’s activity log is provided as below Table 5.1:

Table 5.1: Sample ISP user internet activity log

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Date Time | Client IP | Src Port | Dst IP | Dst Port | Pop Name | Protocol |
| 00:59.0 | 54.255.166.250 | 80 | 10.254.27.206 | 39389 | Cyber-POP-104.30 | TCP |
| 00:59.0 | 10.254.31.76 | 51906 | 31.13.79.63 | 443 | Cyber-POP-104.30 | TCP |
| 00:59.0 | 10.254.27.19 | 42978 | 47.88.65.224 | 6798 | Cyber-POP-104.30 | TCP |
| 00:59.0 | 172.217.166.106 | 443 | 10.254.27.237 | 60589 | Cyber-POP-104.30 | TCP |
| 00:59.0 | 43.245.142.146 | 443 | 10.254.31.202 | 38215 | Cyber-POP-104.30 | TCP |
| 00:59.0 | 10.254.63.230 | 9101 | 123.134.94.228 | 9310 | Cyber-POP-104.30 | UDP |
| 00:59.0 | 31.13.79.36 | 443 | 10.254.27.54 | 53146 | Cyber-POP-104.30 | TCP |
| 00:59.0 | 10.254.63.230 | 9101 | 58.56.111.24 | 9310 | Cyber-POP-104.30 | UDP |
| 00:59.0 | 172.217.163.129 | 443 | 10.254.47.88 | 41857 | Cyber-POP-104.30 | TCP |

**5.3 Monitor HDFS Cluster**

Standard unix commands for administration can be executed to review health status locally. In addition, we can get useful information about run-time status of HDFS cluster with the HDFS dfsadmin command. Try for example:

#hdfs dfsadmin -report

The Name Node offers a summary of health and performance metrics through an easy-to-use web user interface. By default, the page is accessible via port 50070, so type the address in a web browser as http://103.103.35.137:50070 as like Fig. 5.1:

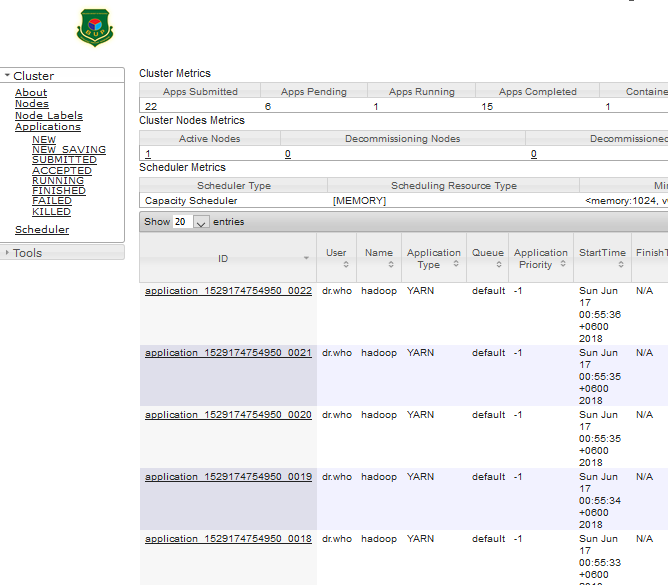


Fig. 5.1**:** HDFS Cluster Monitoring

**5.4 Experiments with Proposed Method**

From below query using Spark’s scala command shell, we can identify the client communicating with destination POP address as Table 5.2 where connected clients source, source port, destination IP address, destination port no. and POP id through which the clients connect to internet are displayed.

scala> df.select("ClientIP","Src\_Port","Dst\_IP","Dst\_Port","Pop\_Name").show()

Table 5.2: Scala output of clients connected with source, destination and pop name

