Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Information Technology
Department of Software

DATA ANALYSIS USING MINHASH AND MACHINE LEARNING IN APACHE SPARK FRAMEWORK

A Dissertation
Submitted to the Council of the College of Information Technology for Postgraduate Studies of University of Babylon in Partial Fulfilment of the Requirements for the Degree of Doctorate of Philosophy in Information Technology - Software

BY
Wafaa Shaker Mohammed Hassan

Supervised By
Prof. Dr. Rafah Mohammed Kadhem
Assist. Prof. Dr. Mehdi Ebady Manaa Mehdi

2021 A.D **1443 A.H**

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

أَنزَلَ مِنَ ٱلسَّمَآءِ مَآءً فَسَالَتْ أَوْدِيَةٌۢ بِقَدَرِهَا فَٱحْتَمَلَ ٱلسَّيْلُ زَبَدًا
رَّابِيًا ۚ وَمِمَّا يُوقِدُونَ عَلَيْهِ فِى ٱلنَّارِ ٱبْتِغَآءَ حِلْيَةٍ أَوْ مَتَٰعٍ زَبَدٌ مِّثْلُهُۥ ۚ كَذَٰلِكَ
يَضْرِبُ ٱللَّهُ ٱلْحَقَّ وَٱلْبَٰطِلَ ۚ فَأَمَّا ٱلزَّبَدُ فَيَذْهَبُ جُفَآءً ۖ وَأَمَّا مَا يَنفَعُ
ٱلنَّاسَ فَيَمْكُثُ فِى ٱلْأَرْضِ ۚ كَذَٰلِكَ يَضْرِبُ ٱللَّهُ ٱلْأَمْثَالَ ﴿١٧﴾

صَدَقَ اللَّهُ الْعَلِيُّ الْعَظِيمُ

سُورَةُ الرَّعْدِ

Dedication

To The Messenger of Mercy and Humanity

To his cousin, Imam Ali Ibn Abi Talib (AS)

To his daughter, Fatima Al Zahraa (AS)

To his grandsons, Imam Al Hassan and

Imam Al Hussein (A.S)

All Loved by Allah

Allah’s Blessings be upon him and his household

Acknowledgement

First and foremost, the greatest thanks to Allah for his gracious and merciful help that the writer was finally able to complete this dissertation. My heartily thanks to my supervisors, Dr. Rafah Mohammed Kadhem and Dr. Mehdi Ebady Manaa, whose encouragement, supervision, and support from the preliminary to the concluding level enabled me to understand the subject with many of their valuable hours.

I would also like grant my big thanks to all lecturers and staff members, at IT - College / University of Babylon, who were kind enough to give me their precious time and assistance.

Many thanks to my dear parents for providing me with support that I needed to get through the PhD. Thanks to my sisters and brothers for giving me a push to get started.

Last, but certainly not least, I offer my regards and blessings to all of those who supported me in any respect during the completion of the study.

Abstract

Social media plays an integral role in generating a large volume of structured, semi-structured and unstructured forms of data. It is considered a powerful marketing tool and a business insight. The major challenge facing these data involves the need to build a robust model using efficient techniques to extract new knowledge and make the right decision in less time, especially in medical data. Sentiment analysis is one of the scale up data analysis techniques used to extract public opinions for large social data which usually encompass short sentences that are, generally speaking, not constructed with proper grammatical rules. This study aims to perform sentiment classification on tweets in an efficient and timely manner using a hybrid model of Min-hash and machine learning to obtain highly accurate results, and to parallel execution using the Apache Spark programming framework. For conducting sentiment analysis, we used two datasets, a COVID-19 tweets Provided by IEEE data port and COVID-19 Vaccine Stance datasets which include three fields – ID of the tweet, Sentiment of the tweet, and the Tweet itself.

The proposed sentiment analysis system consists of the following four steps: data collection, data cleaning and pre-processing, Minhash and sentiment classification. The system enables high-level performance of sentiment classification while taking advantage of combining Minhash with learning-based classifier. The performance of Minhash with Locality Sensitive Hashing (LSH) is compared to that of Minhash with each of Logistic Regression (LR), Random Forest (RF), Naive Bayes (NB), and Support Vector Machine (SVM) in a parallel and a distributed manner. Performance parameters such as user, system and real time, time consumed, and accuracy obtained from confusion matrix are used to classify tweets into positive, negative, and neutral. The obtained results have been applied in the comparative analysis to analyse the behaviour of the classifiers in the Amazon Web Services (AWS) spark Cluster,

the Local Spark cluster and in the conventional system. Moreover, this system can analyse data by decreasing the processing time through adding more nodes in the cluster. Experimental results show that LR and RF outperform SVM and NB classifiers.

Table of Contents

List of Tables

List of Figures

List of Appendices

List of Abbreviations

Abbreviation	Description
APIs	Application Programming Interface
AWS	Amazon Web Services
ANN	Artificial Neural Network
COVID-19	Coronavirus disease
CoV	Coronaviruses
CRC32	Cyclic Redundancy Check 32
CSV	Comma-Separated Values
DT	Decision Tree
DAG	Directed Acyclic Graph
ETC	Ethereum Classic
EC2	Amazon Elastic Compute Cloud
EBS	Elastic Block Store
FP	False Positive
FN	False Negative
GiB	Gibibyte is base 2
HDFS	Hadoop Distributed File System
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IEEE	Institute of Electrical and Electronics Engineers
IaaS	Infrastructure as a Service
IMs	International Media Support
ID	Identity Document
JSON	JavaScript Object Notation

JVM	Java Virtual Machine
KDD	Knowledge Discovers in Databases
LSH	Locality-Sensitive Hashing
LR	Logistic Regression
LSTM	Long short-term memory
ML	Machine Learning
Mllib	Machine Learning library
NB	Naive Bayes
NLP	Natural Language Processing
NCov	novel coronavirus
ncov2019	Novel Coronavirus
OAuth	Open Authorization
PaaS	Platform as a Service
REST	Representational State Transfer
RT-FAV	Retweet Favourite
RDD	Resilient Distributed Dataset
RF	Random Forest
RESTful	Representational state transfer
SA	Sentiment Analysis
SVC	Support Vector Classifier
SVM	Support Vector Machine
SSL	Secure Sockets Layer
SDK	Software Development Kit
SOAP	Simple Object Access Protocol
S3	Simple Storage Service
SaaS	Software as a Service

SQL	Structured Query Language
sarscov2	severe acute respiratory syndrome coronavirus 2
TF-IDF	Frequency-Inverse Document Frequency
TP	True Positive
TN	True Negative
URL	Uniform Resource Locator
VPC	Virtual Private Cloud
VPN	Virtual Private Network
vCore	Voltage Core
web UI	web User Interface
XML	Extensible Markup Language
XGBoost	eXtreme Gradient Boosting
YARN	Yet Another Resource Negotiator

CHAPTER ONE

GENERAL INTRODUCTION

1.1 **Overview**

The data types of healthcare system are divided into three categories: structured data (data that adheres to a set of data types, formats, and structures), semi structured (data having a simple structure and self-descriptive nature) or unstructured format (data that is devoid of any intrinsic structure).

Different types of frameworks required to run different types of analytics. A variety of workloads present in large-scale data processing enterprise. To achieve a business goal, a combination of said workloads can be deployed:

- Batch-oriented processing, for example, MapReduce based frameworks like Hadoop, for recurring tasks such as large-scale data mining or aggregation [1].
- OLTP, such as user-facing e-commerce transactions, with Apache HBase [2]
- Stream processing, to handle stream sources such as social media feeds or sensor data, with Storm being a representative framework [3].
- Interactive ad-hoc query and analysis with Apache Drill [4].

In a local system, any such process requires a single core of the CPU. A framework that makes use of the current CPU in the local system in a distributed context is required. Hadoop is one of the most well-known solutions for accomplishing this goal.

Hadoop is open-source software that designed for analytics purposes. It allows distributed processing of big data across multiple machines with a degree of fault tolerance [5]. Hadoop architecture is known as MapReduce framework. It divides, processes the data, and run it in parallel. However, there are some defects in Hadoop. MapReduce has a high overhead while running a job. Therefore, Hadoop relatively ill-suited while is used for cases of an iterative or low-latency environment [4]. So, there is a need to develop the framework and solve some issues in Hadoop. One of the developed software is Apache Spark that is built on top of Hadoop architecture. Apache Spark was created to work in tandem with Hadoop to alleviate some of the latter's shortcomings.

Apache Spark is an open-source distributed computing framework. It is designed to perform the low latency jobs, and store in-between data and results in memory. Therefore, it is called Memory Computing that improves the efficiency of data computing. As a result, Spark is better than Hadoop for data mining and machine learning applications [4]. Moreover, to perform scalable machine learning, graph analysis, streaming and structured data processing. There are upper-level libraries in it. Spark is a cluster computing framework with Scala, Java, Python and R language-integrated APIs [6]. In addition, it extends the MapReduce model to efficiently use more types of computations which includes Interactive Queries and Stream Processing [7]. The superior feature of Apache Spark is Resilient Distributed Dataset (RDD) which offers fault tolerant objects that can be deployed across a distributed cluster [1]. The advantage of Spark in this case is more about the simplicity of converting the application to a clustered implementation.

Given the tremendous development of data in recent years, especially in the field of healthcare, it became necessary to use a technique that processes, stores and handles such huge data in secure, online system, which led to the introduction of cloud computing. Cloud computing is a technology that connects a group of computers together to provide several services to the user at any time. Cloud computing provides faster data access time, high processing speed, and lower cost [3]. Therefore, the healthcare cloud provides better performance, lower cost, and unlimited scalability [8].

Analysis of data is a challenging task as it involves large, distributed file systems. The infrastructure requires for analyzing data is different from Amazon analysis technology and data mining on various types of data. Mapreduce is widely popular for analysis of data. Mapreduce is working with mapping, sorting, shuffling, and reducing using Master/Slave architecture. Similarly, Amazon MapReduce programming model over large data set is

introduced by Amazon, on the web especially used for healthcare or ecommerce. In this dissertation, Amazon EC2 cloud computing model has been used for central part of designed web and for collection and storing of large data Amazon uses S3. Amazon clusters is a group of servers which is working together to perform any type of tasks on distributed database on different servers in parallel. Amazon services are used in analysis of data and to increase business efficiency.

1.2 Problem Statement

The work's main concern revolves upon the following flaws found in tweets and healthcare data which are:

1. Data collection of social networks is the most crucial step due to data privacy. To get the authorization to view and print the tweets' text, the authorization of the website must be got. In this dissertation, the website is the twitter.
2. The designing of a framework needs to be optimized so that it solves the sentiment analysis of people's opinions in an online system. This can be obtained by using AWS spark cluster.

1.3 Research Questions

In this work, many crucial questions are raised to be answered by analyzing, modeling, and implementing a distributed system to obtain a good result to analyze the coronavirus tweets using data mining classification analysis.

The main addressed questions by this study are: -

a. What is the effect of Minhash on the Randomization?
b. What is the effect of Minhash on the NB?
c. What is the effect of Minhash on the LR?
d. Is data mining with machine learning enhanced the accuracy?
e. Is implementation of machine learning in apache spark reduced time consumed?

1.4 Research Aims

The main aim of data analysis is to help organization make better decisions by allowing scientists and other data users to analyze large volumes of transactional data. As well as other data sources that may have been left untapped by the Conventional system.

The research objectives involve the following five major tasks:

a. Propose a healthcare system based on machine learning and Data processing. The proposed architecture provides a potential platform for handling and classifying data from Twitter.

b. Suggesting a good classification analysis that can be used to sentiment analysis using Minhash with large data scale such as Spark Machine Learning algorithms. In addition, it should be efficient to be implemented in a short period of time.

c. Using AWS, which offers cost-effective, scalable, and secure compute, storage, and database capabilities, helps reduce time to research.

d. Designing and implementing data analysis system using performance evaluation such as accuracy and time.

1.5 Research Impact

This work falls within the scope of application for the corona virus tweets, and to classify Sentiment in less time. It is mandatory to build such classification model using machine learning techniques for large health organizations because of increasing rate of COVID-19 disease cases.

1.6 Research Contribution

The main contributions of this dissertation are: -

a. A proactive Model is implemented to satisfy a good results using MinHash-ML which is an optimized alternative to system MinHash-LSH to build a Sentiment classification.

b. Time consumed is reduced using Apache Spark Cloud Platform.

1.7 **Related Works**

The most related works associated with sentiment classification are explained in the following paragraphs.

In any Machine learning task, cleaning or preprocessing the data is as important as model building. Text data is one of the most unstructured forms of available data and when comes to deal with Human language then it is complex. NLP is a technology that works behind it where before any response lots of text preprocessing takes place. NLP techniques have been used to monitor public opinion around the world regarding mask-wearing during the pandemic [9]. The sane NLP techniques have been used in this dissertation.

The difference between the reactions towards the pandemic in different cultures has been studied by Imran et al. [10] through sentiment and emotion analysis, implemented with deep learning classifiers. Besides the correlation between tweets' polarity from different countries, the authors also state that NLP can be used to link the emotions expressed on social platforms to the actual events during the coronavirus pandemic. A long short-term memory (LSTM) model was employed by Imran et al. [10] to access the sentiment polarity of people from different cultures to the coronavirus using COVID-19-related tweets dataset, called Sentiment140. Their results show a higher correlation between the USA and Canada (0.96 for positive and 0.97 for negative sentiments) and between India and Pakistan (0.81 for positive and 0.86 for negative sentiments). However, a low correlation between Norway and Sweden (0.50 for negative and 0.40 for positive sentiments) exists. In this dissertation, the NLP in preprocessing step is used on the tweets datasets called Coronavirus (COVID-19) Tweets Dataset and COVID-19 Vaccine Stance Dataset.

Researchers have had a great need to establish effective analytical approaches to comprehend the flow of information and improve human perceptions under epidemic situations since the rapid spread of Covid-19 (Coronavirus) infection.

Chakraborty, K. et al. [11] claimed that people mostly tweet positive sentiments related to COVID-19, but they can also re-tweet negative feelings. They used fuzzy inference with VADER sentiment lexicon to label the data into three classes: positive, negative, and neutral. They employed hyper parametric machine learning classifiers (Naïve Bayes, AdaBoost, and Logistic Regression) to analyze 226,668 English tweets related to COVID-19. The data sets which are used in this dissertation also have three classes positive, negative, and neutral. Moreover, researchers have used Naïve Bayes, Logistic Regression with two more (SVM, RF) machine learning classifiers after applying data mining (Minhash)

There have been some works on the use of big data platforms such as apache spark in Twitter data analysis in various application domains [12]. Supervised approaches are mainly used for specified event type detection and the task is mostly framed as a text classification [12]. In this dissertation, the apache spark has been used in twitter data.

Samuel et al. [13] trained two classification models (naïve Bayes and Logistic regression) and compared their performance on classification of COVID-19 tweets into a positive or a negative class. They found that there was a growth of negative sentiment in COVID-19 tweets.

One of the works in the field of sentiment analysis is presented in [13]. Authors conducted their researches in the domain of text analytics specifically on Twitter data on Covid-19 pandemic [13]. To the best of our knowledge, works regarding the public sentiment on Covid-19 vaccine is still less explored.

Four classification algorithms are considered in this dissertation: Naïve Bayes Classifier, Logistic Regression, Random Forest and Support Vector Machine.

Recently, sentiment analysis of social media data in the context of COVID-19 pandemic has attracted attention from the research community. Several studies have utilized social media such as Twitter content as the source of input data to

analyze the impact of the ongoing COVID-19 pandemic on public sentiments. For instance, Barkur et al. have analyzed the impact of lockdown in India due to COVID-19 using sentiment analysis and text mining techniques [14] and their observation showed that the majority of views about lockdown were negative but also there were some positive opinions.

The users' dynamics of the opinions regarding the COVID-19 vaccination in the month following the first vaccine announcement has been analyzed by Cotfas et al. [15]. The authors have considered messages on Twitter regarding the COVID-19 vaccination and have concluded that most of the tweets have had a neutral stance, followed by in favor and against stances [15]. Researchers in this dissertation have used the same dataset (COVID-19 Vaccine Stance Data set).

There are also a small number of studies that focused more on the comparison of different algorithms for sentiment classification of COVID-19 tweets than on their application and analysis of results. One such paper is [16], where the authors compared the performance of five ML algorithms: random forest, extra tree classifier, XGBoost classifier, decision tree, and long short-term memory (LSTM) on the task of sentiment analysis of a dataset of COVID-19 tweets using different experimental settings. For this, they utilized two widely used feature extraction methods: Bag-of-Words (BOW) and Term-Frequency and Inverse Document Frequency (TF-IDF). Their results show that the extra tree classifier with an accuracy of 93.00% outperforms all remaining classifiers. Additionally, they confirmed that deep learning models do not perform well on this dataset.

For each English tweet, the algorithm generates a vector of non-normalized predictions for three sentiment classes: neutral, positive, and negative [16]. Researchers in this dissertation have used the same dataset (The IEEE data port provides COVID-19 tweets).

The implementation of Apache Spark is discussed in [17] it's implemented on 4 nodes Spark cluster for doing the scalability analysis. They concluded adding extra nodes in the cluster are boosting the overall performance. Their research was aimed to process twitter live streams on the distributed mode by applying MLlib library (Naïve Bayes, Logistic Regression and Decision trees) of Apache Spark for the sentiment on their experimental setup. In this dissertation, the same preprocessing steps have been done. 4 nodes have been used for distributed processing. Then, Minhash as a feature extraction with NB, LR, RF and SVM in apache spark to process twitter as a batch processing on the distributed mode.

Table 1.1 : Related Work

Author(s), Year	Dataset	Preprocessing	Methodology	Evaluation Measures	Advantage	Disadvantage
Sanders, A. C. et al, 2020 [9]	Over 1 million Covid-19 mask-related tweets	NLP Techniques	clustering and sentiment analysis techniques	Average sentiment scores, divisiveness scores, and regression line slopes with 95% confidence intervals, and qualitative descriptions of time series trends.	Topic clustering based on mask-related Twitter data offers revealing insights into societal perceptions of COVID- 19 and techniques for its prevention. Relay narratives for each theme using automatic text summarization.	the BART-based decoder is a generative language model which creates summaries autoregressively by repeatedly sampling from next-word probability distributions over an entire vocabulary. Large or irregularly shaped subclusters may be poorly represented by the tweets immediately surrounding the subcluster center.
Imran, A. S. et al ,2020 [10]	Trending Hashtag # Dataset + Kaggle Dataset + sentiment140 dataset + Emotional tweet dataset	POS Stop words & duplicates removal	Deep learning (Deep LSTM) DNN LSTM+FastText LSTM + GloVe LSTM + GloVe Twitter LSTM without pre-trained embedding	F1-score, Accuracy	Provide a new perspective to readers and the scientific community interested in exploring cultural similarities and differences from public opinions given a crisis.	1. Uses twitter for sentiment and emotion extractions, whereas other social media platforms like Facebook, Instagram etc. are not covered for keeping this work to a manageable complexity level. 2. Not able to discriminate in such contexts. 3. Does not extract emotions or sentiments from multimedia contents.
Chakraborty et al. [11]	Two datasets: DATA_SET 1 226,668 tweets & DATA_SET 2 most re-tweeted tweets (23000)	Tweets cleaned from various symbols. Words from tweets are tokenized. Tweets labeled with sentiments. Number of positive,	Paragraph vector (Doc2Vec) Bag-Of-Words model	Accuracy	Understanding the role of sentiment analysis and opinion mining in COVID-19	Doesn't have the features to attend multilingual tweet.

		negative and neutral tweets calculated. • Stop Words removed from tweets • Tweets lemmatized Tweets stemmed Tweets transformed to vectors • Most frequent words				
E. Alomari, R. Mehmood, and I. Katib, 2019[12]	Arabic tweets in the Saudi dialect	1. sparkConnector is used to connect to MongoDB. 2. The tweets are loaded and saved in Spark DataFrame. 3. Iterating over the tweets to remove all numbers, English alphabets and punctuations. 4. Removing the entire hashtag (#, _) and keeping the keywords. 5. Remove all Arabic diacritic and vowel marks such as Shaddah. 6. The text is divided into tokens. 7. Normalize the tokens by replacing letter that has different forms into the basic shape. 8. The Stop Words are filtered using the Arabic stop words list in the Natural Language Toolkit (NLTK).	1. Feature Extractor's algorithm (TF-IDF) provided in Spark ML package. 2. Classification (Tweet Filtering) - machine learning algorithms in the Spark ML package. a. Split the manually labeled data into training sets (80%) and testing sets (20%). b. Build and train model using NB, SVM, and LR algorithms on the training set. c. Evaluate models over the testing set. 3. Event Detection a. Build and train classifier using the NB, SVM, and LR algorithms. To train the events classifier. b. Manually label part of the filtered data from the previous step into eight event categories, which are Fire, Weather, Social	Precision, Accuracy, Recall , and F1-score	Able to detect road traffic related events as well as their location and time, automatically, without any prior knowledge of the events to enable smarter transportation.	Under sampling leads to loss of information.