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ClientIP** | **Src\_Port** | **Dst\_IP** | **Dst\_Port** | **Pop\_Name** |
| 54.255.166.250 | 80 | 10.254.27.206 | 39389 | Cyber-POP-104.30 |
| 10.254.31.76 | 51906 | 31.13.79.63 | 443 | Cyber-POP-104.31 |
| 10.254.27.19 | 42978 | 47.88.65.224 | 6798 | Cyber-POP-104.32 |
| 172.217.166.106 | 443 | 10.254.27.237 | 60589 | Cyber-POP-104.33 |
| 43.245.142.146 | 443 | 10.254.31.202 | 38215 | Cyber-POP-104.34 |
| 10.254.63.230 | 9101 | 123.134.94.228 | 9310 | Cyber-POP-104.35 |
| 31.13.79.36 | 443 | 10.254.27.54 | 53146 | Cyber-POP-104.36 |
| 10.254.63.230 | 9101 | 58.56.111.24 | 9310 | Cyber-POP-104.37 |
| 172.217.163.129 | 443 | 10.254.47.88 | 41857 | Cyber-POP-104.38 |
| 10.254.63.230 | 9101 | 65.254.40.44 | 9310 | Cyber-POP-104.39 |
| 10.254.63.230 | 9101 | 192.200.108.26 | 9310 | Cyber-POP-104.40 |
| 10.254.63.230 | 9101 | 27.221.103.4 | 9310 | Cyber-POP-104.41 |
| 10.254.31.223 | 53185 | 157.240.16.39 | 443 | Cyber-POP-104.42 |
| 52.222.236.16 | 443 | 10.254.27.229 | 59111 | Cyber-POP-104.43 |
| 10.254.27.110 | 34704 | 54.192.159.111 | 443 | Cyber-POP-104.44 |
| 10.254.63.230 | 9101 | 78.129.239.6 | 9110 | Cyber-POP-104.45 |
| 172.217.166.98 | 443 | 172.168.31.223 | 35210 | Cyber-POP-104.46 |
| 172.217.166.98 | 443 | 172.168.31.223 | 35210 | Cyber-POP-104.47 |
| 172.217.166.98 | 443 | 172.168.31.223 | 35210 | Cyber-POP-104.48 |
| 172.217.166.98 | 443 | 172.168.31.223 | 35210 | Cyber-POP-104.49 |
| 172.168.32.15 | 56247 | 103.217.105.211 | 443 | Cyber-POP-104.50 |
| 10.254.63.230 | 9101 | 27.221.103.11 | 9110 | Cyber-POP-104.51 |
| 10.254.31.223 | 53186 | 157.240.16.39 | 443 | Cyber-POP-104.52 |

From below query we can identify any specific client’s having MAC address: “84:16:f9:b0:3a:5b” with different ISP and Destination as Fig. 5.2:

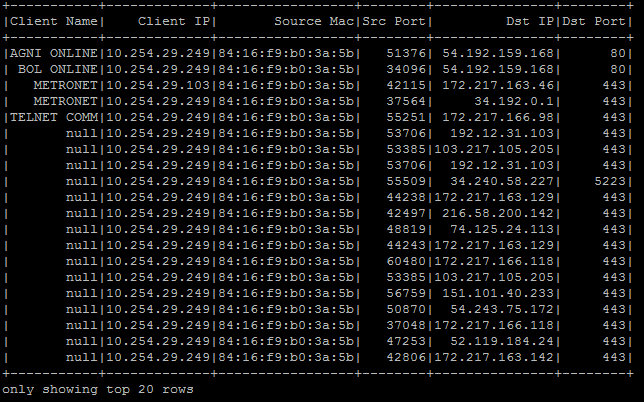


Fig. 5.2: Scala output of apache spark framework for MAC based query

**5.5 Comparison between Proposed Method and Map Reduce**

Proposed Spark can do processing in memory, while Hadoop MapReduce has to read from and write to a disk. As a result, the speed of processing differs significantly, Spark may be up to 100 times faster. However, the volume of data processed also differs.