		9. If the remaining number of tokens is equal to zero, the tweet is excluded from the analysis.	Events, Traffic Condition, Roadwork, Road Damage, Accident, and Road Closures. c. Divided the events into small-scale and large-scale events based on the number of tweets. d. The tweet can be about Traffic Condition and Accident at the same time. They treat each label as a separate binary classification problem (eight binary classifiers) to address the problem. e. Consider the tweets about the event as positive while all the remaining tweets about the other types of events as negative. f. Random under sampling for the negative (majority) class to make the dataset balanced before evaluation. 4. Extract the time of occurrence using the time, and date information from 'created_at' attribute in the tweets object. 5. Extract information about each event including location information using the top frequent terms since people usually refer			

			to the event place using the hashtag. 6. Model evaluation. 7. Validate the effectiveness of the event detection approach a. Extract the top vocabularies from the tweets of each detected events. b. Use these vocabularies to search in the official news/ newspapers websites to confirm the occurrence of the events. c. Compare the extracted information by the method including time and location with the real information in the official sources.			
Samuel, J., Ali, G., Rahman, M., Esawi, E. and Samuel, Y., 2020 [13]	Tweets from Twitter® API were extracted using R language. Around 9 lakhs tweets	Data Acquisition and Preparation. Word and Phrase Associations. Geo-Tagged Analytics. Association with Non-Textual Variables	Naïve Bayes and Logistic Regression classifiers have been used. The accuracy for shorter tweets was obtained as 91% and 74%, respectively.	Accuracy	To trail the growth of panic amongst Twitter® users based on a specific keyword.	The paper analyzed the sentiments based on a single keyword tracking based on only the fear of the people based in USA. Further aspects barring geographical constraints could also be explored.
G. Barkur, Vibha and G. Kamath, , 2020 [14]	Datasets was collected using Twitter® application interface by R. Approximately 24 thousand tweets were used by extracting from the handles	Wordcloud of the extracted tweets.	Analysis was done by the help of software R and by using WordCloud only.	NA	To analyse the sentiments of Indians post lockdown imposed by the government	Very few tweets were considered of a particular country. The study portrayed that Indians took the strategy of the government positively on imposing the lockdown.

	#IndiaLockdown and #IndiafightsCorona within the time span of 25th to 28th March, 2020.					
Cotfas, L. A., 2021 [15]	COVID-19 Vaccine Stance Data set.	1. Removal of links 2. All the user mentions, links and email addresses have been normalized. 3. The emoticons have been replaced with the corresponding words. 4. Minor spelling mistakes have been automatically corrected to improve performance. 5. Contractions and hashtags have been unpacked. 6. Elongated words have been corrected and annotated. 7. All the letters have been converted to a lowercase representation. 8. Additional processing has been performed through Natural Language Toolkit (NLTK) library.	1. Feature extraction the raw textual data is converted to numerical feature vectors. a. Bag-of-Words (BoW)+ word embeddings (Datastories , GloVe and FastText) and BERT have been used. b. Term Frequency - Inverse Document Frequency (TF-IDF) c. Various combinations of unigrams, bigrams and trigrams have been considered as features for the machine learning algorithms. 2. The classical machine learning and deep learning classifiers (MNB, RF, SVM, Bidirectional Long Short-Term Memory (Bi-LSTM) and Convolutional Neural Network (CNN). 3. The classification models are trained	Persian, F-score, Recall, Accuracy.	Monitor the evolution of The stance towards COVID-19 vaccination from tweets, by matching the number of Twitter messages with the main events reported by the media in the analyzed period.	1. The selected dataset, which only includes tweets extracted between November 9, 2020 and December 8, 2020, written in English. 2. Limitation is represented by the classification algorithms selected.

			using 2600 tweets and evaluated using the remaining 649 tweet.			
Rustam, F. et al, 2021 [16]	The IEEE data port provides COVID-19 tweets.	1. Removal of usernames and links. 2. Removal of punctuation marks and conversion to lower case. 3. Removal of stop words and numeric values. 4. Stemming.	1. TextBlob technique on the dataset to find sentiment scores. 2. The dataset is split into a training set and a testing set with a ratio of 80:20 (6022: 1506 Tweets Tweets) 3. Extract features of tweets (TF-IDF, BoW, Concatenation of BoW and TF-IDF). 4. Classification algorithms (RF, XGboost, SVC, ETC, DT).	Accuracy, precision, recall, and F 1 score	Using a Log Loss function, which is helpful to minimize the loss and improve accuracy in XGBoost.	The results cannot be conclusive without further investigation of LSTM on other datasets or with more tuned architecture.
H. Elzayady, K. Badran and G. Salama,2018 [17]	1.578.627 classified tweets (the dataset are University of Michigan Sentiment Analysis competition on Kaggle and Twitter Sentiment Corpus by Niek Sanders) 70% training data 30% testing data	• Removal of URL's • Removal of special symbol • Removal of Username • Removal of Hashtag • Removal of additional white spaces • Removal of stop words • Lower casing • Standardizing word	Unigram feature TF-IDF technique NB LR DT Using apache spark MLlib	F-Measure Calculation time in one, two, or three nodes.	In case of adding extra nodes in the cluster, higher performance capability will be obtained.	Does not establish an online service that that uses the Spark Streaming.

Based on the advanced strategies introduced in literature review, most of the studies have announced comparable outcomes in terms of accuracy and reliability. Nonetheless, not entirety of the studies is based on the similar datasets. Also, not all studies are implemented in the same environment.

1.8 Dissertation Structure

This chapter introduces the general concepts of the study landscape. It presents the main problem of data classification and how the proposed system questions and objectives are introduced. The contribution of the proposed system is illustrated to address the main problems of data sentiment analysis using data mining techniques. In addition, the related works are discussed to illustrate the recent sentiment analysis of data with data mining techniques. The rest of this dissertation is organized as follows: -

Chapter two reviews the background of data analysis concepts and the Minhash, SVM, RF, NB, and LR algorithms.

Chapter three provides the details of the proposed system, which includes all architecture and algorithms.

Chapter four discusses and compares the results with the other techniques.

The final chapter gives the conclusions to the study, recommendations, limitations and suggestions.

CHAPTER TWO
THEORETICAL BACKGROUND

2.1 Overview

Social media provides the possibility of interaction between people despite their different attitudes and directions. The rapid development of websites and social networks provide a variety of tools such as Facebook and Twitter. It contributed to increasing opportunities for extracting and processing large amounts of data, including files, bookmarks, and multimedia like images or videos, that indicates vital communication between users of social media networks. The classification methods varied according to user information in the social network, as it is known that communication networks are heterogeneous and related differently.

This chapter presents a theoretical background of sentiment analysis, general machine learning techniques, cloud computing, detailed pre-processing steps, Minhash algorithm and performance metrics. In these days, healthcare is an important aspect and contains huge data that need techniques for processing, storing, and retrieving this data. In this study, machine learning techniques were used to process data and obtain knowledge of it, using cloud computing to store and retrieve the data. A healthcare cloud as shown in Figure (2.1) is built to link many machines and servers expressly devoted to serving the healthcare industry's needs. The healthcare facility is given to the customer over an Internet link, who may either be a doctor or a patient. The cloud infrastructure helps authorized customers to securely control third party-managed hardware and applications.

Machine learning is a method utilized to extract insightful and interesting data from raw data. Machine learning is very useful technique generated to help people focusing on the most important information. Therefore, it can be used for many applications such as market analysis, customer reservation, cheat detection, science investigation and so on. Basically, machine learning has so many efficient techniques: classification, clustering, and association rule [18].

This dissertation employs one of the most effective common techniques which are classification.

2.2 Social Networks

Twitter is a free social networking micro - blogging site that allows users to send brief messages known as tweets. Using a variety of platforms, Twitter users can broadcast tweets and follow the tweets of other users. Tweets and tweet replies can be sent through text message, desktop software, or the Twitter.com website.

2.3 The data types in social network

The data types in social networks are classified into unstructured, structured, and semi-structured.

2.3.1 Unstructured data type

This type of data cannot be identified by its structures, i.e., it cannot be stored in rows and columns in interconnected databases. The data is stored in an unbound way without any defined scheme of storage, for example, storing various pictures and videos from social network sites. One advantage of this type is that it does not require additional components for classifications purposes. However, the limitation of this type is that it cannot control the navigation within the unstructured content. Example for unstructured data is text, audio, images, video, web chats, logs, and so on [19]. The dataset in this dissertation is an unstructured data.

2.3.2 Structured data type

This type of data in social media networks is based on a graph structure as a simple framework. The data is described in the scheme of a graph G = (V, E), as the V is a collection of nodes/objects, for instance, people, organizations, or products, while E represents a group of edges or relationships for nodes connect response patterns. Measuring and dealing with this type of data is done by analyzing the social network. As well, applications that are based on graph

analysis based on the intelligence methods of correlated data [20]. The Figure (2.1) shows the example of relational structured database type.

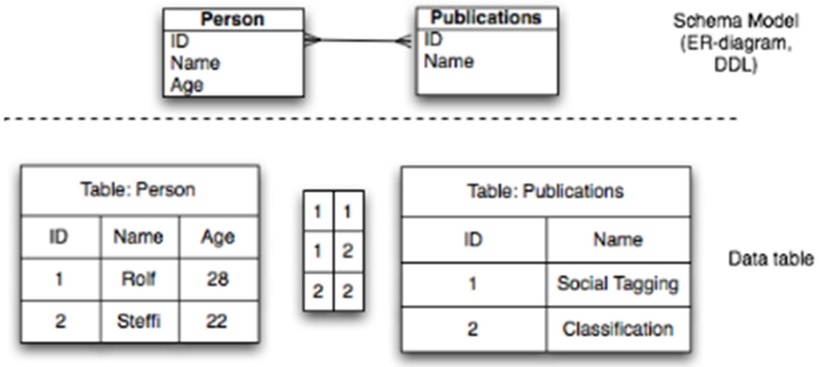

Figure 2.1: Example of Structured Relational Database [21].

2.3.3 Semi-structured data

It is considered a special case of structured data that does not depend on a formal and traditional structure to represent data models for interconnected data, and therefore contains signs and other evidence for the separation of semantic elements as well as the adoption of a hierarchy to represent data. For example, the data obtained through e-mail and deposit data is often represented in specific forms such as JavaScript Object Notation (JSON) and Extensible Markup Language (XML) [22]. The Figure (2.2) shows the Resource Description Framework graph representation as a form of semi-structured data type.

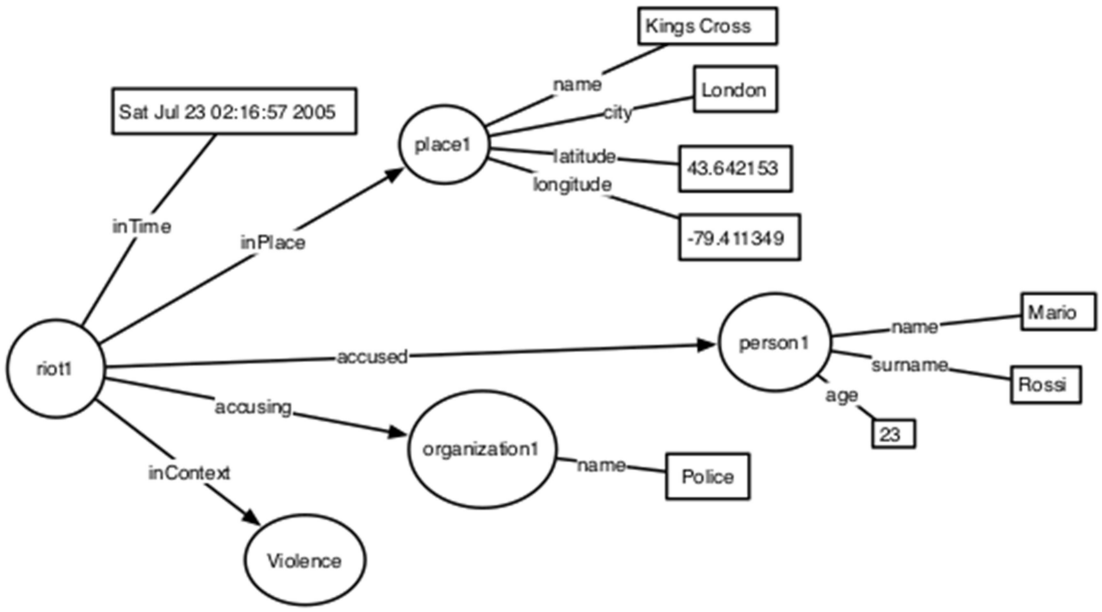

Figure 2.2 : The RDF Graph Representation [23]

2.4 Sentiment Analysis

Sentiment analysis can be defined as a process that automates mining of attitudes, opinions, views and emotions from text, speech, tweets, and database sources through NLP. Sentiment analysis involves classifying opinions in text into categories like "positive" or "negative" or "neutral". Spark's ML was applied in this study. It is one of the machine learning libraries that might be utilized for large-scale data classification, however because Spark ML is a new library, there was not much study done on it. According to the researchers' best knowledge, only a few studies have been done to analyze sentiment for large-scale data using Spark's ML, therefore more research is needed in this field. This study seeks to conduct novel sentiment classification experiments on large-scale data using Spark's ML, comparing the performance of several ML classification algorithms. The used dataset is unstructured type because it has text feature which is tweet text. The unstructured data has described above in 2.3.1.

2.5 Data Preprocessing

Data preprocessing is a broad field that consists of a variety of different methods and techniques that are intricately interlinked. One of the machine learning techniques that are used to transform data into an understandable method for the model used is preprocessing the data which are found in the dataset.

2.6 Minhash Technique

This method has been discovered by Andrei Broder. It was initially used for AltaVista search engine purposes to detect the presence of duplicate pages on the web and remove them from search results, as well as in the process of solving the problems of grouping on large-scale documents by finding similarities between the vocabulary group for documents [24].

Minhash is a data mining algorithm used to find similarity estimations between tweets. The algorithm's key feature is the large number of hash functions it has. This dissertation incorporates several Minhash algorithms. Equation (2.1) shows the general form of the Minhash [24].

$$h(x)= (a\ x + b) \bmod c \quad \ldots (2.1)$$

Where (a and b) are two random numbers, x is the hash value, and c is a prime number greater than the whole single set's maximum number. The MinHash algorithm is used in this paper to produce hash keys for each class in the dataset.

2.6.1 The Cyclic Redundancy Check (CRC32) hashing technique

CRC is an error-detection code that is widely used in digital networks and storage devices to detect unintentional data alterations. In these systems, data blocks are assigned a brief check value depending on the remainder of a polynomial division of their contents. When the data is retrieved, the computation is performed, and if the check values do not match, corrective action against data manipulation can be done. CRCs can be used to remedy mistakes [25].

The algorithm is based on cyclic codes, and the check (data verification) value is a redundancy (It amplifies the message without providing additional information). CRCs are widely used because they are simple to build in binary hardware, to analyze analytically, and to detect frequent errors in transmission channels caused by noise. Because it has a specified length, the function that generates the check value is occasionally employed as a hash function.

2.7 Chernoff Bounds

Chernoff bounds are another kind of tail bound. Like Markoff and Chebyshev, they bound the total amount of probability of some random variable Y that is in the "tail", i.e., far from the mean [26].

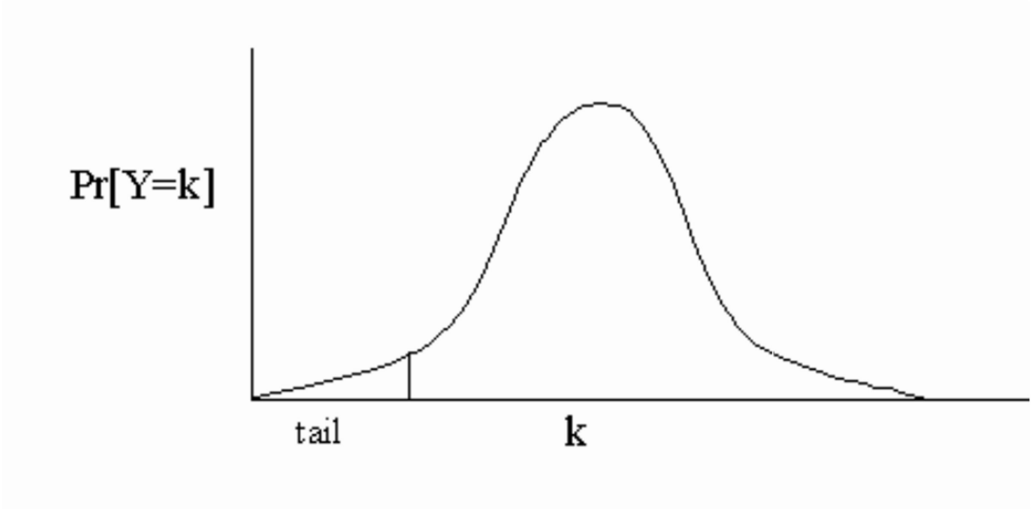

Figure 2.3: Chernoff Bounds

2.7.1 Chernoff Bounds (lower tail)

Let $X_1, X_2, \ldots, X_n$ be independent Poisson trials with $Pr[X_i = 1] = p_i$. Then if X is the sum of the X_i and if μ is E[X], for any $\delta \in (0, 1]$[26]:

$$Pr[X < (1 - \delta)\mu] < (e^{-\delta} | (1-\delta)^{(1-\delta)})^{\mu} \quad \ldots(2.2)$$

This bound is quite good but can be clumsy to compute. It can be simplified to a weaker bound which is:

$$Pr[X < (1 - \delta)\mu] < \exp(-\mu\delta^2/2) \quad \ldots(2.3)$$

the simplified bound makes it clear that the probability decreases exponentially with distance δ from the mean.

2.7.2 Chernoff Bounds (upper tail)

Let $X_1, X_2, ..., X_n$ be independent Poisson trials with $Pr[X_i = 1] = p_i$. Then if X is the sum of the X_i and if μ is E[X], for any $\delta > 0$ [26]:

$$Pr[X > (1 + \delta)\mu] < (e^{\delta} / (1 + \delta)^{(1+\delta)})^{\mu} \quad \ldots(2.4)$$

2.8 Apache Spark

Apache Spark is a fast and general motor for extensive handling of information on a scale [27]. Essentially, it is the advance of the development of Hadoop. Hadoop is the MapReduce demonstrate open-source execution and is widely used for conveyed preparation by different servers. It is ideal for cluster-based procedures when all the knowledge must be encountered. Whatever it may be, its execution drops rapidly for writing other problems (e.g., when iterative or chart-based calculations need to be managed). Spark is a series of numerous tightly organized modules and conquers Hadoop's problems. It has an execution motor for Directed Acyclic Graph (DAG) that supports cyclic information stream and recording in memory. It can then run programs up to 100x faster in memory than Hadoop, or 10x faster on the network [28]. Spark offers APIs to support a fast application development in various languages including Java, Python and Scala [29]. In addition to this, Spark can access various sources of information, such as HDFS or Hbase [30].

Apache Spark is a distributed computing framework that is open source. Matei Zaharia designed it in 2014 and contributed it later that year to the Apache Spark Foundation. The design is built on top of the HDFS.

It operates by distributing data processing among several worker nodes. A master node, which dispatches and arranges dispersed tasks, oversees the worker nodes. Spark, as a result, necessitates the use of a cluster manager as well as a distributed storage system. Because of its faster in-memory data engine and developer-friendly API, it is the framework of choice [31].

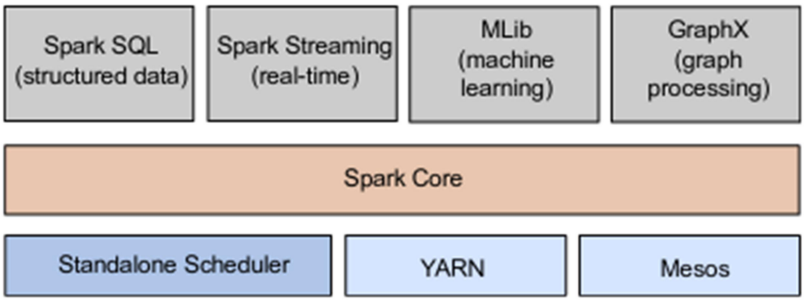

Figure 2.4: Apache Spark Ecosystem [32]

Spark is a computing engine that schedules, distributes, and monitors applications comprising multiple tasks across nodes of a computing cluster [32]. For cluster management, Spark supports its native Spark cluster (standalone), Apache YARN [33], or Apache Mesos [34]. At a higher level, it provides support for multiple tightly integrated components for handling various types of workloads such as SQL, streaming and machine learning.

The API sends a function to Spark and utilizes a driver application to call parallel operations on an RDD. The core coordinates parallel execution across the clusters available.

Until Spark 2.1.x, the primary API was RDD. It is a read-only, fault-tolerant dataset collection that is distributed among cluster servers for processing. The Dataset API is promoted in Spark 2.2.x. However, RDD is still in use. Spark also leverages Broadcast variables and Accumulators, in addition to RDD, to enable clusters share common read-only data and reduce program size [35].

Spark SQL, Spark Streaming, Machine Learning library, and GraphX are the four components that sit on top of the Spark Core. Spark SQL is a Spark module that works with structured data in the form of data frames. To query and handle data, it has a SQL-like interface [36].

In addition to Spark's batch processing, Spark Streaming provides for real-time data processing. It separates the incoming data stream into small batches and runs batch procedures in Spark. Both procedures share the same code and infrastructure, which saves time and effort [36].

GraphX, a distributed framework developed on top of Spark, is used to handle graph structures. It enables users to interactively construct and analyze graph organized data [37].

The Machine Learning library enables the distributed implementation of machine learning pipelines, considerably reducing total processing time. A growing number of machine learning methods are available for classification, clustering, collaborative filtering, regression, and dimension reduction. Because all the methods are distributed, structured datasets may be easily extracted, selected, and transformed. It contains tools for creating, analyzing, and tuning machine learning pipelines, as well as saving and loading algorithms, models, and pipelines [38].

Since Spark's release, the RDD API has been the main API. It is an immutable set of data items that's spread throughout the cluster's nodes. The operations provided for RDDs [27] might then be used to run these nodes in parallel. The processes are divided across the cluster and run in parallel, which reduces processing time [39]. RDDs are fault-tolerant because they can be recreated if the sequences that created them are kept track of.

They can be created in one of two ways: either by parallelizing an existing dataset in your application or by referring to a dataset from an external storage program like HDFS, Hbase, or others [27].

“Transformations” and “actions” are the two sorts of operations possible with RDDs. After performing calculation on RDD, Actions return a value to the driver program, whereas Transformations create a new RDD from an existing one. All transformations in Spark are lazy, meaning they do not calculate results immediately. Instead, they remember the necessary computation and only use it when the driver program performs an action. Spark can run more efficiently with this architecture in place.

The most frequent transformation is "map," which passes each RDD element through a defined function and returns a new RDD, and "reduce," which passes the RDD through a defined function and reduces it to a single value.

As previously stated, transformations such as map are lazy operations that are evaluated only when reduce or another action on the RDD is called. Spark's ability to cache datasets in memory is one of its most intriguing features. When a dataset is persisted, each node that utilized it to conduct a computation saves it in partitions so that it can be reused later. This makes it easier to work with datasets. The datasets can be saved in several ways, including memory only (the default), memory and disk, and disk only. These storage tiers can also support simultaneous replication across two nodes.

Spark is a workflow application at its heart. In many ways, however, it improves on earlier workflow schemes, including [27]:

1. A better way to deal with failures.

2. A more effective method of grouping tasks and scheduling function execution across compute nodes.

3. Looping (which practically removes it from the acyclic workflow class of systems) and function libraries are integrated into programming languages. The RDD, is Spark's core data abstraction. A RDD is a set of items of the same kind. The files of key-value pairs used in MapReduce systems are the primary example of an RDD.

1. A more effective approach of dealing with failures.

2. A more efficient technique of job grouping and function execution scheduling across computing nodes.

3. Programming languages provide looping (which effectively removes it from the acyclic workflow class of systems) and function’s libraries. Spark's main data abstraction is RDD. A RDD is a collection of similar goods. The major example of a RDD is key-value pairs files used in MapReduce systems. Unlike

MapReduce's key-value-pair abstraction, there are no restrictions on the type of things that make up an RDD [27].

They are also the files that are passed around from one department to the next. RDDs are "distributed" in the sense that they are typically split into chunks and stored across multiple compute nodes. They are "resilient" in the sense that it can be expected to be able to recover if some or all RDD chunks go missing. A Spark program consists of a series of steps, each of which performs a function on an RDD to generate another RDD. Such operations are referred to as transformations. It is also possible to convert data from a nearby file system, such as HDFS, to an RDD, as well as to convert an RDD back to a nearby file system or to generate a result that is really returned to a Spark program. Actions refer to the above type of service [38].

2.8.1 MapReduce in Apache Spark

The Map transformation takes a function as a parameter and applies it to each part of an RDD, resulting in another RDD. This operation is like Map in MapReduce, but it is not identical.

To begin with, a Map function in MapReduce can only be used on a key-value pair. Second, in MapReduce, each key-value pair generated by a Map function is treated as an independent component of the Map function's performance.

In Spark, a Map function can be applied to any object type and always returns the same result. Although the output object may take the shape of a collection, it is not the same as creating several objects from a single input object. Spark has Flatmap, which is like Map in MapReduce but does not require all types to be key-value pairs to build a collection of objects from a single object.

Filter is a Spark operation that works like a constrained version of Map. The Filter transformation accepts a predicate that applies to the type of objects in the input RDD instead of a function as a parameter. The filter function returns true or false for each object, and the output RDD of the Filter transformation

contains only those objects from the input RDD for which the filter function returns true [40].

In Spark, reduce is an event rather than a transformation. Instead of returning another RDD, reduce accepts an RDD and returns a value. Reduce accepts as a parameter a function that takes two T-type elements and returns another T-type element. Reduce is applied to each pair of consecutive items in an RDD with form T elements, reducing them to a single element. When there is only one variable left, reduce is utilized. When the Reduce, instance is applied to an RDD with integer elements and the parameter is the addition function, the outcome is a single integer that is the sum of all the integers in the RDD. If the function parameter is an associative and commutative function, such as addition, the sequence in which the input RDD components are merged makes no difference. However, if any combination of elements in any order has been satisfied, any function can be used [40].

2.8.2 Spark Context

Spark functionality is accessed through the Spark Context. It enables the driver application to connect to clusters using a cluster manager such as YARN, Mesos, or other similar software. Per JVM, only one SparkContext can be active at a time, and it must be ended using stop() before another can be generated [41]. To create the SparkContext, SparkConf is needed. It saves configuration data such as cluster size, cores, and the application name used to link the driver software to cluster, among other things.

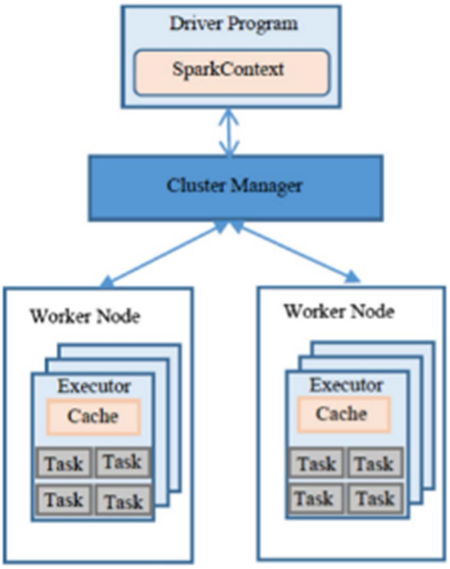

Figure 2.5: Apache Spark architecture [42].

The link between the driver application, the cluster resource management, and the executors is depicted in Figure (2.5). Each cluster consists of one driver node and one or more worker nodes. The driver application can access these worker nodes' executors via the Spark Context.

As soon as the driver program starts, the Spark Context produces a job and divides it into phases. The Spark Context of each executor organizes these stages into tasks. These executors are responsible for running the application's user code, as well as computing and caching data. The data is sent to the driver application [43].

2.8.3 Cluster Operation Modes

The Spark Context oversees managing Spark applications that are clustered. It can communicate with a variety of cluster managers to allocate resources across applications, as well as send tasks to executors and receive their results [43].

Spark supports four different types of modes: local, standalone, Apache Mesos, and Hadoop YARN.

Local mode, also known as Pseudo cluster mode, is a single JVM deployment mode in which the driver, executor, and master are all on the same computer. The default parallelism is determined by the number of threads specified in the master URL [44].

The basic cluster manager provided by Spark is standalone mode. It is made up of a master and workers who have memory and cores allocated to them. Unless otherwise specified, it will take up all cores by default. Automatic recovery, a web UI for cluster, SSL encryption and job status are all available.

For establishing and administering large-scale clusters, Apache Mesos comes in handy. It uses dynamic resource allocation and isolation to handle workload. It confirms workers' identification with the master and offers frameworks for resource requests and worker allocation.

In Hadoop YARN, the functions of resource management and task scheduling are divided between a Global Resource Manager and a per-application Application Manager. Permission security, a web UI for controlling resources, and manual recovery capabilities are all available. The local spark and the Hadoop YARN modes have used to submit the application in this study.

2.9 Cloud Computing

Cloud computing is a powerful technology used in the management of applications and information on demand and is reliable and coordinated and this institution do not require maintenance or configuration of the infrastructure of its own computer. It also provides resources, applications, programs, and services to its customers continuously and at low cost, depending on the number of resources allocated to them and the extent of consumption. Cloud computing has easy and on-demand network access, such as a server or memory access. Cloud computing is a profitable business because it has important features such as resource pooling, fast flexibility, demand management, and fast network access. Despite the good characteristics of cloud computing, it has several disadvantages, the most important of which is protection. Cloud computing suffers from many security problems such as data access, reliability, integrity, identity management, and others [39].

In this new era, cloud computing and data sharing are becoming increasingly important. With the use of the Internet, the data may be viewed and shared via

many websites. For improved performance, it also offers a customizable user architecture, storage capacity, and device compatibility. In the cloud storage context, data confidentiality and performance are critical considerations [45].

2.9.1 Characteristics of Cloud Computing

Cloud Computing has several characteristics that make it one of the most rapidly growing industries today. Cloud services' versatility, in the form of an ever-growing collection of tools and methodologies, has expedited their adoption across industries.

1. Broadband Access: the services provided by the cloud can be accessed from anywhere at any time, through powerful devices such as laptops, tablets, and smartphones.
2. Resource Polling: The cloud computing resources are available to many people via a multi-tenant model. The users do not have any information about the location of the services provided by the service owner.
3. Rapid Elasticity: the service provider provides the services and resources needed by the user through the payment on use. Transparency between users and service providers is essential.
4. On-demand self-service: through self-service requests, services are requested by users and services are processed by service providers such as storage, processing capacity, etc. without any interference between them.
5. Rapid Elasticity: services are added or removed automatically at the request of the user, and the user has confidence in the unlimited ability to develop at any time [46].

2.9.2 Deployment Cloud Models

Cloud deployment models indicate how the cloud services are made available to users. The four deployment models associated with cloud computing is as follows:

i. public clouds: Service providers give these services to the public, and the public cloud delivers many of the essential features and benefits of service

providers, such as the lack of initial capital investment in infrastructure and the elimination of risks for infrastructure providers. But the public cloud lacks precise control over network settings, data, and security, and this reduces its effectiveness in many projects and businesses.

ii. Private clouds: also called internal and private withdrawals, these withdrawals are designed for exclusive use by one organization, through external providers or a specific organization that is controlled, created, and managed, and enjoys a high degree of reliability, protection, and security, but many criticized these withdrawals for being similar in Servicing and monopolizing the servers traditionally and offering services and features without paying upfront capital costs.

iii. The combination of public cloud and private cloud: it is called hybrid clouds as in the Figure (2.6), which try to solve the boundaries of all projects, where most public clouds use part of the infrastructure services, and the other part remains with the private cloud. The hybrid cloud is highly flexible compared to the public and private cloud, especially because they have control and security. In a way that takes care of application data compared to public clouds, with facilities to expand and reduce service on request. Pro-grammatically, this service requires careful determination of the best coordination between private and public cloud components.

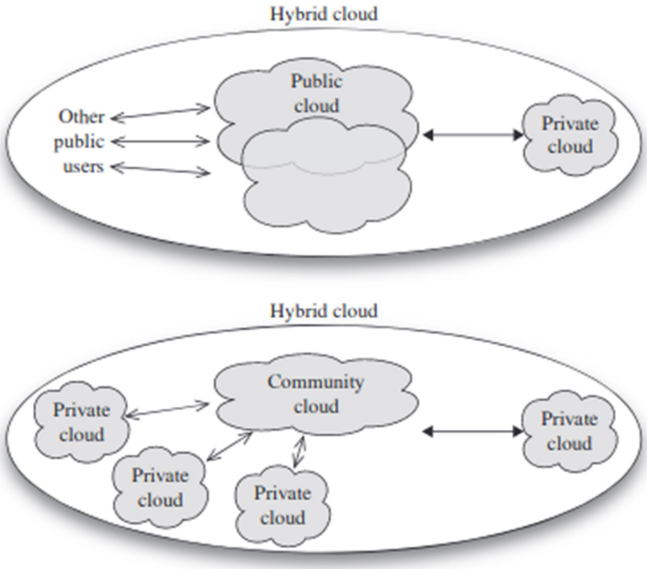

Figure 2.6: A hybrid cloud [47]

iv. An alternative workaround is to break the constraints posed by Virtual Private Cloud (VPC). VPC improves Virtual Private Network (VPN) technology, allowing service providers to build up and customize their own security. It acts as a model on the surface of public clouds. This represents a major difference with the rest of the draw. VPC consists of visualized applications and servers as well as the backbone network. It represents the bulk of most companies and has an easy migration mechanism between cloud-based infrastructure and other virtual network layers [48]. AWS has been reserved as a public cloud computing.

2.9.3 Service Model Types

The cloud services can be classified into:

i. Platform as a Service (PaaS): through which applications can be developed by users, implemented, and installed, these applications can be created by various software languages, services, tools, and libraries supported by the service provider. So, it is a development platform provided as a service. Like Google Apps Engine, where applications can be developed in several languages like Python and Java, Software Development Kit (SDK) provides both two languages and has a plug-in to develop the Eclipse environment.

ii. Infrastructure as a Service (IaaS): in this model, the service provided by the cloud computing infrastructure includes (usually virtual) servers with limited storage or processing capacity, and user can see and control all operating systems or storage resources as well as the applications available on it, and but it can control the network settings. For example, the Amazon in the department Simple Storage Service (S3), all users can access and store their data using a web service interface, as well as Amazon's Elastic Compute Cloud (EC2) that allows clients to create and configure virtual machines.

iii. Software as a Service (SaaS): applications that use the infrastructure of cloud service providers can be used for users, in a way that enables customers to access their applications through specific interfaces, and users cannot manage the cloud network components, services, operating system and memory, and users can Sometimes you manage certain settings for these applications [49][41]. Regarding the approach, the Platform used as a service type, being a good environment for building, modifying applications and installing software, in addition, it supports many programming languages and contains multiple libraries and tools. Also, it allows ordinary users to develop their capabilities in various systems.

AWS is one of the modes which is used and compared its results with other modes. AWS is an Amazon cloud computing platform that comprises IaaS, PaaS, and packaged SaaS. AWS services can provide a company with resources like as computing power, database storage, and content distribution.

2.9.4 Common Cloud Computing Services

This section will illustrate the most common cloud computing product:

2.9.4.1 Amazon Web Services (AWS)

Cloud Computing provides a suite of cloud services such as AWS that allow organizations and users to develop their applications and services on-demand with a cost that is commensurate with the goods. Simple Object Access Protocol (SOAP) and Representational State Transfer (REST) protocols can be used to

connect to AWS using HTTP. Cloud computing allows users to run and administer various services in data centers, as well as manages applications, tools, and other requirements use software platforms. For example, EC2 is a virtual computing cloud where users can create, run, or download their applications and make as many changes and requirements as needed and can be replicated at any time. Users can fully control the nature of the programs loaded in the cloud, as well as with their counterparts so that they become their own devices. Through this feature, this feature increases the difficulty of Amazon to reduce the size of resources automatically.

EC2 provides the possibility of distributing resources in areas and different places geographically, so that they are designed and selected accurately to be saved from falling into error in cases of failure to access the areas available. They are characterized by the appropriate cost of communication and weak in cases of stagnation in the network and increase in peak conditions with customer requirements in the area where the cloud is located [50].

2.10 Machine Learning Techniques

Typically, machine learning is the procedure of extracting useful input from large groups of data. It defines as a computational operation of analyzing data to obtain patterns and useful information. The main aim of this method is to discover patterns that were already unnamed. So that once these patterns found, they can moreover utilize to make decisions for the evaluation of work. In the future, machine learning tools will predict future trends and take place in various fields: information technology, medicine, sociology, and physics. On other hand, machine learning is often referred to as knowledge discovery, data mining, and knowledge extraction [51].

The following explanation has a brief discussion on each machine learning steps as given below:

1. Extracting raw data and transforming it into pre-processed data.
2. Managing and storing data in a multidimensional data warehouse system.

3. Preparing data for data modeling and implementing data modeling through one of the following techniques:

A. Artificial neural networks (ANNs) are a type of non-linear predictive model that learns over time. ANN or neural network is other terms for the same thing. It is a computational example based on the functional side of neural networks [52].

B. Genetic algorithms are design optimization strategies that rely on a specific process such as original combination and natural selection.

C. Classification: A mechanism used to classify every record in used dataset based on a collection of classes.

D. Rule induction: Rule induction is the process of extracting if-then rules from data using statistical significance.

E. Clustering: It is cluster datasets into groups; the data in one cluster is similar and different from other clusters.

4. Lastly, the model gained after data modeling is transformed into knowledge and can be expounded in a graph or a table [53]. Some people treat machine learning as if it is the synonym for Knowledge Discovers in Databases (KDD). on the other hand, others believe that machine learning is a critical stage in the knowledge discovery process [54].

2.10.1 Classifications

Classification is a popular machine learning technique that uses a set of pre-classified examples to create a sample that can group records in huge datasets. The categorization algorithms used in this method are either decision trees or neural networks. Test data are utilized in classification to assessment the reliability of classification rules. The classification task picks a collection of records as input which usually called training set. Each record of these is collected of attributes and one of these attributes is indicated the class of the record. The main goal of classification is to discover a proper sample for the class attributes added to the function of the values of the rest of attributes. The

generated sample will be utilized later to predict the class attribute of the observed records [55]. In this dissertation, the following classification algorithms are used:

2.10.1.1 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised machine learning approach used for classification and regression. The basic form is SVM is a binary classifier that classifiers the data points into one of the two classes. SVM training involves determining the maximum margin hyperplane that separates the two classes. The maximum margin hyperplane is one which has the largest separation from the nearest training data point.. A standard SVM finds a hyperplane w.x-b=0, which correctly separates the training data points and has a maximum margin which is the distance between the two hyperplanes w.x-b=1 and w.x-b=-1, as shown in Figure (2.7).