Spark is used in proposed DFS architecture which performs faster in comparison to map reduce because it supports in-memory processing and this becomes very useful in iterative computing of big data. Spark comes with many cluster manager options such as Amazon EC2, Standalone, Mesos and Yarn. Spark’s performance over map reduce may be obtained with work count programs performance which was tested and result provided in Fig. 5.3:

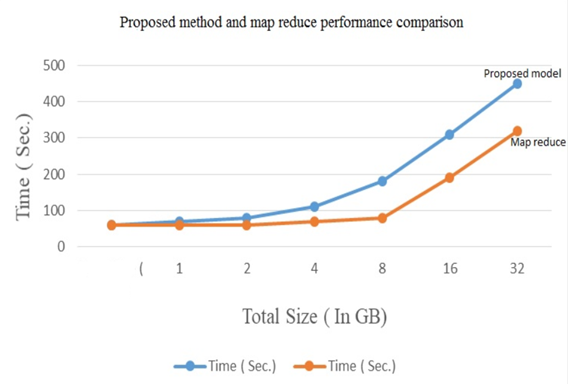


Fig. 5.3: Comparison between proposed method and map reduce (3 Instances)

From Fig. 5.3 the analysis used 10 Data Nodes and processed upto 48 GB data and comparison of required time by hadoop’s map reduce and spark are shown in the graph as blue (Hadoop) and red or orange (Spark).

**5.6 Comparison with Present Scenario of ISP Log**

Currently government bodies and law enforcement agencies tracing a user’s internet access activity through a distributed process. This is very time consuming and success rate is very low. ISP’s are not able to maintain user internet access log with details and there is no backup or redundant copy of user log. The log is not preserved for long duration. They may modify the log and hide information. In our proposed system, logs will sync real-time. We have shown a comparison below.

In current scenario law enforcement agencies query to ISP for logs, System administrator may manipulate the log and hide information. Also there can be some human error in provided data. In our proposed solution, government agencies can perform their query directly into the system; there will be no dependency to other people which will ensure the accuracy of the data.

In current process data is requested through mail or in written paper which may expose those data to unauthorized people. In our proposed system all the ISP’s are connected to central Name Node through a secured network channel and law enforcement agencies are performing their query centrally. They don’t need to share the data in mail or paper. This will ensure the integrity of the information.

All the ISP’s will be connected through a secured channel. Data will be authentic. ISP’s and users cannot deny the information in the log.

In current process it may take 15/20 days to collect information for all the ISP’s. In our proposed solution log can be retrieved in a secured manner in less than one second.

**5.7 Monitor Access to ISP Log Analytics Using Proposed Solution**

In order to impose restrictions on logging into ISP log server’s master node, we have implemented access control using Apache Ranger where user and group are defined with specific privileges. We can monitor user session from ranger’s audit menu as Fig. 5.4:

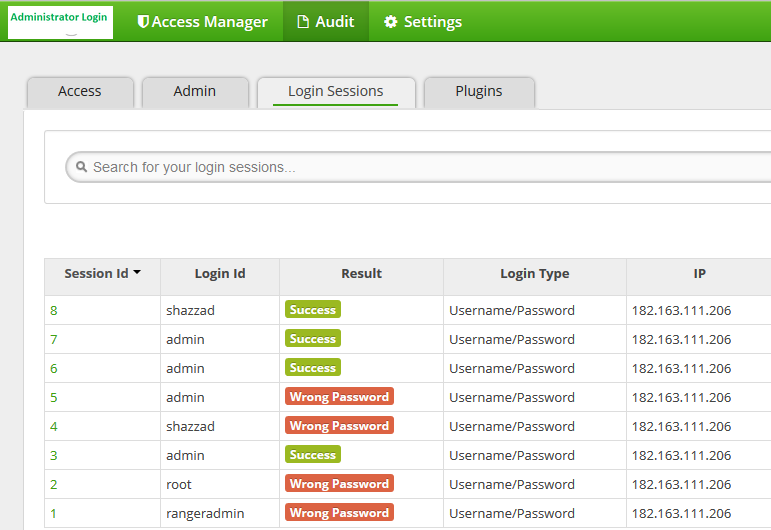


Fig. 5.4: User activity monitor

**5.8 Performance benchmarking of Proposed DFS over NFS and CIFS**

Fig. 5.5 shows that using proposed DFS provides much better read latency over NFS and CIFS. For 100 GB data read operation, latency of proposed DFS is lower than that of CIFS and NFS.