The optimal hyperplane with maximum margin can be obtained by solving the following quadratic programming problem[56],

$$\min \frac{1}{2}\|w\|^2 + C\sum_i \xi_i \qquad \ldots (2.5)$$

Subject to $y_i(w \cdot x_i + b) \geq 1 - \xi_i,$

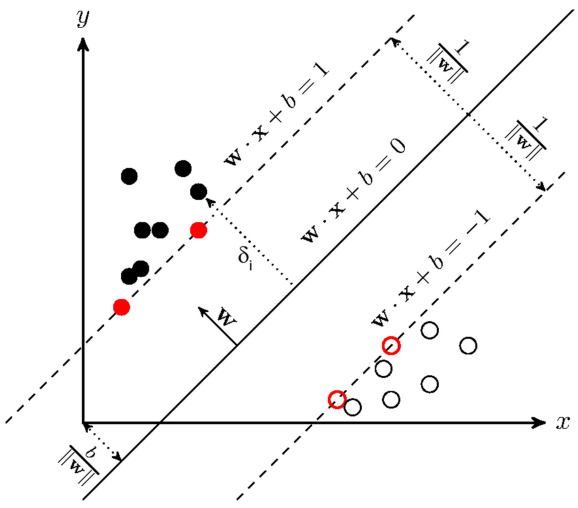

Figure 2.7: Margins of SVM

$\xi_i \geq 0$,1< I < l , where C is the soft margin parameter and ξ_i is a slack variable for the non-separable case. The optimal hyperplane is given as,

$$f(x) = sign(C \sum_{i=1}^{l} \propto i\ yi\ \ K(xi, x) - b) \quad \ldots (2.6)$$

Where αi is the Lagrange multiplier and K (xi,x) is the kernel function. A standard SVM is a two-class classier where the outcome is 1 or -1. When sets are not linearly separable, the data points in the original finite-dimensional space are mapped to a higher dimensional space where they can be separated easily. The performance of a SVM classifier depends on the selection of kernel, the kernel's parameters, and soft margin parameter C. The commonly used kernels include [56]:

- Linear: k(xi,xj) = < xi,xj >
- Polynomial: k(xi,xj) = (γ < xi,xj > + r)^d
- Radial Basic Function (RBF): k(xi,xj) = exp(-γ || xi-xj ||²)
- Sigmoid: k(xi,xj) = (tanh < xi,xj > + r)

Algorithm 2.1: Support Vector Machine [57]

Input: X: matrix with features to train, y:vector of labels, hyper parameters (n_neighbors, weights, metrics)

1: function **RSEARCHCV(n_neighbors, weights, metrics)**

2: Create a dictionary with the gamma, kernel, decision function values to try.

3: Use Randomized search on hyper parameters

4: return **best_estimator**

5: procedure **SVM_CLASSIFIER(X_train, y_train, x_test, y_test)**

6: Use RSEARCHCV to get the best_estimator

7: Fit the model with X_train, y_train

8: Predict labels for x_test

9: return **predicted values** → **Score can also be obtained by using y_test**

End

2.10.1.2 Random Forest (RF)

There are two ensemble learning paradigms: bagging and boosting. Bagging consists of creating many "copies" of the training data (each copy is slightly different from another) and then applies the weak learner to each copy to obtain multiple weak models and then combine them. The bagging paradigm is behind the random forest learning algorithm.

The "vanilla" bagging algorithm works like follows. Given a training set, B is random samples S b (for each b = 1, . . . , B) of the training set and build a decision tree model f b using each sample S b as the training set. To sample S b for some b, we do the sampling with replacement. This means that we start with an empty set, and then pick at random an example from the training set and put its exact copy to S b by keeping the original example in the original training set. We keep picking examples at random until the |S b | = N .After training, there is B decision trees. The prediction for a new example x is obtained as the average of B predictions:

$$y \leftarrow \hat{f}(\mathbf{x}) \stackrel{\text{def}}{=} \frac{1}{B}\sum_{b=1}^{B} f_b(\mathbf{x}) \qquad \ldots (2.7)$$

in the case of regression, or by taking the majority vote in the case of classification. The random forest algorithm is different from the vanilla bagging in just one way. It uses a modified tree learning algorithm that inspects, at each split in the learning process, a random subset of the features. The reason for doing this is to avoid the correlation of the trees: if one or a few features are very strong predictors for the target, these features will be selected to split examples in many trees. This would result in many correlated trees in our "forest." Correlated predictors cannot help in improving the accuracy of prediction. The main reason behind a better performance of model ensemble is that models that are good will likely agree on the same prediction, while bad models will likely disagree on different ones. Correlation will make bad models

more likely to agree, which will hamper the majority vote or the average. The most important hyper parameters to tune are the number of trees, B, and the size of the random subset of the features to consider at each split [58].

Algorithm 2.2: Random Forest [59].

Input: training set with "m" features

Output: predicted target as the **final prediction**

1: Randomly select "k" features from total "m" features.
Where k << m

2: Among the "k" features, calculate the node "d" using the best split point.

3: Split the node into daughter nodes using the best split.

4: Repeat 1 to 3 steps until "l" number of nodes has been reached.

5: Build forest by repeating steps 1 to 4 for "n" number times to create "n" number of trees.

6: Takes the **test features** and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome (target)

7: Calculate the **votes** for each predicted target.

8: Consider the **high voted** predicted target as the **final prediction** from the random forest algorithm.

End

2.10.1.3 Naive Bayes (NB)

Naïve Bayes is a probabilistic classification algorithm based on the Bayes theorem with a naïve assumption about the independence of feature attributes. Given a class variable C and feature variables F1, F2, ….., Fn, the conditional probability according to Bayes theorem is given as,

$$P(C|F1,\dots,Fn) = \frac{\mathrm{P(F1,\dots,Fn|C)P(C)}}{\mathrm{P(F1,\dots,Fn)}} \quad \dots(2.8)$$

Where, $P(C|F1,\dots,Fn)$ is the posterior probability, $\mathrm{P(F1,\dots,Fn|C)}$is the likelihood and P(C) is the prior probability and P(F1,…,Fn) is the evidence.

Naïve Bayes makes a naïve assumption about the independence every pair of features given as,

$P(F1,\ldots,Fn|C)= \prod_{i=1}^{n} P(Fi|C)$ … (2.9)

In practice, since the evidence P(F1,…,Fn) is constant for a given input and does not depend on the class variable C, only the numerator of the posterior probability is important for classification. Therefore,

$P(C|F1,\ldots Fn) \alpha P(C) \prod_{i=1}^{n} P(Fi|C)$ …(2.10)

With this simplification, classification can then be done as follows,

$C=\text{argmax}_C P(C) \prod_{i=1}^{n} P(Fi|C)$ … (2.11)

There are different versions of Naïve Bayes which differ in the naïve assumption made. Some of them include:

Gaussian Naïve Bayes: Gaussian Naïve Bayes assumes the likelihood P(F1,…Fn) as,

$P(F1,\ldots, Fn|C) = \prod_{i=1}^{n} P(Fi|C)$ …(2.12)

Where,

$x = \frac{1}{\sqrt{2\pi\sigma c^2}} \exp((-(Fi - \mu c)^2|2\pi\sigma c^2))$ …(2.13)

where μ is the mean and σc is the standard deviation for values in Fi in class C. Gaussian Naïve Bayes id suitable for problems in which the feature variables have continuous values which are assumed to have a Gaussian distribution.

Multinomial Naïve Bayes: Multinomial Naïve Bayes uses multinomial distribution for each of the feature variables. This is suitable for problems which have discrete features such as tweet classification.

Bernoulli Naïve Bayes: Bernoulli Naïve Bayes is also suitable for problems which have discrete features. The likelihood in Bernoulli Naïve Bayes is as follows[56],

P(Fi|C) = P(i|CFi(1-P(i|C))(1-Fi) … (2.14)

Algorithm (2.3) shows an example of binary classification using Gaussian Naïve Bayes.

Algorithm 2.3: Naïve Bayes [60]

Input: Training dataset and Testing dataset

Output: Class with maximum probability value

1: Build a frequency table using training dataset ∀ feature Vector ($F_1, F_2, \dots, F_n$) against every class C_i, where i =1 to k

2: Create the likelihood table ∀ feature against each class C_i

3: Compute the conditional probabilities for Test dataset ∀ Class:

$P(Ci|F_1, F_2, \dots, F_m) = \prod_{j=1}^{n} P(Fj|Ci).P(Ci|P(F1, F2, \dots, Fn)$

For $1 \leq i \leq k$

4: Calculate $\max_i P(Ci|F_1, F_2, \dots, F_n)$

5: End

2.10.1.4 Logistic Regression (LR)

is a form of regression used to predict the likelihood of a binary output from an input dataset modeled [60]. The standard logistic function (also known as the sigmoid function) is defined by (2.15) [61]:

$$P = \frac{1}{1+e^{-z}} \quad \dots (2.15)$$

where P is the frequency probability of the case and can take values between 0 and 1 and produce an S-shaped curve z the linear combination of the variables observed corresponds to that set in X. The Linear Regression model equation is defined by (2.16).

$$z = \theta_0 + \theta_1 x_1 + \cdots + \theta_i x_i \quad \dots (2.16)$$

where θ_0 is the intersection point of the model θ_i belongs to the regression coefficients and x_i to the independent variables for the LR model.

Thus, the LR model allows to determinate an event occurrence; the regression model can determinate if the autoclave is into a specific stage of the sterilization process.

Algorithm 2.4: Logistic Regression [60]

Input: Training dataset and Testing dataset

Output: Class label

Begin

1: For I = 1 to k

2: For each training dataset instance d_i:

3: Set the target value for the regression to $z_i = \frac{yj - P(\frac{1}{dj})}{[P(1|dj).(1-P(1|dj))]}$

4: Initialize the weight of instance d_j to $P(1|d_j)$. (1-P $(1|d_j)$

5: Finalize a f(j) to the data with class value (z_j) and weights (w_j)

Classification Label Decision

6: Assign (class label: 1) if $P(1|d_j) > 0.5$

7: Assign (class label: 2) if $P(1|d_j) < 0.0$, otherwise (class label: 0)

End

2.10.2 Association Rules

Association rules are used to discover repeated items set over a large set of data. This type of rules helps in making certain decisions in businesses such as catalog design and customer shopping behavior analysis. There are sets of records and each of which contains numbers of items from a known given collection, the essential goal of the association-rule is to make rules which can predict the existing of the item based on the existing of other items. For instance, the electronic shop that's so ever one is involved in understanding the proper way to propose items for marketing reasons and increase the purchases. The association rules manage records and arrange them in a way to find similar items easily [62].

For the example above, the association rules will look back to the void purchasing history and analyses it.

2.10.3 Clustering

Recently, clustering has been widely studied because it starts taking place in wide applications and in various areas such as search engines, web mining, and

information retrieval. Clustering is the most utilized data mining techniques due to its different applications which perform various functions, such as visualization, classifications, and document organization additionally to some more techniques like indexing and collaborative filtering.

Clustering is an automatic grouping technique of documents, so that documents in same group (cluster) are very similar to each other compared to documents in other clusters.

Thus, clustering can help in text document organization for effective and efficient browsing. A clustering of input set of items is built in a way that cluster separation measured based on number of terms like the underlying approximation measure and the maximized. Clustering is done for two reasons: data interpretation and data compression [62].

2.11 Performance Metrics

The model's performance will be assessed by calculating the experiment outcomes. The primary comparison in supervised classification issues is to match the true class label with the predicted one to calculate the accuracy.

A data point's result might be either a True Positive, TP (positive prediction and positive label), or a False Positive, FP (label is negative but prediction is positive), True Negative, TN (label is negative and prediction is negative) and False Negative, FN (label is positive but negative prediction). In addition to absolute accuracy, time spent can be considered.

This step is responsible for the classifier's achievement estimation function. The expected class labels of the testing documents from the classification phase, as well as the actual associated class labels, are the inputs for this phase. The results of comparing the expected class labels with the real class labels are used to estimate a classifier's success.

True Positives (TP) - It means correct values in actual and prediction for positives values which mean actual value yes and predicate value yes.

True Negatives (TN) - It means correct values in actual and prediction for negative values which means actual value is no and predicate value is no.

False Positives (FP) – It means incorrect values in actual and prediction for positives values which means actual value is no and predicate value is yes.

False Negatives (FN) – It means incorrect values in actual and prediction for positives values which means actual value is no and predicate value is yes.

Classification analysis evaluation can be calculated using a confusion matrix and performance criteria such as accuracy. Each column in the confusion matrix represents the predicted class, while each row represents the actual class. Confusion matrix can be used for two or more class matrix.

In this proposed system, many terms have defined to evaluate the total performance for this system: -

Accuracy: it can be defined as the total number of correctly classified of the positive instances. In other words, it is the ratio of the sum of true positive and true negative to the sum of all true and false prediction instances.

Also, it is an evaluation of the efficiency of an algorithm which can be defined in equation (2.17).

$$\text{Accuracy} = TP+TN/TP+FP+FN+TN \quad \ldots (2.17)$$

2.12 Summary

In this chapter, a wide and general discussion has been presented on most of the concepts that are practically implemented in this dissertation. It started with an introduction to the healthcare cloud, the general social network tools for researching and society sides explained, in addition to the data types in social network and their characteristics, explaining Sentiment Analysis, machine learning techniques, cloud computing and its multi-functionality. In addition, the cloud is explained and its characteristics, the most deployed models and the types of services provided. Descriptions of Minhash technique have been presented. Besides, the hashing technique CRC32 and its main characteristics have been explained. Finally, the performance metrics have been explained.

CHAPTER THREE

DATA ANALYSIS USING MINHASH-ML SPARK MODEL

3.1 Overview

The cloud computing provides resources for storage, computing, and network. Predictive analysis uses a combination of statistical, modeling, data mining, and machine learning techniques to examine and interpret twitter data, allowing the data miner to generate predictions about future events. Classification analysis of healthcare data set involves the analysis of tweets records.

This chapter includes the main concepts in this dissertation. The outlines of this chapter can be summarized in the following subjects: the first section proposes an introduction to the topic. The second section illustrates the proposed system in detail. The third section deals with the tweet's dataset and preprocessing dataset. The fourth section is showing the machine learning techniques. The last sections are the evaluation stage, and conclusions.

3.2 Proposed Methodology

The necessary software must be installed and met a few conditions. The system's environment variables must be adjusted after installing the needed software so that the software may access the path variables. Path variables are operating system environment variables that define which directories executable applications. Path variables must be specified for Spark, Hadoop, Java, R, and Rstudio. This process is required for both the local spark and the YARN client mode. One of the dissertation's objectives is to look at how machine learning algorithms perform in two Apache Spark modes. Four machine learning methods are developed on datasets with sizes for this purpose, allowing the Apache Spark cluster computing system to be better investigated and performance appraised. The structure of the proposed system consists of several stages that all cooperate to propose a technique for sentiment analysis in social media. The flow chart in Figure (3.1) illustrates the system architecture and describes the steps involved in the tweets sentiment analysis system that differ from traditional methods. To get the Twitter feed working, Consumer Key, Consumer Secret, Access Token, and Access Token Secret are needed. The following are the steps of get those four keys:

- Log in to the Twitter website at https://apps.twitter.com/app/new.
- Fill in the desired Application Name, Description, and website address, making sure to include the http:// in the whole address. The callback URL can be left blank if desired.
- Accept the Terms of Service and complete the form by clicking the Create the Twitter Application button.
- After building the Twitter application, go to the Keys & Access Tokens panel, and then provide access to the Twitter account to utilize it. To do so, go to the Create my Access Token page and click the Create my Access Token button. Finally, paste the Consumer key (API key), Consumer Secret, Access Token, and Access Token Secret from the screen into the Twitter Options page of the plugin and test.

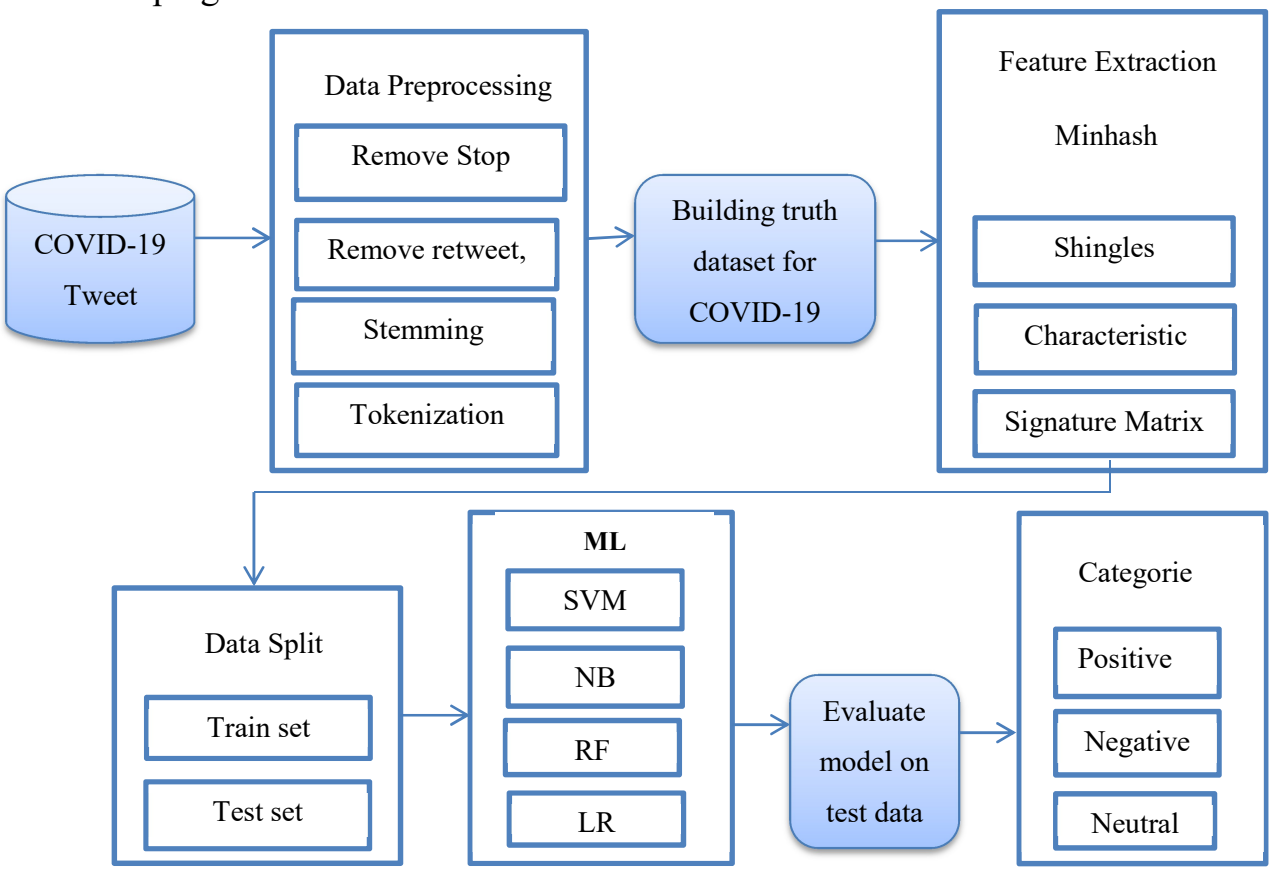

Figure 3.1: The Proposed System of Tweets Classification

Algorithm 3.1: ***Pseudocode for the proposed Approach***

Input : COVID-19 Vaccine Stance, IEEE COVID Data Port tweet data set which contain three columns tweet_id, tweet_text, label

Output: Classify the label of the tweet

(Accuracy, time_RF, time_NB, time_SVM, time_LR and positive, negative, neutral for tweet label).

// Preprocessing Stage

1: For each row in data set

2: Call Preprocessing Function

3: End for

// k-Shingling Stage

7: For each cleaned row in data set

8: Identify k=4 or k=5

9: Call Shingling Function

10: End for

// Minhashing Stage

11: For all Unique shingles

12: Call Signature Matrix Function

13: End for

// Partition data set to training and testing

14: For i in range (1: total number of rows of TSM-L Matrix

15: Split TSM-L Matrix according to Chernoff bound into training and testing data set

16: End for

17: Call Apache Spark Connection

18: Copy train dataset into Spark, and return a train Spark DataFrame.

19: Copy test dataset into Spark, and return a test Spark DataFrame.

20: Register a train Spark DataFrame and return train Spark Table

21: Register a test Spark DataFrame and return test Spark Table

22: Force a train Spark table to be loaded into memory

23: Force a test Spark table to be loaded into memory

// Random Forest

24: Call ml_random_forest Function

```
25: Create Confusion Matrix
26:  Calculate Accuracy
27:  Calculate time spent (time_RF)
// Logistic Regression
28: Call ml_logistic_regression  Function
29: Create Confusion Matrix
30: Calculate Accuracy
31: Calculate time spent (time_LR)
// Naïve Bayes
32: Call ml_naive_bayes Function
33: Create Confusion Matrix
34: Calculate Accuracy
35: Calculate time spent (time_NB)
//  Support Vector Machine
36:  Call ml_linear_svc Function
37: Create Confusion Matrix
38: Calculate Accuracy
39: Calculate time spent (time_SVM)
```

Algorithm 3.2: Apache Spark Connection

```
// Connect to Apache Spark  (Local Mode)
1: x = spark_connect(master = "local", version = "2.4.3")
//  Connect to Apache Spark (YARN Mode)
1: Sys.setenv(spark_home="/usr/lib/spark")
2: x = spark_connect(master = "yarn-client", version = '2.4.3')
```

Three independent human raters categorized the tweets' views on vaccination into three categories: in favor, against, and neutral. The in-favor tweets assigned to the class showed a positive attitude about the vaccination.

Tweets belonging to the against vaccination class express a negative opinion towards COVID-19 vaccination. COVID-19 vaccination is viewed negatively in tweets that fall into against category.

News about vaccine development and tweets that do not indicate a definite opinion, such as inquiries about the vaccine, informative tweets about vaccination, and off-topic tweets all are fall into the neutral category.

The two tweets dataset has three features, the first of which is the class label (1 for positive, -1 for negative, and 0 for neutral), followed by two features.

The authorization to view and print the tweets' text has gotten. OAuth is an open protocol to permit secure authorization in a straightforward and standard strategy from web, mobile and desktop applications. This system allows users to allow their consent to act on their sake without sharing the account password. After the client has given permission, OAuth will return a token, and this token itself awards get form demands on sake of the client.

The second stage of the proposed system is the data cleaning process which removes all white-spaces and punctuation marks, and changes the capital letters into small ones, in addition to removing all tags and data that are unnecessary to simplify the process throughout the following steps.

The preprocessing of the data can be considered as the most important stage. The aim of its steps is to make the data more understandable by machine. Therefore, uncertainty is decreased in feature extraction. Not all tweets' texts are consistent in the use of capital letters. Case folding is used to change all letters in the tweets into lowercase. 'a' to 'z' are accepted only.

Data cleaning is the act of removing irregularities from data to produce better ordered and structured data. At this stage, text was checked based on existing tweets. This step used to remove punctuation and corrects words that can damage the actual expression. It has been used in pipeline. It replaces all the substrings coordinating the Regular Expressions ("@[a-zA-Z0-9_]+", "&(lt)?(gt)? (amp)?(quot)?;") to delete the nickname and html labels from the tweet. We are doing it to maintain a strategic distance from handling of words that are not related to the sentence that the algorithm will analyze.

The results from Tokenizer function are the number of tokens in each sentence. Within the filtering stage, stop word algorithm has been used to delete the word less imperative or wordless. Stop words are non-descriptive words that can be displaced within the bag-of-words approach. It clears articles, prepositions, conjunctions, pronouns, as they are not semantically vital to characterize the opinion. Stop word removal can reduce index size and processing time. In addition, it can also reduce the noise level. The third stage involves the algorithms to deal with the data. First, the Minhash technique is responsible for generating a new matrix with new values, in form of several hashes and tweets. The number of hashes might range from 1 to 20, whereas the number of tweets indicates the actual number of tweets on which the similarity to be verified is based. Secondly, converting words to vectors, or word vectorization, is a NLP process. The process uses language models or techniques to map words into vector space, that is, to represent each word by a vector of real numbers. Meanwhile, it allows words with similar meanings to have similar representations. The fourth stage deals with the machine learning classification. The classifier operation is a term utilized both recognizing and circulation sorts of" things" into various gatherings and for the subsequent arrangement of classes, just as the task of components is to pre-setup classes. To characterize the wide importance given above is an essential idea and a piece of practically a wide range of exercises. Four classifier procedures have been used, namely NB, SVM, LR and RF.

Three criteria have been used throughout the five-stage continuous device assessment to test the Accuracy, CPU time. Accuracy is the most intuitive output indicator which is essentially the ratio of the cumulative input to a correctly discovered note. The performance of the classification algorithms has been evaluated by several evaluation metrics: accuracy, and computation time. The proposed system for this dissertation is shown in Figure (3.1) used to classify tweets.

3.3 Data Preprocessing

Given the fact that the tweets from the Twitter portal need a set of processors, the following preprocessing procedures have been adopted to normalize their content.

This was done with the tm package in R with its own Tokenizer class as in the Figure (3.2):

- Lowercase and derived letters to reduce word conjugation.
- Remove punctuations.
- Remove stop words.
- Removes Hashtags, URLs, Mentions, retweet Favorite (RT-FAV))

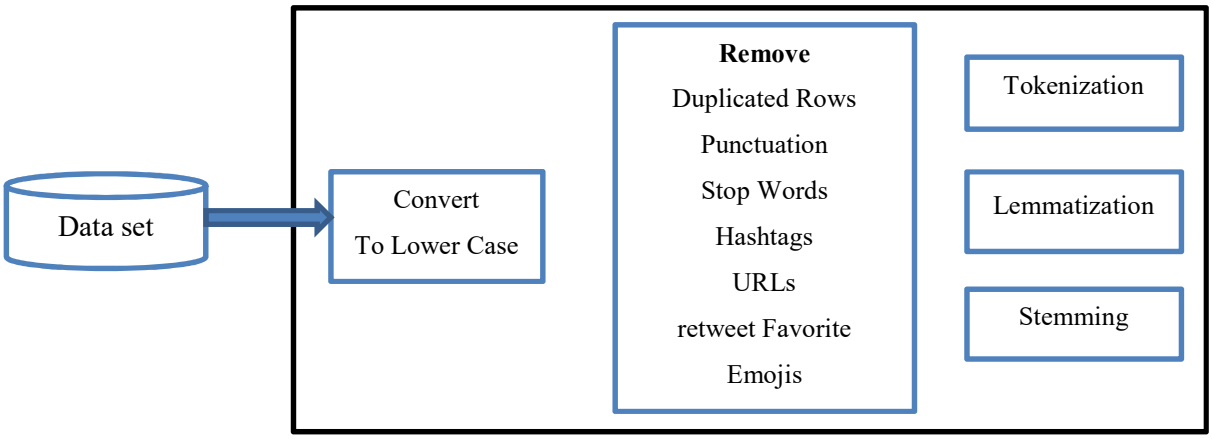

Figure 3.2: Preprocessing Step

Algorithm 3.3: Preprocessing Function

1: For each row in data set
2: Convert tweet text to corpus
3: Tokenize tweets' texts:
4: Assign each one of the following to TRUE:
remove_url
remove_punct
remove_symbols
remove_separators
5: Call tokens_tolower
6: Call Remove_stopwords
7: End for

3.4 k-shingle Approaches and Characteristic Matrix

The main objective of conducting the research is to classify similar tweets and evaluate the system according to the weight of each tweet. Therefore, the work has been prepared to improve the detection system for sentiment analysis on social media and to make the result more accurate than the rest of the systems built in different way. Before matching tweets for similarity, they should be represented as groups. This process is called shingling, and it basically constructs the tweet based on k number of words to generate what is called a "Token".

Shingles for a sentence would be a set created from some consecutive words and then stored into a dictionary. For instance, Table (3.1) shows example of one tweet and take k=5 for generate set of shingles.

Table 3.1: Generate Shingles

ID	Tweets
1	We learn some lessons from covid 19 https://t.co/8XOjefZJZ7
Shingles	
We learn some lessons from learn some lessons from covid some lessons from covid 19 lessons from covid 19 https://t.co/8XOjefZJZ7	

a. The k-shingle is k of the tokens (words) seen consecutively in a tweet. Occasionally, to slice short-length strings and use groups of hash values to convey tweets, hash shingles are useful.

b. The k words are chosen to divide the tweet into tokens.

The tweets were set to smaller subgroups that were generated from the shingles and were fragmented. The next step in the technology process is to create a characteristic matrix, which represents each group or tweet as columns while indicating the items of standardized groups as rows. Generally, the cell is given a value of 1 when the shingles is part of the group; otherwise it is given a 0.

Algorithm 34: Shingling_Function

Input: Cleaned data set

Output: Unique shingles sets across all tweets

```
Shingling_function(tweet, k)
   shingles = character( length = length(tweet) - k + 1 )
   For( i in 1:( length(tweet) - k + 1 ) )
      shingles[i] = paste( tweet [ i:(i + k - 1) ], collapse = " " )
   End for
   Return (unique(shingles) )
```

3.5 Minhash Technology

Minhash is a technique for approximating the Jaccard Similarity between two different set, making use of “random hash functions”. The aim of using Minhash is to exchange huge sets of data by smaller representations that are called signatures. The importance of using the Minhash property is that the signatures need to be compared for the two sets of data.

For more accurate results, the Minhash hash function was used. The Minhash is calculated in the tweet by specifying change rows for a column from the featured matrix described earlier. As can be seen, the quantity of Minhash for each column is equal to the number of the first row in the alternative request, where the column has a value of 1.

Consequently, the signature matrix consists of the hashes as rows and the tweets as columns. Each column refers to the Minhash signature for the tweet. It should be noticed that the signature matrix is much smaller than the characteristic matrix.

Algorithm 3.5: Signature Matrix Function

Input: Unique shingles sets across all tweets

Output: Transpose Signature Matrix bind with label (TSM-L)

```
1: Create Characteristic Matrix
2: Identify signature_num = 20
3: Find prime
4: Generate the unique coefficients coeff_a, coeff_b (randomly)
5: Call  crc32 Function and get hex hash values (hexa_value_crc32)
6: Call hex_to_int function to convert hexa_value_crc32 to int_value_crc32
7: Generate permutations
        For( i in 1:nrow(Characteristic Matrix) )
            hash =  (coeff_a[s] * as.numeric(int_value_crc32) + coeff_b[s]) %% prime
        End for
 8: Convert permutations to data frame
 9: Make the matrix as many permute and bind with the original characteristic matrix
10: Obtain the non zero rows' index for all columns of Characteristic Matrix
11: Initialize signature matrix (SM)
# for each column (Tweet)
For   i in 1:ncol(Characteristic Matrix)
      # for each hash function (signature)'s value
       For( s in 1:signature_num )
           SM [ s, i ] =  min( permute_df[, s][ non_zero_rows[[i]] ] )
       End for
End for
12: TSM=Transpose (SM)
13: TSM-L= cbind (TSM,label)
14: TSM-L$label= factor (TSM-L$label)
```

3.6 Classifier Phase

Classification is a term used both for the classification method (distinguishing and distributing kinds of" things" into various groups) and for the resulting class

collection, as well as for the assigning of items to pre-established classes. It is an interdisciplinary area of analysis. Four classification techniques have been used in the proposed method, including NB, SVM, LR and RF. Before input the data to the classifier, the dataset was separated into training and testing.

3.6.1 Support Vector Machine (SVM)

To assign them to class labels, the data points are represented as points in space separated by a distinct gap. This gap is formed by generating hyperplanes in high-dimensional space while training the data. This hyperplane is depicted in 2-D space as a line dividing the plane into two pieces. Each of these components holds the class label. Based on whatever side of the hyperplane it is on, it is mapped into space and allocated to a class. This is requiring when a new data point requires classification. The training tweets data is input to SVM algorithm to predict sentiment of tweets cases, using hyperplane for isolating the positive tweet from the others for example.

Algorithm 3.6: Support Vector Machine Function

Input: train Spark Table, test Spark Table, spark connection (x)

Output: SVM model

```
ml_linear_svc(x,formula = NULL,fit_intercept = TRUE, reg_param = 0, max_iter =
100, standardization = TRUE, weight_col = NULL, tol = 1e-06, threshold = 0,
aggregation_depth = 2, features_col = "features", label_col = "label", prediction_col
= "prediction", raw_prediction_col = "rawPrediction", uid =
random_string("linear_svc_"), ...)
```

3.6.2 Naive Bayes algorithm (NB)

The Naive-Bayes Classifier is based on Bayes' Theorem and belongs to the family of probabilistic classifiers. It presupposes that the traits are highly independent of one another.

Algorithm 3.7: Naïve Bayes Function

Input: train Spark Table, test Spark Table, spark connection (x)

Output: NB model

```
ml_naive_bayes = function(x, formula = NULL, model_type = "multinomial", smoothing
= 1, thresholds = NULL, weight_col = NULL, features_col = "features",
label_col = "label",prediction_col = "prediction", probability_col = "probability",
raw_prediction_col = "rawPrediction", uid = random_string("naive_bayes_"), ...)
        check_dots_used()
        UseMethod("ml_naive_bayes")
```

3.6.3 Logistic Regression (LR)

At the nominal, ordinal, interval, or ratio level, the regression method of LR is used to quantify the connection between a dependent variable and other independent predictor variables. A binary LR model is used when the dependent variable is binary, while a multinomial LR model is used when there are more than two categories.

Algorithm 3.8: Logistic Regression Function

Inpu: train Spark Table, test Spark Table, spark connection (x)

Output: LR model

```
ml_logistic_regression = function(x, formula = NULL, fit_intercept = TRUE,
elastic_net_param = 0, reg_param = 0, max_iter = 100, threshold = 0.5, thresholds = NULL, tol = 1e-
06, weight_col = NULL, aggregation_depth = 2, lower_bounds_on_coefficients =
NULL, lower_bounds_on_intercepts = NULL, upper_bounds_on_coefficients = NULL,
upper_bounds_on_intercepts = NULL, features_col = "features", label_col =
"label", family = "auto", prediction_col = "prediction", probability_col =
"probability", raw_prediction_col = "rawPrediction", uid =
random_string("logistic_regression_"), ...)
 check_dots_used()
 UseMethod("ml_logistic_regression")
```

3.6.4 Random Forest (RF)

RF is a classification and regression method based on ensemble learning. For classification, ensemble learning approaches incorporate more than one algorithm of the same or different type. A decision tree ensemble is created by RF. It creates a set of decision trees from a randomly selected subset of the training data. As the trees grow, the model becomes more unpredictable. Instead, then looking for the most important feature when dividing a node, it seeks for the best feature among a random sample of features.

The training tweets data is input to RF algorithm models for all cases of the tweets' groups and the output of this algorithm is multiple decision trees.

Algorithm 3.9: Random Forest Function

Input: train Spark Table, test Spark Table, spark connection (x)

Output: RF model

```
ml_random_forest = function(x, formula = NULL, type = c("classification"),
features_col = "features", label_col = "label", prediction_col = "prediction",
probability_col = "probability", raw_prediction_col = "rawPrediction",
feature_subset_strategy = "auto", impurity = "auto", checkpoint_interval = 10,
max_bins = 32, max_depth = 5, num_trees = 20, min_info_gain = 0,
min_instances_per_node = 1, subsampling_rate = 1, seed = NULL, thresholds = NULL,
cache_node_ids = FALSE, max_memory_in_mb = 256, uid = random_string("random_forest_"),
response = NULL, features = NULL)
        formula = ml_standardize_formula(formula, response, features)
        response_col = gsub("~.+$", "", formula) %>% trimws()
        sdf = spark_dataframe(x)
        schema = sdf_schema(sdf)
            if (!response_col %in% names(schema))
                stop("`", response_col, "` is not a column in the input dataset.")
            End if
      response_type = schema[[response_col]]$type
      type = rlang::arg_match(type)
            model_type = if (!identical (type, " classification"))
            "classification"
          End if
```

```
if (identical (model_type, "classification"))
        if (!impurity %in% c("gini", "entropy"))
            stop("`impurity` must be \"gini\" or \"entropy\" for classification.")
        End if
    impurity
End if
```

3.7 Evaluation Phase

It is necessary to quantify the outcomes of the algorithm while developing a machine learning model. A dataset is usually split into a dataset for training and a dataset for testing. The test dataset is used to assess the model's results.

3.8 Summary

Overall, this chapter proposed the main concepts of the sentiment classification, dealing with data that represents a human report on social media like Twitter. If the reason of the whole project is to enhance the result that classifying sentiment on media network, so different techniques must be used, which are k-shingle, Minhash and the machine learning classifier.

CHAPTER FOUR

SYSTEM IMPLEMENTATION AND RESULTS

4.1 Overview

This chapter presents a discussion of the experimental results for the proposed system, using two data sets that are label as positive, negative, and neutral. First, the data sets are preprocessed through the approaches described Chapter two, after that the data is represented in a form that is acceptable for machine learning algorithm classifiers to train the model.

A testing structure parameter of this chapter, as the implementation results of k-shingle technique will be shown onto tweet dataset. Then, the result of Minhash technique implementation on k-shingle is presented. Finally, the classifier implementation results are admitted also for all datasets utilized with the system.

To begin the implementation, the program's Spark Context must be specified. It is the main entry point for Spark functionality and must be defined before beginning to generate RDDs.

The keyword “local” denotes a local spark cluster. The number of cores determines how many worker nodes are produced. In Spark, the processing work is done by worker nodes, as explained previously.

The Local Spark mode and the YARN Client mode are the two modes of operation available. The YARN Client mode is run on the AWS server or in offline cluster, whereas the local spark mode is prepared on a laptop.

The Local spark is powered by an Intel Core i7 processor with 16 GB RAM and an NVIDIA Corporation GP107M graphics card.

The Hadoop YARN cluster has one m5.xlarge master, 4 vCore, 16 GiB memory, and 64 GiB EBS only storage. Also, the cluster has two or four m5.xlarge cores, 4 vCore, 16 GiB memory, and 64 GiB EBS only storage.

2 or 4 number of cores has been utilized in local spark mode. After that, the environment has been setup by installing the most recent versions of Java, RStudio, R, and Spark 2.4.3, as well as establishing the system's path variables.

These methods were followed for the Local Spark laptop at this time. Section 4.2 illustrates the whole installation.

The local spark mode was used to execute experiments on a Vaccine tweet dataset and a COVID-19 Tweets IEEE Data Port Dataset after successful installation and completion of other prerequisites.

4.2 Setting Up Spark Multi-Node Cluster

Table 4.1: Hardware and Software requirements.

Category	Requirements, Conventions or Software Version Used
Hardware	One server and four clients.
System	Ubuntu 16.04 in server node (krb5.wafaa.io) Ubuntu 19.10 in client nodes (client1.wafaa.io, client2.wafaa.io, client3.wafaa.io, client4.wafaa.io)
Software	Java 8 - 64 Bit Hadoop-2.7.3 Spark-2.4.3
Access	Privileged access to the Linux system as root or via the sudo command.
Internet	Connect all the nodes to the Internet.
Static IPs	All IPs of nodes are static and the hostnames as fully qualifies domain.

Configuring static IP address from Graphical User interface (GUI)

Click on the top right Network icon and Wired/Wireless settings

Click on the network configuration.

Select Manual IPv4 radio box and set desired static IP address and relevant network configuration.

Once ready hit Apply button.