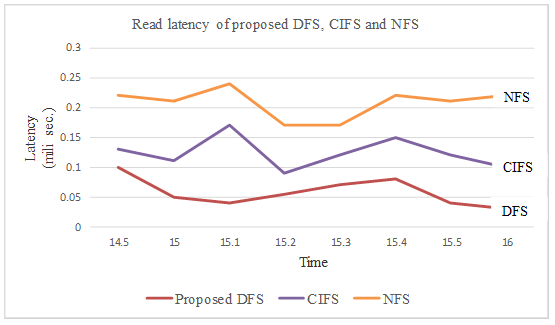


Fig. 5.5: Read latency of proposed DFS, CIFS and NFS

Fig. 5.6 shows that using proposed DFS provides much better write latency over NFS and CIFS. For 100 GB data write operation, latency of proposed DFS is lower than that of CIFS and NFS.

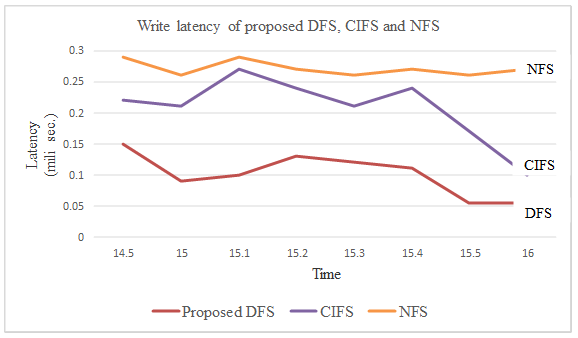


Fig. 5.6: Write latency of proposed DFS, CIFS and NFS

**5.9 Security Comparison of Proposed DFS over NFS, CIFS**

Security comparison among NFS, CIFS and proposed DFS can be described as below Table 5.3:

Table 5.3: Security comparison among NFS, CIFS and proposed DFS

|  |  |  |  |
| --- | --- | --- | --- |
| File System | NFS | CIFS | Proposed DFS |
| Architecture | File-based protocol | File-based protocol | Cluster based |
| Processes | Stateless | Stateful | Stateful |
| Naming | No central metadata server | No central metadata server | Central metadata server |
| Synchronization | No synchronization | No synchronization | Write-one-read-many, give locks on objects to clients, using leases |
| Session Requirement | No | Yes | Yes |
| Communication speed | Fast | Slow | Fast |
| Fault Tolerance | No | No | Yes |
| Security mechanism | No | Kerberos | Security in the form of AAA |

From Table 5.3 it is found that the architecture of NFS and CIFS is file based protocol but proposed DFS is cluster based. NFS has stateless process, CIFS and proposed DFS have stateful process. There is no central metadata server for NFS and CIFS except proposed DFS. There is no synchronization in NFS and CIFS. Communication speed is fast in both NFS and proposed DFS. There is no security mechanism in NFS, kerberos in CIFS and AAA security in proposed DFS.

**CHAPTER SIX**

**CONCLUSIONS AND FUTURE WORKS**

**6.1 Conclusion**

In Perspective of Bangladesh we have designed a centralized log analytics system to access and monitor user’s activity in internet using commodity hardware and open source software for cost optimization and rapid deployment. Since the generated log by all ISP exceeds regular computing storage and processing power, the best big data solution using Distributed file system has been used for the project. Throughout the chapter 2 and 3, we have collected present status of ISP user log preservation and retention policy in our country and drawn comparison with neighborhood and European Union. In chapter 4, we implemented ISP log analytics using Apache hadoop with spark and ranger. Network security required for communication among master and all nodes has been discussed. A sample script to merge all ISP log into a single csv file has been included which will be rotated according to log retention policy. The csv file has been used in experiments for specific purpose. Finally, all experiment results regarding log analytics distributed file system implementation, map reduce algorithm, no-sql query and access control features have been included in chapter 5. The log analytics system offers flexibility and ability to process large datasets. Our proposed solution will ensure authorized access and protect user’s privacy.

**6.2 Major Contribution and Recommendation for Future Works**

In future Govt. regulatory agency may benefit from our system for conducting investigation on cybercrime or generate statistics relating to internet usage. Our proposed solution will also preserve user activity log for a long duration to detect and investigate suspicious activity that occurred a long time ago. Through implementation of one master node and three slave nodes we have simulated our design and performed search for any user’s internet activity. We have also implemented fundamental security features and proposed optimized usage of resources. Where volume, velocity and verity of data set is a challenge for processing and beyond normal processing capacity, our proposed architecture can be implemented with comprehensive storage and retention period. In future following development can be performed on ISP log analytics DFS:

1. Software defined perimeter (SDP) might be used here for securing communication channel between master node and Data Node.
2. Two-factor authentication might be used for administrative users.
3. Our proposed platform could be migrated into Docker platform for maximum utilization in current hardware setup.
4. Migration of the proposed solution to cloud infrastructure.

**REFERENCES**

|  |  |
| --- | --- |
| [1] | Dang Van Thin∗, Vu Duc Nguyen, Kiet Van Nguyen, Ngan Luu Thuy Nguyen, *A Transformation Method for Aspect-based Sentiment Analysis*. Journal of Computer Science and Cybermetrics, V.34, N.4, 2018. |
| [2] | Shashank Gupta, *Sentiment Analysis: Concept, Analysis and Applications*. <https://towardsdatascience.com/>, January, 2018. |
| [3] | Md. Mahfuzur Rahman, Sheikh Shah Mohammad Motiur Rahman, Shaikh Muhammad Allayear, Md. Fazlul Karim Patwary and Md Tashir Ahmed Munna, *A Sentiment Analysis Based Approach for Understanding the User Satisfaction on Android Application*. Springer, Singapore, January, 2020. |
| [4] | Priyanka Tyagi and Dr. R.C. Tripathi, *A Review towards the Sentiment Analysis Technique for the Analysis of Twitter Data*, In 2nd International Conference on Advanced Computing and Software Engineering ICASE-2019. |
| [5] | User reviews of amazon’s electronic products, <https://www.kaggle.com/datafiniti/consumer-reviews-of-amazon-products> , 2018. |
| [6] | R. Arora and S. Suman, “*Comparative analysis of classification algorithms on different datasets using WEKA,*” International Journal of Computer Applications, vol. 54, no. 13, pp. 21–25, 2012. |
| [7] | J. Xia, F. Xie, Y. Zhang, and C. Caulfield, “*Artificial intelligence and data mining: algorithms and applications,*” Abstract and Applied Analysis, vol. 2013, Article ID 524720, 2 pages, 2013. |
| [8] | M.Rambocas, and J. Gama, “*Marketing Research: The Role of Sentiment Analysis*”. The 5th SNA-KDD Workshop’11. U nivers ityof Porto, 2013 |
| [9] | Harpreet Kaur, Veenu Mangat, Nidhi, *A survey of sentiment analysis techniques*, International Conference on I-SMAC 2017. |
| [10] | Navathe, S.B., Ramez, *E: Data warehousing and data mining. Fundam. Database Syst*., 841–872 (2000) |
| [11] | Gupta, V., Lehal, G.S, *A survey of text mining techniques and applications*. J. Emerg. Technol. Web Intell. 1(1), 60–76 (2009). |
| [12] | Yogapreethi.N, Maheswari.S, *A Review on Text Mining in Data Mining*, International Journal on Soft Computing (IJSC) Vol.7, N0. 2/3, August 2016. |
| [13] | Francisco Moraes Oliveira-Neto, Lee D. Han, And Myong Kee Jeong.” *An Online Self-Learning Algorithm for License Plate Matching*”, IEEE Transactions On Intelligent Transportation Systems, Vol. 14, No. 4, December 2013. |
| [14] | Yuefeng Li, Abdulmohsen Algarni, Mubarak Albathan, Yan Shen, And Moch Arif Bijaksana,”*Relevance Feature Discovery For Text Mining*”, IEEE Transactions On Knowledge And Data Engineering, Vol. 27, No. 6, June 2015. |