Restart network

1. Edit the '/etc/hosts' file of the server and all clients by adding the following lines:

192.168.43.99 krb5.wafaa.io

192.168.43.71 client1.wafaa.io

192.168.43.194 client2.wafaa.io

192.168.43.173 client3.wafaa.io

192.168.43.101 client4.wafaa.io

2. Disable Firewall and Iptable if enabled by the following commands:

systemctl status firewalld

systemctl disable firewalld

systemctl stop firewalld
apt install iptables-services
systemctl enable iptables
systemctl start iptables
service iptables stop
service ip6tables stop

3. Install Java

sudo addgroup hadoop
sudo adduser --ingroup hadoop spuser
sudo visudo add this line:
spuser ALL=(ALL) ALL

reboot and login as spuser

sudo apt-get -y update
sudo apt-get -y upgrade
sudo apt-get update

Dowload jdk 1.8 from the link

http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

Download 64Bit java for Linux in tar.gz format (jdk-8u181-linux-x64.tar.gz) into the Desktop.

sudo mkdir /usr/local/java
sudo chmod 777 /usr/local/java
sudo cp ~/Desktop/jdk-8u181-linux-x64.tar.gz /usr/local/java
cd /usr/local/java
sudo tar xvzf jdk-8u181-linux-x64.tar.gz
sudo vim /etc/profile
Add the following lines:
JAVA_HOME=/usr/local/java/jdk1.8.0_181
JRE_HOME=$JAVA_HOME/jre
PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin
export JAVA_HOME
export JRE_HOME
export PATH

Inform the Ubuntu Linux system where the Oracle Java JDK/JRE is located:
sudo update-alternatives --install "/usr/bin/java" "java" "/usr/local/java/jdk1.8.0_181/jre/bin/java" 1
sudo update-alternatives --install "/usr/bin/javac" "javac" "/usr/local/java/jdk1.8.0_181/bin/javac" 1

Inform the Ubuntu Linux system that Oracle Java JDK/JRE must be the default Java:
sudo update-alternatives --set java /usr/local/java/jdk1.8.0_181/jre/bin/java
sudo update-alternatives --set javac /usr/local/java/jdk1.8.0_181/bin/javac
Reload the system wide PATH /etc/profile by typing the following command:
. /etc/profile

Test to see if Oracle Java was installed correctly on the system.
Run the following commands and note the version of Java
root@krb5.wafaa.io:~$ java -version
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)

4. Create password-less public ssh key:

rm -rf /home/root/.ssh
mkdir -p /root/.ssh
chmod 0700 /root/.ssh
ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa

5. Add the public key to the authorized_keys file:

cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

6. Set authorizing permission to ssh key:

chmod 600 ~/.ssh/authorized_keyscat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

7. Change the password of root user:

sudo passwd root

8. Open the configuration file sshd_config:

sudo vim /etc/ssh/sshd_config
Add the following line to it:
PermitRootLogin yes

9. Copy ssh key from each node to all nodes:

The private key should never be copied to another machine.
ssh-copy-id -i $HOME/.ssh/id_rsa.pub root@krb5.wafaa.io
ssh-copy-id -i $HOME/.ssh/id_rsa.pub root@client1.wafaa.io
ssh-copy-id -i $HOME/.ssh/id_rsa.pub root@client2.wafaa.io
ssh-copy-id -i $HOME/.ssh/id_rsa.pub root@client3.wafaa.io
ssh-copy-id -i $HOME/.ssh/id_rsa.pub root@client4.wafaa.io
NOTE:
If ssh-copy-id command is not be installed follow step 10 and 11.

10. First create .ssh directory on server

ssh spuser@krb5.wafaa.io "umask 077; test -d .ssh || mkdir .ssh"
ssh root@client1.wafaa.io "umask 077; test -d .ssh || mkdir .ssh"
ssh root@client2.wafaa.io "umask 077; test -d .ssh || mkdir .ssh"

ssh spuser@client3.wafaa.io "umask 077; test -d .ssh || mkdir .ssh"
ssh root@client4.wafaa.io "umask 077; test -d .ssh || mkdir .ssh"

11. cat local id.rsa.pub file and pipe over ssh to append the public key in remote server:

cat $HOME/.ssh/id_rsa.pub | ssh spuser@krb5.wafaa.io "cat >> .ssh/authorized_keys"
cat $HOME/.ssh/id_rsa.pub | ssh root@client1.wafaa.io "cat >> .ssh/authorized_keys"
cat $HOME/.ssh/id_rsa.pub | ssh root@client2.wafaa.io "cat >> .ssh/authorized_keys"
cat $HOME/.ssh/id_rsa.pub | ssh spuser@client3.wafaa.io "cat >> .ssh/authorized_keys"
cat $HOME/.ssh/id_rsa.pub | ssh root@client4.wafaa.io "cat >> .ssh/authorized_keys"

12. copy files to external USB pen/hard drive:

cp -avr $HOME/.ssh/ /mnt/usb/backups/

13. Test the connection of server with all nodes:

ssh krb5.wafaa.io
ssh client1.wafaa.io
ssh client2.wafaa.io
ssh client3.wafaa.io
ssh client4.wafaa.io

14. Disable IPV6

Hadoop and IPV6 do not agree on the meaning of 0.0.0.0 address, thus it is advisable to disable IPV6 by editing sysctl.conf:

sudo vim /etc/sysctl.conf
# disable ipv6
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1

Check if IPv6 is disabled:

After a system reboot the output of

cat /proc/sys/net/ipv6/conf/all/disable_ipv6

should be 1, meaning that IPV6 is actually disabled.

15. Hadoop Installation

Download hadoop-2.7.3.tar.gz
Save it to spuser/Desktop.
Move the above downloaded file to /usr/local/

sudo mv ~/Desktop/hadoop-2.7.3.tar.gz /usr/local
cd /usr/local
sudo tar -xvf hadoop-2.7.3.tar.gz
sudo rm hadoop-2.7.3.tar.gz
sudo ln -s hadoop-2.7.3 hadoop

```
sudo chown -R spuser:hadoop hadoop-2.7.3
sudo chmod 777 hadoop-2.7.3
```

Add the following to /usr/local/hadoop/etc/hadoop/hadoop-env.sh

```
sudo vim /usr/local/hadoop/etc/hadoop/hadoop-env.sh
export HADOOP_OPTS=-Djava.net.preferIPv4Stack=true
export HADOOP_HOME_WARN_SUPPRESS="TRUE"
export JAVA_HOME=/usr/local/java/jdk1.8.0_181
```

also, remove the following line:

```
export JAVA_HOME=${JAVA_HOME}
```

Add the following lines to the end of the $HOME/.bashrc file of user spuser.

```
vim ~/.bashrc
# Set Hadoop-related environment variables
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_MAPRED_HOME=${HADOOP_HOME}
export HADOOP_COMMON_HOME=${HADOOP_HOME}
export HADOOP_HDFS_HOME=${HADOOP_HOME}
export HADOOP_YARN_HOME=${HADOOP_HOME}
export HADOOP_CONF_DIR=${HADOOP_HOME}/etc/hadoop
# Native Path
export
HADOOP_COMMON_LIB_NATIVE_DIR=${HADOOP_PREFIX}/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_PREFIX/lib"
# Set JAVA_HOME (we will also configure JAVA_HOME directly for Hadoop
later on)
export JAVA_HOME=/usr/local/java/jdk1.8.0_181
# Some convenient aliases and functions for running Hadoop-related commands
unaliasfs&> /dev/null
aliasfs="hadoop fs"
unaliashls&> /dev/null
aliashls="fs -ls"
export
PATH=$PATH:$HADOOP_HOME/bin:$PATH:$JAVA_HOME/bin:$HADOO
P_HOME/sbin
```

```
source ~/.bashrc
```

```
sudo gedit /usr/local/hadoop/etc/hadoop/slaves
```

Add the following lines:

```
krb5.wafaa.io
```

client1.wafaa.io
client2.wafaa.io
client3.wafaa.io
client4.wafaa.io
Create masters file and edit as follows:
sudo gedit /usr/local/hadoop/etc/hadoop/masters
Add this line:
krb5.wafaa.io

Add the following snippets in yarn-site.xml and core-site.xml between the <configuration> ... </configuration> tags.

```
sudo  vim /usr/local/hadoop/etc/hadoop/yarn-site.xml
<property>
      <name>yarn.nodemanager.aux-services</name>
       <value>mapreduce_shuffle</value>
 </property>
 <property>
     <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
     <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
<property>
   <name>yarn.nodemanager.pmem-check-enabled</name>
   <value>false</value>
</property>
<property>
   <name>yarn.nodemanager.vmem-check-enabled</name>
   <value>false</value>
</property>
<property>
    <name>yarn.resourcemanager.resource-tracker.address</name>
    <value>krb5.wafaa.io:8025</value>
</property>
<property>
    <name>yarn.resourcemanager.scheduler.address</name>
  <value>krb5.wafaa.io:8030</value>
</property>
<property>
    <name>yarn.resourcemanager.address</name>
  <value>krb5.wafaa.io:8050</value>
</property>

sudo vim /usr/local/hadoop/etc/hadoop/core-site.xml
<property>
```

```
    <name>hadoop.tmp.dir</name>
    <value>/app/hadoop/tmp</value>
    <description>A base for other temporary directories.</description>
      </property>
 <property>
    <name>fs.default.name</name>
    <value>hdfs://krb5.wafaa.io:9000</value>
    <description>default host and port</description>
</property>
```

Create the above temp folder and give appropriate permission:

```
sudo mkdir -p /app/hadoop/tmp
sudo chown yarn:hadoop -R /app/hadoop/tmp
sudo chmod 750 /app/hadoop/tmp
sudo vim  /usr/local/hadoop/etc/hadoop/mapred-site.xml
 <property>
       <name>mapreduce.framework.name</name>
       <value>yarn</value>
  </property>
  <property>
       <name>mapreduce.jobhistory.address</name>
       <value>krb5.wafaa.io:10020</value>
       <description>Host and port for Job History Server (default
0.0.0.0:10020)</description>
</property>
```

Create a temporary directory which will be used as base location for DFS (in server host):

```
 sudo rm  -R /usr/local/hadoop_tmp/hdfs/datanode
 sudo rm  -R /usr/local/hadoop_tmp/hdfs/namenode
 sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
 sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanodee
 sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode
 sudo chown  spuser:hadoop -R /usr/local/hadoop_tmp/
 sudo chmod 777 /usr/local/hadoop_tmp/hdfs/namenode
 sudo chmod 777 /usr/local/hadoop_tmp/hdfs/datanodee
 sudo chmod 777 /usr/local/hadoop_tmp/hdfs/datanode
sudo /usr/local/spark/bin/spark-submit --deploy-mode cluster --class
org.apache.spark.examples.SparkPi /usr/local/spark/examples/jars/spark-
examples_2.11-2.4.3.jar 10
```

Create a temporary directory which will be used as base location for DFS (in client host):

```
sudo rm  -R /usr/local/hadoop_tmp/hdfs/datanode
sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode
sudo chown spuser:hadoop  -R /usr/local/hadoop_tmp/
sudo chmod 777 /usr/local/hadoop_tmp/hdfs/datanode
```

Edit hdfs-site.xml on server krb5.wafaa.io (the server in our cluster will work as server and client)

```
sudo vim /usr/local/hadoop/etc/hadoop/hdfs-site.xml
 <property>
      <name>dfs.replication</name>
      <value>5</value>
 </property>
 <property>
      <name>dfs.namenode.name.dir</name>
      <value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
 </property>
 <property>
     <name>dfs.datanode.data.dir</name>
     <value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
 </property>
```

Edit hdfs-site.xml on server client (1,2,3,4).wafaa.io (4 client hosts)

```
sudo vim /usr/local/hadoop/etc/hadoop/hdfs-site.xml
 <property>
      <name>dfs.replication</name>
      <value>5</value>
 </property>
 <property>
     <name>dfs.datanode.data.dir</name>
     <value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
 </property>
```

7. Installing Scala

Go to /usr/local/ where Java and Hadoop installed.

```
cd /usr/local
sudo wget https://downloads.lightbend.com/scala/2.11.12/scala-2.11.12.tgz
sudo tar -xzf scala-2.11.12.tgz
sudo ln -s scala-2.11.12 scala
```

8. Installing Spark

```
sudo wget https://www.apache.org/dyn/closer.lua/spark/spark-2.4.3/spark-2.4.3-bin-hadoop2.7.tgz
sudo tar -xzf spark-2.4.3-bin-hadoop2.7.tgz
sudo ln -s spark-2.4.3-bin-hadoop2.7 spark
```

sudo vim ~/.bashrc
add the following four lines:
export SCALA_HOME=/usr/local/scala
export PATH=$PATH:$SCALA_HOME/bin
export SPARK_HOME=/usr/local/spark
export PATH=$PATH:$SPARK_HOME/bin

source ~/.bashrc
scala -version

sudo vim /usr/local/spark/conf/spark-env.sh
add the following lines:
export SCALA_HOME=/usr/local/scala
export JAVA_HOME=/usr/local/java/jdk1.8.0_181
export SPARK_HOME=/usr/local/spark
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
export SPARK_WORKER_DIR=/usr/local/spark/work
export SPARK_LOG_DIR=/usr/local/spark/log
export SPARK_MASTER_IP=krb5.wafaa.io

Edit the file slaves in the same directory by adding all clients:
sudo vim /usr/local/spark/conf/slaves
krb5.wafaa.io
client1.wafaa.io
client2.wafaa.io
client3.wafaa.io
client4.wafaa.io
NOTE: In the rest of the configuration.

In the client machine, we have to change hostname from krb5.wafaa.io to client1.wafaa.io, client2.wafaa.io, client3.wafaa.io, or client4.wafaa.io.
5 Add the following properties to the mapred-site.xml file:

```
<!-- TaskController settings -->
<property>
  <name>mapred.task.tracker.task-controller</name>
  <value>org.apache.hadoop.mapred.LinuxTaskController</value>
</property>
<property>
  <name>mapreduce.tasktracker.group</name>
  <value>hadoop</value>
</property>
```

2) Find out which process is listening on port 20:
sudo ss -tulpn | grep :20
stop the process
3) Edit the file /etc/ssh/sshd_config by adding the line:
PermitRootLogin yes
4) Format the namenode and start Hadoop
5) Start master and slaves in Spark

The complete /usr/local/hadoop-2.7.3/etc/hadoop/core-site.xml file (In the server machine krb5.wafaa.io and in all four clients):

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl"
<configuration>
<property>
        <name>hadoop.tmp.dir</name>
        <value>/app/hadoop/tmp</value>
        <description>A base for other temporary directories.</description>
</property>
<property>
        <name>fs.default.name</name>
        <value>hdfs://krb5.wafaa.io:9001</value>
        <description>default host and port</description>
</property>
<property>
  <name>dfs.namenode.https-address</name>
  <value>0.0.0.0:9871</value>
</property>
<property>
  <name>dfs.namenode.name.dir</name>
  <value>file:/usr/local/hadoop-2.7.3/data/name</value>
</property>
<property>
  <name>dfs.dataname.data.dir</name>
  <value>file:/usr/local/hadoop-2.7.3/data/data</value>
</property>
<property>
  <name>hadoop.tmp.dir</name>
  <value>file:/usr/local/hadoop-2.7.3/data/tmp</value>
</property>
</configuration>
```

The complete /usr/local/hadoop-2.7.3/etc/hadoop/hadoop-env.sh file (In the server machine krb5.wafaa.io and in all four clients):

```
# The java implementation to use.
export JAVA_HOME=${JAVA_HOME}
#export JSVC_HOME=${JSVC_HOME}
export HADOOP_CONF_DIR=${HADOOP_CONF_DIR:-"/etc/hadoop"}
# Extra Java CLASSPATH elements.  Automatically insert capacity-scheduler.
for f in $HADOOP_HOME/contrib/capacity-scheduler/*.jar; do
  if [ "$HADOOP_CLASSPATH" ]; then
    export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
  else
    export HADOOP_CLASSPATH=$f
  fi
done
# The maximum amount of heap to use, in MB. Default is 1000.
#export HADOOP_HEAPSIZE=
#export HADOOP_NAMENODE_INIT_HEAPSIZE=""
# Extra Java runtime options.  Empty by default.
export HADOOP_OPTS="$HADOOP_OPTS -Djava.net.preferIPv4Stack=true"
# A string representing this instance of hadoop. $USER by default.
export HADOOP_IDENT_STRING=$USER
export HADOOP_OPTS=-Djava.net.preferIPv4Stack=true
export HADOOP_HOME_WARN_SUPPRESS="TRUE"
export JAVA_HOME=/usr/local/java/jdk1.8.0_181
```

The complete /usr/local/hadoop-2.7.3/etc/hadoop/hdfs-site.xml file (In the server machine krb5.wafaa.io).

NOTE: In all clieents, change the hostname krb5.wafaa.io to client1.wafaa.io, client2.wafaa.io, client3.wafaa.io, or client4.wafaa.io

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
   <name>dfs.replication</name>
   <value>5</value>
</property>
<property>
   <name>dfs.namenode.name.dir</name>
   <value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
</property>
<property>
   <name>dfs.datanode.data.dir</name>
   <value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
</property>
<property>
```

```
  <name>dfs.datanode.address</name>
  <value>0.0.0.0:50010</value>
</property>
<property>
  <name>dfs.datanode.http.address</name>
  <value>0.0.0.0:50075</value>
</property>
</configuration>
```

The complete /usr/local/hadoop-2.7.3/etc/hadoop/slaves file (In the server machine krb5.wafaa.io and in all four clients):

```
krb5.wafaa.io
client1.wafaa.io
client2.wafaa.io
client3.wafaa.io
client4.wafaa.io
```

The complete /usr/local/hadoop-2.7.3/etc/hadoop/masters file (In the server machine krb5.wafaa.io and in all four clients):

```
krb5.wafaa.io
```

The complete /usr/local/hadoop-2.7.3/etc/hadoop/mapred-site.xml file (In the server machine krb5.wafaa.io and in all four clients):

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!-- Put site-specific property overrides in this file. -->
<configuration>
<property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
</property>
<property>
   <name>mapreduce.jobhistory.address</name>
   <value>krb5.wafaa.io:10020</value>
   <description>Host and port for Job History Server (default
                0.0.0.0:10020)</description>
</property>
<!-- TaskController settings -->
<property>
  <name>mapred.task.tracker.task-controller</name>
  <value>org.apache.hadoop.mapred.LinuxTaskController</value>
</property>
<property>
  <name>mapreduce.tasktracker.group</name>
```

```
  <value>hadoop</value>
</property>
<property>
  <name>mapreduce.jobhistory.principal</name>
  <value>mapred/krb5.wafaa.io@WAFAA.IO</value>
</property>
</configuration>
```

The complete /usr/local/hadoop-2.7.3/etc/hadoop/yarn-site.xml file (In the server machine krb5.wafaa.io).

NOTE: In all clieents, change the hostname in krb5.wafaa.io@WAFAA.IO to client1.wafaa.io, client2.wafaa.io, client3.wafaa.io or client4.wafaa.io.