| [15] | Vandana Shrivastava, *A Methodical Study of Web Crawler,* VandanaShrivastava Journal of Engineering Research and Application ISSN : 2248-9622 Vol. 8, Issue 11 (Part -I) Nov 2018, pp 01-08. |
| [16] | Jian Ma, Wei Xu, Yong-Hong Sun, Efraim Turban, Shouyang Wang, And Ou Liu,” *An OntologyBased Text-Mining Method To Cluster Proposals For Research Project Selection*”, IEEE Transactions On Systems, Man, And Cybernetics—Part A: Systems And Humans, Vol. 42, No. 3, May 2012. |
| [17] | Chien-Liang Liu, Wen-Hoar Hsaio, Chia-Hoang Lee, Gen-Chi Lu, And Emery Jou,” *Movie Rating And Review Summarization in Mobile Environment*”, IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, Vol. 42, No. 3, May 2012. |
| [18] | Yue Hu And Xiaojun Wan,” *Ppsgen: Learning-Based Presentation Slides Generation For Academic Papers*”, IEEE Transactions On Knowledge And Data Engineering, Vol. 27, No. 4, April 2015. |
| [19] | Xiuzhen Zhang, Lishan Cui, And Yan Wang, “*Commtrust: Computing Multi-Dimensional Trust By Mining E-Commerce Feedback Comments*”, IEEE Transactions On Knowledge And Data Engineering, Vol. 26, No. 7, July 2014. |
| [20] | Dnyanesh G. Rajpathak And Satnam Singh,” *An Ontology-Based Text Mining Method To Develop D-Matrix From Unstructured Text*”, IEEE Transactions On Systems, Man, And Cybernetics: Systems, Vol. 44, No. 7, July 2014. |
| [21] | Jehoshua Eliashberg, Sam K. Hui, And Z. John Zhang,” *Assessing Box Office Performance Using Movie Scripts: A Kernel-Based Approach*”, IEEE Transactions On Knowledge And Data Engineering, Vol. 26, No. 11, November 2014. |
| [22] | Donald E. Brown,” *Text Mining The Contributors To Rail Accidents*”, IEEE Transactions On Intelligent Transportation Systems, Vol. 17, No. 2, February 2016. |
| [23] | Luís Filipe Da Cruz Nassif And Eduardo Raul Hruschka,” *Document Clustering For Forensic Analysis: An Approach For Improving Computer Inspection*” IEEE Transactions On Information Forensics And Security, Vol. 8, No. 1, January 2013. |
| [24] | Charu C. Aggarwal, Yuchen Zhao, And Philip S. Yu, “*On The Use Of Side Information For Mining Text Data*”, IEEE Transactions On Knowledge And Data Engineering, Vol. 26, No. 6, June 2014. |
| [25] | Luís Filipe Da Cruz Nassif And Eduardo Raul Hruschka,” *Document Clustering For Forensic Analysis: An Approach For Improving Computer Inspection*” IEEE Transactions On Information Forensics And Security, Vol. 8, No. 1, January 2013 |
| [26] | W. Y. Wang, “” liar, liar pants on fire”: A new benchmark dataset for fake news detection,” arXiv preprint arXiv:1705.00648, 2017. |
| [27] | N. J. Conroy, V. L. Rubin, and Y. Chen, “*Automatic deception detection: Meth- ods for finding fake news,*” in Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community, p. 82, American Society for Information Science, 2015. |
| [28] | E. Tacchini, G. Ballarin, M. L. Della Vedova, S. Moret, and L. de Alfaro, “*Some like it hoax: Automated fake news detection in social networks*,” arXiv preprint arXiv:1704.07506, 2017. |
| [29] | VDM Verlag, “*Designing New Crawling and Indexing Techniques for Web Search Engines*”, Q. Tan (2009). |
| [30] | C. Olston&M. Najork “*Web Crawling*”, Now Publishers Inc. (2010). |
| [31] | Addison Wesley, “*An Introduction to Search Engine and Web Navigation*”, M. Levene (2005). |
| [32] | C. D. Manning, P. Raghavan, and H. Schu¨tze, “*Support vector machines and machine learning on documents,*” Introduction to Information Retrieval, pp. 319– 348, 2008 |
| [33] | N. M. Nasrabadi, “*Pattern recognition and machine learning,*”, Journal of elec- tronic imaging, vol. 16, no. 4, p. 049901, 2007. |