```
<?xml version="1.0"?>
<configuration>
<!-- Site specific YARN configuration properties -->
<property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
</property>
<property>
   <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
   <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
<property>
   <name>yarn.nodemanager.pmem-check-enabled</name>
   <value>false</value>
</property>
<property>
   <name>yarn.nodemanager.vmem-check-enabled</name>
   <value>false</value>
</property>
<property>
    <name>yarn.resourcemanager.resource-tracker.address</name>
  <value>krb5.wafaa.io:8025</value>
</property>
<property>
    <name>yarn.resourcemanager.scheduler.address</name>
  <value>krb5.wafaa.io:8030</value>
</property>
<property>
    <name>yarn.resourcemanager.address</name>
  <value>krb5.wafaa.io:8050</value>
</property>
```

```
<property>
  <name>yarn.nodemanager.linux-container-executor.group</name>
  <value>hadoop</value>
</property>
<property>
  <name>yarn.nodemanager.linux-container-executor.path</name>
  <value>/usr/local/hadoop-2.7.3/bin/container-executor</value>
</property>
<property>
  <name>yarn.nodemanager.webapp.https.address</name>
  <value>0.0.0.0:8044</value>
</property>
</configuration>
```

The complete /usr/local/spark/conf/spark-env.sh file (In the server machine krb5.wafaa.io and in all four clients):

```
export SCALA_HOME=/usr/local/scala
export JAVA_HOME=/usr/local/java/jdk1.8.0_181
export SPARK_HOME=/usr/local/spark
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
export SPARK_WORKER_DIR=/usr/local/spark/work
export SPARK_LOG_DIR=/usr/local/spark/log
export SPARK_MASTER_IP=krb5.wafaa.io
export PYSPARK_PYTHON=/usr/bin/python3.7
export PYSPARK_DRIVER_PYTHON=jupyter
export PYSPARK_DRIVER_PYTHON=/home/spuser/anaconda3/bin/jupyter
# Setup env variable for Jupyter + Pyspark
export PYSPARK_DRIVER_PYTHON_OPTS="notebook --no-browser"
```

The complete /usr/local/spark/conf/slaves file (In the server machine krb5.wafaa.io and in all four clients):

```
rb5.wafaa.io
client1.wafaa.io
client2.wafaa.io
client3.wafaa.io
client4.wafaa.io
```

4.3 Dataset Description

Several tests were conducted on a set of data to distinguish sentiment and to understand the accuracy of the proposed strategy, and are explained as follows:

4.3.1 COVID-19 Vaccine Stance Dataset

In short, dataset is CSV file; with a content of plain texts represent tweets, classes, and the ID of each tweet. Each line of the content has a unique ID to refer to it. Table (4.1) shows the statistics of the used dataset.

Between November 9 and December 8, 2020, of tweets concerning the topic of COVID-19 vaccination. Only tweets written in English have been considered. Duplicated tweets have been discarded, as well as retweet. The remaining number of tweets in the dataset is 2790.

The stance of the tweets towards vaccination has been evaluated by three independent human raters into three classes: in favor, against and neutral. Tweets that have been assigned to the class in favor express a positive opinion regarding the vaccination. Tweets belonging to the against vaccination class express a negative opinion towards COVID-19 vaccination. The neutral class mainly includes news related to the development of vaccines, tweets that do not express a clear opinion, such as questions regarding the vaccine, informative tweets concerning vaccination, as well as off-topic tweets. The used dataset is available at the following link: https://github.com/liviucotfas/covid-19-vaccination-stance-detection.

The authorization to view and print the tweets' text has gotten. OAuth is an open protocol to permit secure authorization in a straightforward and standard strategy from web, mobile and desktop applications. This system allows users to allow their consent to act on their sake without sharing the account password. After the client has given permission, OAuth will return a token, and this token itself awards get form demands on sake of the client. However, some tweets' texts have

removed by the users who wrote the tweets after sometimes. Also, some of tweets are not authorized by normal authenticated user. This is the reason of getting 2790 instead of 7530 tweets.

Table 4.2: Statistics for the used COVID-19 Vaccine Dataset

Class	Number	Precent
Against	789	28.27957
Neutral	1009	36.16487
In favor	992	35.55556
Total	2790	100%

Table 4.3: Snapshot of Snapshot of Sample of COVID-19 Vaccine Stance Dataset.

Tweet_Id	Tweet_text	Label
1	Imagine if the new covid19 vaccine has a side effect of making people infertile? #DepopulationAgenda	-1
2	Effective at WHAT? Is it full of poison and heavy metals like ALL the other â€œvaccinesâ€ you make? No thanks. Prettyâ€¦ https://t.co/xWm74DeLgB	0
3	I knew the opposition's shrieking over the vaccine was just another manufactured scandal, meant to distract. https://t.co/Oybiis4qST	1
4	@itvnews The #coronavirus "vaccine" has been approved by Nazi butcher Josef Mengele, and the UK's press are pushingâ€¦ https://t.co/DBXt6ObhEX	0
5	Q&A: Where are we in the COVID-19 vaccineÂ race? https://t.co/s12BAAIuMB	1
6	Hospitals are reportedly being prepped for a possible coronavirus vaccine roll out within three weeks.â€¦ https://t.co/qLVrd9kaMT	1
7	GBP/USD pushing against 1.3200 amid upbeat COVID-19 vaccine reports https://t.co/GKWcVY2BLA #GBPUSD #Currencies	1

The Snapshot of Sample of the dataset as in Table (4.3) which shows only seven tweets from 2790 COVID-19 Vaccine Stance Dataset tweets:

4.3.2 COVID-19 Tweets IEEE Data Port Dataset

The dataset is obtained from the IEEE data port on May 31, 2020. It contains the tweet IDs and sentiment scores of tweets. Tweets IDs are extracted using the following filters: language “en” and keywords “corona”, “coronavirus”, “covid”,

"pandemic" and variants such as "sarscov2", "nCov", "covid-19", "ncov2019", "2019ncov" and their hashtags. IEEE data port does not provide tweet text, resorting us to develop an in-house crawler that could extract them from Tweeter based on the tweet IDs. The sentiment scores are classified as positive (greater than 0), neutral (equal to 0) and negative (less than 0).

Some tweets' texts have removed by the users who wrote the tweets after sometimes. Also, some of tweets are not authorized by normal authenticated user. This is the reason of getting 20765 tweets only.

In Table (4.4) the statistics of all classes of this dataset.

Table 4.4: Counts of tweets within the three classes.

Sentiment	Count
Neutral	7064
Positive	7504
Negative	6197
Total	20765

4.4 Clean Dataset Results for COVID-19 Vaccine Stance Dataset

The first stage of the initial processing process is cleaning the data by removing all non-alphanumeric characters, reducing word inflections with lowercase, Twitter Mentions, URLs and Space Patter, as shown in Table (4.5). The same seven tweets which are shown in Table (4.3) have been taken in the rest steps as illustrated in Table (4.5), Table (4.6), Table (4.7) and Table (4.8).

Table 4.5: ***Sample of Cleaned Dataset***

Tweets
[1] "imagin new covid19 vaccine side effect make people infertil depopulationagenda"
[2] " effect full poison heavi metal like vaccine make thank pretti "
[3] "knew opposite shriek vaccine just another manufacture scandal meant distract"
[4] "vaccine approv trump penc take chance dead stay alive thanx"

[5] “q amp covid vaccine race”
[6] “hospit report prep possible coronavirus vaccine roll within three week”
[7] “push amid upbeat covid vaccine report gbpusd”

4.5 k-shingle Results for COVID-19 Vaccine Stance Dataset

In this section of testing, the k-shingle approach is implemented as shown in Chapter Three.

The next step is finding the Characteristic Matrix for tweets based on hashing of k-shingles then finding Minhash based on the proposed approach referred to in the previous chapter.

Table 4.6: Sample of k-shingles result when k=5

Shingles
imagin new covid19 vaccine side
new covid19 vaccine side effect
covid19 vaccine side effect make
vaccine side effect make people
side effect make people infertile
effect make people infertil depopulationagenda
effect full poison heavi metal
full poison heavi metal like
poison heavi metal like vaccine
heavi metal like vaccine make
metal like vaccine make thank
like vaccine make thank pretty
knew opposite shriek vaccine just
opposite shriek vaccine just another
shriek vaccine just another manufacture
Vaccine just another manufacture scandal
just another manufacture scandal meant
another manufacturer scandal meant distract
vaccine approv trump penc take
approv trump penc take chance
trump penc take chance dead
penc take chance dead stay
take chance dead stay alive
chance dead stay alive thanx

q amp covid vaccine race
hospit report prep possible coronavirus
report prep possible coronavirus vaccine
prep possible coronavirus vaccine roll
possible coronavirus vaccine roll within
coronavirus vaccine roll within three
vaccine roll within three week
push amid upbeat covid vaccine
amid upbeat covid vaccine report
upbeat covid vaccine report gbpusd
covid vaccine report gbpusd currenc

4.6 k-shingle and Minhash Results for COVID-19 Vaccine Stance Dataset

As it has been proposed earlier, the characteristic matrix is huge and needs to be made dense before using it. Therefore, the Minhash technique is used to generate signature matrix as proposed in Chapter Three. Table (4.7) and Table (4.8) explain the result of implementing hash function-based k-shingle to characteristic matrix.

*Table 4.7: Sample of **Characteristic Matrix***

Shingles (k=5)	tweet_1	tweet_2	tweet_3	tweet_4	tweet_5	tweet_6	tweet_7
imagin new covid19 vaccine side	1	0	0	0	0	0	0
new covid19 vaccine side effect	1	0	0	0	0	0	0
covid19 vaccine side effect make	1	0	0	0	0	0	0
vaccine side effect make people	1	0	0	0	0	0	0
side effect make people infertile	1	0	0	0	0	0	0
effect make people infertil depopulationagenda	1	0	0	0	0	0	0
effect full poison heavi metal	0	1	0	0	0	0	0
full poison heavi metal like	0	1	0	0	0	0	0
poison heavi metal like vaccine	0	1	0	0	0	0	0
heavi metal like vaccine make	0	1	0	0	0	0	0
metal like vaccine make thank	0	1	0	0	0	0	0
like vaccine make thank pretty	0	1	0	0	0	0	0
knew opposite shriek vaccine just	0	0	1	0	0	0	0
opposite shriek vaccine just another	0	0	1	0	0	0	0
shriek vaccine just another manufacture	0	0	1	0	0	0	0
Vaccine just another manufacture scandal	0	0	1	0	0	0	0
just another manufacture scandal meant	0	0	1	0	0	0	0

another manufacturer scandal meant distract	0	0	1	0	0	0	0
vaccine approv trump penc take	0	0	0	1	0	0	0
approv trump penc take chance	0	0	0	1	0	0	0
trump penc take chance dead	0	0	0	1	0	0	0
penc take chance dead stay	0	0	0	1	0	0	0
take chance dead stay alive	0	0	0	1	0	0	0
chance dead stay alive thanx	0	0	0	1	0	0	0
q amp covid vaccine race	0	0	0	0	1	0	0
hospit report prep possible coronavirus	0	0	0	0	0	1	0
report prep possible coronavirus vaccine	0	0	0	0	0	1	0
prep possible coronavirus vaccine roll	0	0	0	0	0	1	0
possible coronavirus vaccine roll within	0	0	0	0	0	1	0
coronavirus vaccine roll within three	0	0	0	0	0	1	0
vaccine roll within three week	0	0	0	0	0	1	0
push amid upbeat covid vaccine	0	0	0	0	0	0	1
amid upbeat covid vaccine report	0	0	0	0	0	0	1
upbeat covid vaccine report gbpusd	0	0	0	0	0	0	1
covid vaccine report gbpusd currenc	0	0	0	0	0	0	1

Table 4.8: Sample of Signature Matrix

	tweet_1	tweet_2	tweet_3	tweet_4	tweet_5	tweet_6	tweet_7
minhash_1	418	13	27	107	284	332	539
minhash_2	28	23	329	469	75	159	99
minhash_3	258	16	301	321	506	518	429
minhash_4	508	24	31	71	441	465	287
minhash_5	374	43	370	67	220	376	345
minhash_6	134	40	388	205	342	7	5
minhash_7	442	32	352	9	355	487	71
minhash_8	226	53	394	171	501	142	318
minhash_9	50	40	89	369	144	312	192
minhash_10	170	7	299	359	351	387	120
minhash_11	455	40	103	463	415	68	155
minhash_12	97	13	312	412	211	271	389
minhash_13	513	24	337	517	493	38	363
minhash_14	541	47	103	423	5	197	462
minhash_15	345	19	40	160	144	216	245
minhash_16	300	48	382	119	79	259	50
minhash_17	129	40	82	322	290	434	492
minhash_18	196	28	63	263	424	544	217
minhash_19	389	53	123	523	282	522	431
minhash_20	274	27	55	215	6	102	516

Hence, from the characteristic matrix, a signature matrix is generated, and its size is less than the characteristic matrix. The matrix's rows refer to the Minhash number while its columns refer to the tweets. The results show an example of the signature matrix using numbers of hash functions. The signature matrix has been transposed before entered to machine learning. The next step is performing the machine learning to classify the tweets as positive, negative, or neutral tweet.

4.7 Model Evaluation

Two modes have been used local spark, and in YARN client and in R. To run the job in local mode, there are some specific properties of this mode as below:

This is designed to test code with a small quantity of data in a local environment. It does not offer the benefits of a distributed environment.

The number of CPU cores to be assigned to the local operation is either 2 or 4. By using breakpoints while running the code from Rstudio, it aids in debugging.

The driver program runs on the YARN client. The command to submit the spark application may not be written in a machine in the YARN cluster. The tasks are completed on the executors in the YARN cluster's node managers in this manner, even though the drive program is executing on the client workstation. When using a cluster, the Core and RAM of the cluster is used to perform computations (done by the workers). However, the driver program can be run on the cluster or in the machine.

It has been used the Amazon EMR to create the Apache Spark cluster. There are many features of Amazon EMR for Apache spark as below:

On Amazon EMR clusters, a performance-optimized for Apache Spark runtime environment is enabled by default.

Amazon EMR security features can be considered as one reason of choosing it. Customers in regulated areas, such as healthcare, for example, select Amazon EMR as part of their data strategy. They do so to meet stringent regulatory criteria imposed by organizations like the Health Insurance Portability and

Accountability Act (HIPAA). Section 4.7.1 explains the implementation of sparklyr with an Apache Spark cluster in Amazon Web Services.

4.7.1 Implementation of sparklyr with an Apache Spark cluster in Amazon Web Services:

A cluster is built up with four core nodes and one master node with ease using Elastic Map Reduce (EMR). The Elastic Compute Cloud (EC2) provides virtual servers for nodes.

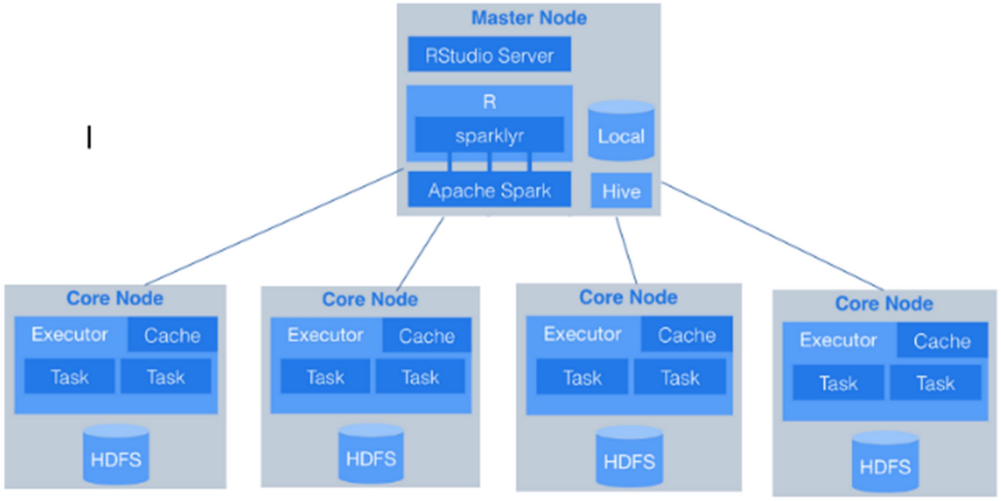

Figure 4.1: A cluster with four core nodes and one master node

Build an EMR cluster

It must be first built the following on AWS before commencing the EMR wizard setup:

To SSH into the EC2 master node, we will need an AWS key pair (.pem key).

A security group that allows remote access to IP port 22 and port 8787.

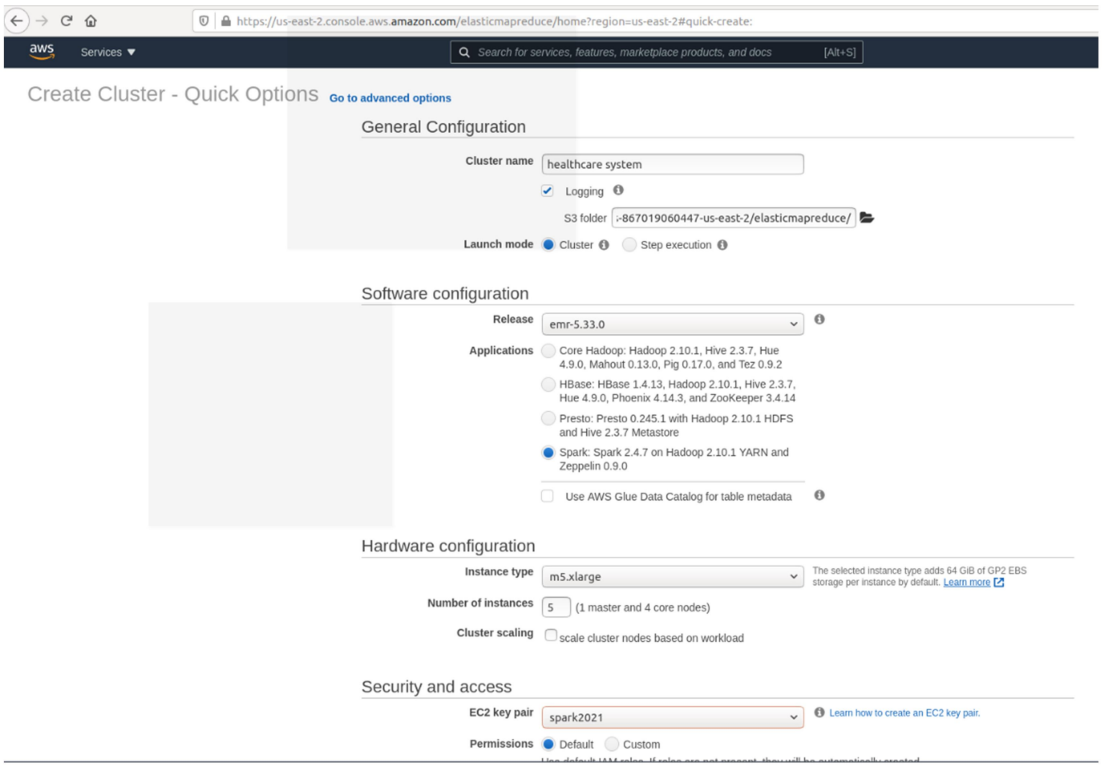

Figure 4.2: Create EMR cluster

Step 1: Select software

As part of the installation, be sure to select Hive and Spark. Note that by selecting Spark, R will be installed as part of the installation on the master node.

Step 2: Select hardware

Install 4 core nodes and 1 master node, each with 64 GB of m3.xlarge storage. Later, we can easily expand the number of nodes.

Select an EC2 instance type that corresponds to the size of the data and the amount of processing required for the analysis. R operates on a single core node by default, and it uses a lot of RAM in many circumstances. The general-purpose T2 instance types are adequate and inexpensive for programming and development, and t2.micro is offered under the AWS Free Tier. While not exhaustive, the following table contains a collection of instance kinds.

.

Table 4.9: A list of instance types

Instance family/type	Suggested use case	Multi-user?
T2.micro **T2.medium** T2.large	Free and simple testing Daily development Analysis with small data sets	No No No
M4.large **M4.2xlarge** M4.10xlarge	Daily development and analyses Parallelized analyses Parallelized analyses and huge data sets	No Yes Yes
G2.2xlarge G2.8xlarge	R-gpu development R-gpu big analyses	No Yes

It can be altered the instance type: simply stop the instance, change the instance type, and restart it.

Step 3: Select general cluster settings.

Click next on the general cluster settings.

Step 4: Select security

Enter the EC2 security group and key pair. Ports 22 and 8787 should be open in the security group.

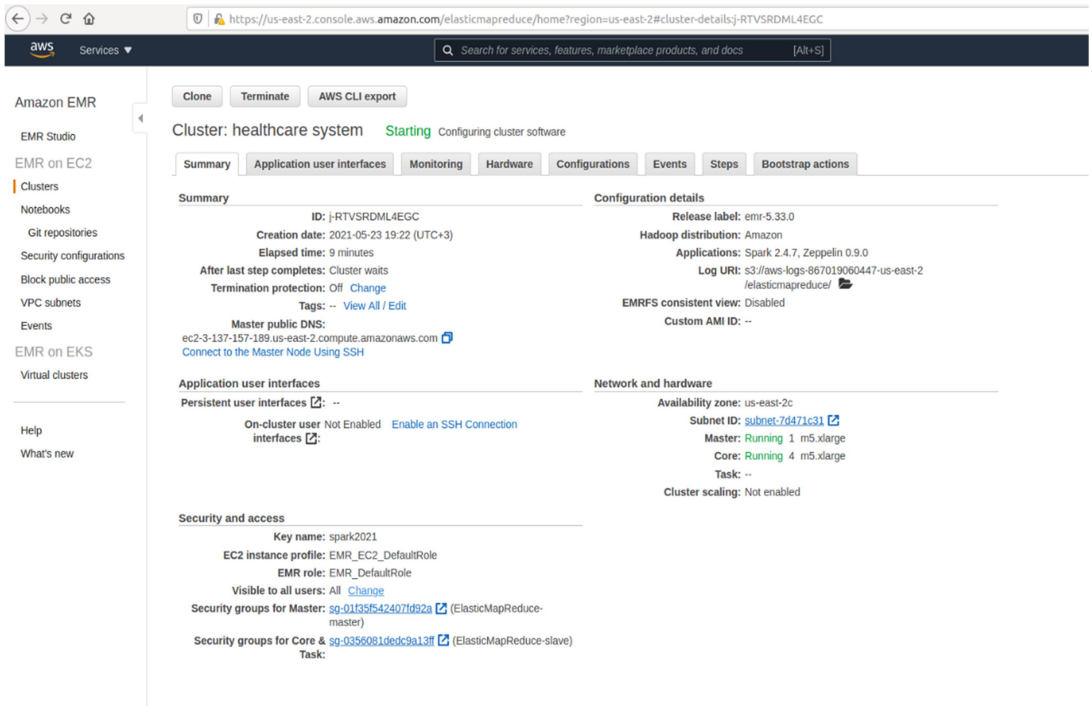

Figure 4.3: Starting EMR cluster

Connect to EMR

The cluster page will include information on the EMR cluster as well as instructions on how to connect to it.

ssh -i spark2021new.pem

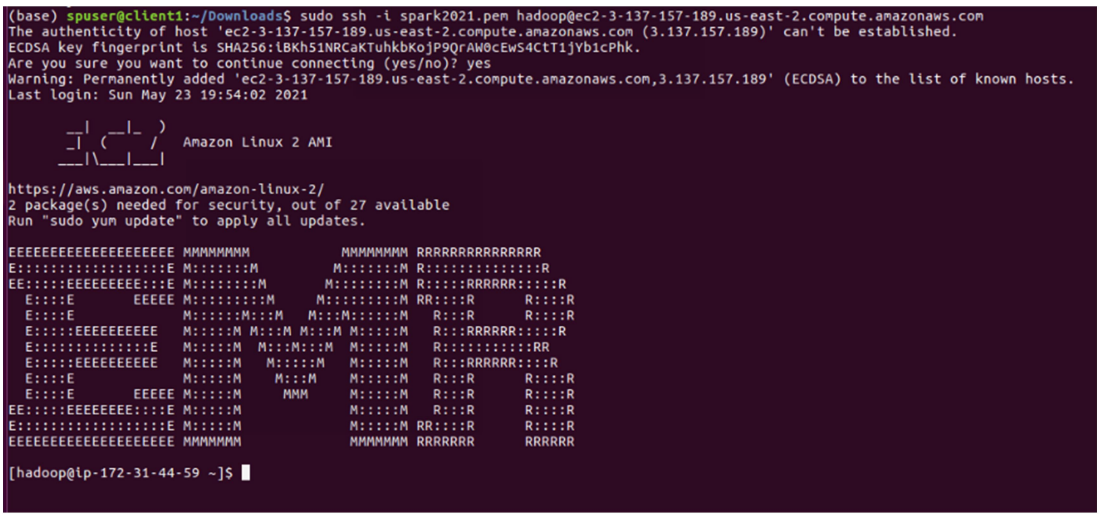

Figure 4.4: Connect to EMR

Install RStudio Server

Rstudio is a popular IDE for working with R that is available commercially or under the AGPLv3 license. If we do not want to connect to a server via SSH and instead use code editors like vim to do analytics, this is excellent. The desktop version of RStudio is available, as well as a server version that lets you use R through a web browser.

We might want to visualize the results after we have studied them. Shiny is a fantastic R tool for creating interactive dashboards that can be licensed commercially or under the AGPLv3 license.

Shiny is an R-based web application framework. It converts the studies into interactive online applications without requiring any HTML, CSS, or JavaScript skills. Customers can access the R visualization through a web browser, and Shiny Server can run R tasks in the background, including database queries.

To launch the server on Amazon Web Services (AWS), it is referred to as an EC2 instance.

The first step is to set up an Amazon Elastic Compute Cloud instance. We need to select an Amazon Machine Image (AMI), which contains all the information needed to launch an instance. An AMI, for instance, specifies the operating system and software loaded on the EC2 machine.

We can use the Amazon Linux AMI, which is free and offers a stable version of R in its repository. AWS maintains this AMI, which contains packages and configurations that enable native AWS and other software interaction.

We can share the R-based analysis server with a group of scientists using RStudio Server. Each scientist should have their own Linux user account, and multiple scientists can work on the same workstation. At least one CPU and RAM are required for each user. Use a m4.2xlarge instance type for multiuser activities at the very least.

Amazon Linux, which is built on Centos, is used by EMR. Install dependencies that will be utilized by R packages on the master node.

```
sudo yum update
sudo yum install libcurl-devel openssl-devel
```

To install R, RStudio Server, the Shiny package, and Shiny Server, run the following script:

```
sudo yum install -y R
sudo wget https://download2.rstudio.org/rstudio-server-rhel-1.0.153-x86_64.rpm
sudo wget https://download2.rstudio.org/server/centos7/x86_64/rstudio-server-rhel-1.4.1717-x86_64.rpm
sudo yum install rstudio-server-rhel-1.4.1717-x86_64.rpm
sudo yum install -y --nogpgcheck rstudio-server-rhel-1.0.153-x86_64.rpm
sudo rm rstudio-server-rhel-1.0.153-x86_64.rpm
R -e "install.packages('shiny', repos='http://cran.rstudio.com/')"
```

sudo wget https://download3.rstudio.org/centos5.9/x86_64/shiny-server-1.5.4.869-rh5-x86_64.rpm

sudo yum install -y --nogpgcheck shiny-server-1.5.4.869-rh5-x86_64.rpm

rm shiny-server-1.5.4.869-rh5-x86_64.rpm

We can also add more users at this point if we are working in a multiuser environment.

Create a rstudio-user account to handle the data analysis. Using the hadoop fs command, create a user directory on HDFS for rstudio-user.

sudo useradd rstudio-user

echo rstudio-user:password | rstudio-user

hadoop fs -mkdir /user/rstudio-user

hadoop fs -chmod 777 /user/rstudio-user

su rstudio-user

We must make sure the EC2 instance for the R-based data science environment has permission to read data from the chosen S3 bucket. Applications should only be able to read the S3 files that they need to execute, according to the concept of least privilege.

We create a security group in the EC2 launch wizard, which works as a virtual firewall for one or more instances, controlling traffic. We need to open ports 8787 and 3838 for RStudio Server and Shiny Server, respectively, for the R-based analysis environment.

After we have finished configuring the security group, click Create to start the instance. After the EC2 instance is up and running, we can connect to RStudio Server and R via a web browser. Use the newly established user and password for login credentials. The URL is as follows:

By navigating to the master node IP:8787, you can log into RStudio Server.

http://ec2-3-129-63-125.us-east-2.compute.amazonaws.com:8787/auth-sign-in

Some R packages necessitate the installation of Linux packages. Install the curl-devel Linux package so that we can use the R package "Rcurl" in the next stages. SSH into the EC2 instance and run the following command:

sudo yum install curl-devel

Set the environment variable SPARK_HOME and then run spark_connect.

```
library(sparklyr)
library(dplyr)
library(ggplot2)
Sys.setenv(SPARK_HOME="/usr/lib/spark")
config <- spark_config()
sc <- spark_connect(master = "yarn-client", config = config, version = '2.4.3')
```

The data can then be loaded and the R code for analysis obtained. We track changes as we transfer code from the laptop to AWS.

Amazon S3 is a highly scalable, secure, and long-lasting object storage service. It is simple to use, with a straightforward web service interface for storing and retrieving any quantity of data from anywhere on the internet.

```
install.packages("RCurl")
library("RCurl")
data <- read.table(textConnection(getURL(https://cgiardata.s3-us-west-
2.amazonaws.com/ccafs/COVID-19.csv")), sep=",", header=TRUE)
```

Shiny Server listens on port 3838 by default, therefore the new application may be found at the following URI:

http://ec2-3-129-63-125.us-east-
2.compute.amazonaws.com:3838/Rstudiouser/classification

The instance can be started in less than five minutes. Stopping is preferable to terminating since terminating can wipe out any data and code stored on the instance.

Shiny is a great way to quickly test a machine learning model or propose a machine learning challenge so that others can quickly input test data and be

astonished by the model. The models have been saved and a shiny app has been created, the CSV test data file is uploaded.

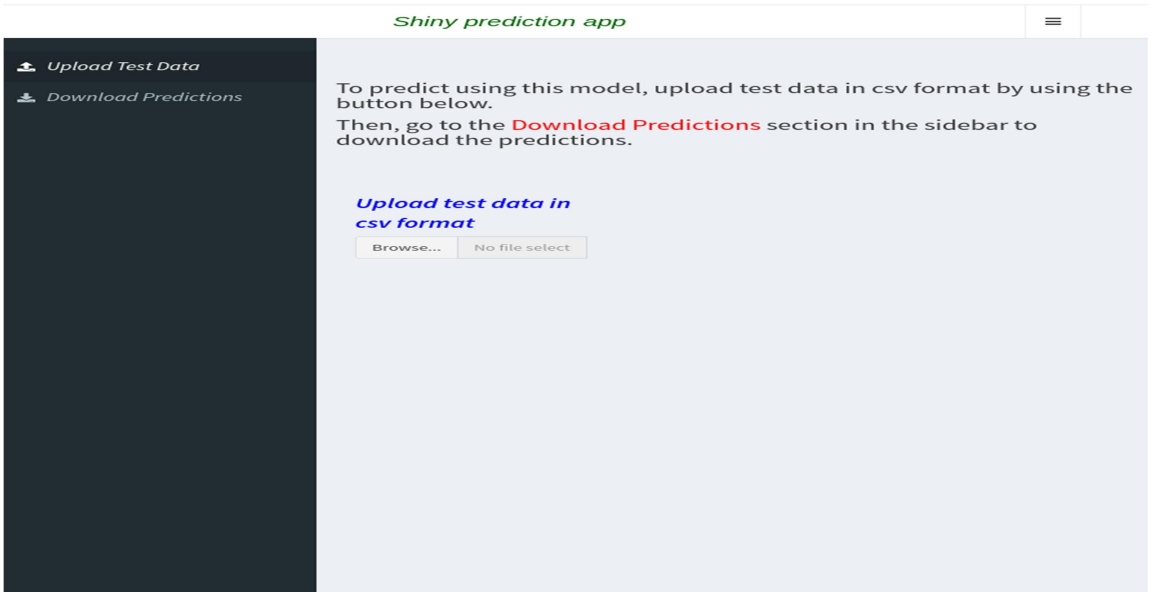

Figure 4.5:Upload CSV file (test data)

In the local spark mode, the output from Minhash function has been entered into the SVM, NB, RF, and LR Classification algorithm, and the performance metrics have been obtained. The accuracy and time consumed are the performance indicators. We repeat the experiment in local R, parallel R, and YARN Client mode to obtain the results. The tables listed below show the results of all three environments.

The four machine learning algorithms have been applied to the two datasets (COVID Vaccine Tweet and COVID IEEE Data Port Tweet) in the same way.

4.7.2 COVID IEEE Data Port Tweet Dataset Result

4.7.2.1 Performance Metric (time consumed) of all algorithms (k=4)

Table 4.10: ***Performance metric (time consumed) of SVM, NB, RF, and LR algorithms in local R and local spark mode (k=4, No. of Minhash=20)***

Mode	SVM	NB	RF	LR
Local R	65.04852 sec	43.8746 sec	43.9848 sec	45.23569 sec
Local Spark	39.36064 sec	42.61976 sec	43.53997 sec	40.0672 sec
YARN Spark	39.00956 sec	40.00453 sec	42.3468 sec	39.8795 sec

Minhash-NB has consumed lower time than the rest models in local R mode when k=4 with 43.8746 seconds. 39.36064 seconds have been spent by performing Minhash-SVM model in local spark mode. It is the lowest time in that mode as shown in Table (4.10).

Table 4.11: Performance metric (time consumed, Accuracy) of Minhash-LSH algorithm (k=4, No. of Minhash=20)

Minhash-LSH	k=4	Accuracy
Local Spark	55.40 sec	0.8500
YARN Client	49.568 sec	

4.7.2.2 Performance Metric (Accuracy) of all algorithms (k=4)

Table 4.12: Performance metric (accuracy) of SVM, NB, RF, and LR algorithms in all modes (k=4, No. of Minhash=20)

Mode	SVM	NB	RF	LR
Local R	1	0.9991	0.9999	1
Parallel R	1	1	1	1
Spark (4 nodes)	1	0.5022167	1	1
YARN (4 nodes)	1	0.5072333	1	1

- Naive Bayes is known as a bad estimator, so the probability outputs are not to be taken too seriously.
- Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

4.7.2.3 Performance Metric (time consumed) of all algorithms (k=5)

Table 4.13: ***: Performance metric (time consumed) of SVM, NB, RF, and LR algorithms in local R local spark and YARN Spark modes (k=5, No. of Minhash=20)***

Mode	SVM	NB	RF	LR
Local R	34.31487 sec	25.68507 sec	26.66179 sec	26.74767 sec
Local Spark	21.95478 sec	24.14055 sec	23.8856 sec	22.32216 sec
Yarn Spark	21.9087 sec	23.26701 sec	22.77633 sec	21.98765 sec

Table 4.14: Performance metric (time consumed, Accuracy) of Minhash-LSH (k=5, No. of minhash=20)

Minhash-LSH	k=5	Accuracy
Local Spark	30.53 sec	0.8543
YARN Client	26.652 sec	

4.7.2.4 Performance Metric (Accuracy) of all algorithms (k=5)

Table 4.15: ***Performance metric (accuracy) of SVM, NB, RF, and LR algorithms in all modes (k=5, No. of Minhash=20)***

Mode	SVM	NB	RF	LR
Local R	1	1	1	1
Parallel R	1	1	1	1
Spark (4 nodes)	1	0.50996	1	1
YARN (4 nodes)	1	0.5088833	1	1

4.7.3 COVID-19 Vaccine Stance Dataset Result

4.7.3.1 Performance Metric (time consumed) of all algorithms (k=4)

Table 4.16: Performance metric (time consumed) of SVM, NB, RF, and LR algorithms in local R, local spark and YARN Spark modes (k=4, No. of Minhash=20)

Mode	SVM	NB	RF	LR
Local R	12.51281 sec	7.192847 sec	7.322701 sec	6.291513 sec
Local Spark	1.668843 sec	3.928369 sec	2.04314 sec	1.70599 sec
YARN Spark	1.662387 sec	2.86571 sec	1.98307 sec	1.69549 sec

Table 4.17: Performance metric (time consumed, Accuracy) of Minhash-LSH algorithm (k=4, No. of Minhash=20)

Minhash-LSH	k=4	Accuracy
Local Spark	43.046 sec	0.836
YARN Client	38.994 sec	

Table (4.16) shows the implementation in R consumed much time than in Apache Spark. Minhash-LSH has consumed much time than Minhash-ML algorithms as shown in Table (4.16) and Table (4.17). When k=4, Minhash-SVM has consumed lower time than the rest models in local spark mode. While Minhash-LR consumed time is the lowest time in local R.

4.7.3.2 Performance Metric (Accuracy) of all algorithms when k=4

*Table 4.18: **Performance metric (accuracy) of SVM, NB, RF, and LR algorithms in all modes (k=4, No. of Minhash=20)***

Mode	SVM	NB	RF	LR
Local R	1	1	0.9944	1
Parallel R	1	1	0.9944	0.9944
Local Spark (4 cores)	0.9856333	0.548833	0.9777	0.9888
YARN Client (4 nodes)	0.9805015	0.5222	0.9861	0.9916

Table (4.18) shows that the Minhash-ML in local R obtained the highest accuracy in all algorithms. Moreover, Minhash-LR model accuracy is the most significant value in local spark and YARN Client modes. In these modes with different number of nodes, the accuracy does not show significant improvement. This is since all modes use the same algorithm. Also, there is no effect on the value of accuracy while using parallel R with 4 cores.

4.7.3.3 Performance Metric (time consumed) of all algorithms when k=5

Table 4.19: ***Performance metric (time consumed) of SVM, NB, RF, and LR algorithms in local R , local spark and YARN Spark mode (k=5, No. of Minhash=20)***

Mode	SVM	NB	RF	LR
Local R	19.83298 sec	5.440004 sec	6.073364 sec	5.951169 sec
Local Spark	1.640383 sec	1.765832 sec	1.782956 sec	1.656483 sec
YARN Spark	1.639979 sec	1.665678 sec	1.692945 sec	1.64034 sec

Table 4.20: Performance metric (time consumed, Accuracy) of Minhash-LSH algorithm (k=5, No. of Minhash=20)

Minhash-LSH	k=5	Accuracy
Local Spark	30.530 sec	0.841
YARN Client	26.653 sec	

Table (4.19) shows the Minhash-ML consumed in local R less time than in Apache Spark. Minhash-NB has consumed lower time than the rest models in local R when k=5.

4.7.3.4 Performance Metric (Accuracy) of all algorithms (k=5)

Table 4.21: ***Performance metric (accuracy) of SVM, NB, RF, and LR algorithms in all modes (k=5, No. of Minhash=20)***

Mode	SVM	NB	RF	LR
Local R	1	1	1	1
Parallel R	1	1	1	1
Local Spark (4 cores)	0.9812333	0.5012733	0.9805	0.9773
YARN Client (4 nodes)	0.9836601	0.5082967	0.9902	1

Table (4.21) shows that the Minhash-MLs models in local R have obtained the highest accuracy in all algorithms with 100%. There is no effect on the value of accuracy while using parallel R with 4 cores.

Number of shingles has been chosen as k=4 or k=5. When k=5, RF accuracy is better than when k=4 in all modes. k=5 is the best choice in Minhash-SVM, Minhash-RF, and Minhash-LR models in local R and YARN Client mode and in Minhash-LSH model. While k=4 is the best option in Minhash-NB model in all modes and Minhash-SVM and Minhash-LR in local spark mode.

4.8 Analysis and Discussion

In comparison to YARN Client mode, the algorithms in Local R mode take a lot longer. All the algorithms can be run in parallel, resulting in significant time savings. Overall, the YARN Client mode takes significantly less time to run algorithms than the local spark mode, proving that parallel computing with spark uses more nodes and reduces processing time.

CHAPTER FIVE

CONCLUSIONS AND FUTURE WORKS

5.1 Overview

An attempt is made in this dissertation to compare the performance of two modes accessible in Apache Spark and the local R environment. Using the Apache Spark machine learning framework, four classification algorithms have been implemented to do this comparison.

To guarantee that Apache Spark's parallel computing capabilities are fully leveraged, two datasets have been used. In terms of time and accuracy, we compare how well the algorithms perform. Also, Minhash-LSH has been performed and compared with Minhash-ML.

5.2 Conclusions

The study's findings are as follows:

1. The usage of a larger number of cores in the Local Cluster mode reduces the algorithm's execution time slightly, but not much. This is due to the Local mode, which uses a single machine to create the illusion of parallelism.

2. The running time of algorithms in the Hadoop YARN Client mode is faster than the local spark mode. With more cores in a cluster, more parallelism is achieved in computing the predictions.

3. Other performance measurements, such as accuracy, indicate no discernible differences between the two spark modes. This shows that the internal implementation of machine learning algorithms is unaffected by the number of cores used. Although the classifications are computed faster, the method by which they are produced stays the same and produces nearly identical results every time.

With more regularly distributed datasets, a higher level of accuracy can be reached.

4. k-shingles help to recode data set into actual class assignment which leads to a high accuracy result compared with standard works.

5. Minhash technique has increased the quality of the whole work, it used to distribute the tweets in iterative process to generate characteristics matrix and signature matrix.
6. The AWS used to reduce the implementation time in the testing stage with scalable framework.

5.3 Limitation

Apache Spark was built with the Scala programming language in mind and is extremely efficient.
It has R APIs which we employed. Although using these APIs does not result in performance degradation, it is important to note that Spark interprets these APIs into Scala for subsequent processing.

5.4 Future Work

1. Production-quality classifier will incorporate many different features beyond the vectors corresponding to the words in the text. Meaning that add feature like location of user, age of user and node connected link to other user this may be help us to make it feature selection to improve the classification performance.
2. Moreover, it can be investigate the effect of different Bloom filter bit vector sizes, in classification performance and storage space compression

REFERENCES

[1] M. Zaharia *et al.*, "Apache spark: a unified engine for big data processing," vol. 59, no. 11, pp. 56-65, 2016.

[2] A. R. M. Forkan, I. Khalil, and M. Atiquzzaman, "ViSiBiD: A learning model for early discovery and real-time prediction of severe clinical events using vital signs as big data," *Computer Networks,* vol. 113, pp. 244-257, 2017/02/11/ 2017, doi: https://doi.org/10.1016/j.comnet.2016.12.019.

[3] P. Zhang, S. Hu, J. He, Y. Zhang, G. Huang, and J. Zhang, "Building Cloud-Based Healthcare Data Mining Services," in *2016 IEEE International Conference on Services Computing (SCC)*, 27 June-2 July 2016 2016, pp. 459-466, doi: 10.1109/SCC.2016.66 .

[4] J. Fu, J. Sun, and K. Wang, "SPARK – A Big Data Processing Platform for Machine Learning," in *2016 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology ,Industrial Information Integration (ICIICII)*, 3-4 Dec. 2016 2016, pp. 48-51, doi: 10.1109/ICIICII.2016.0023 .

[5] H. S. Bhosale, D. P. J. I. J. o. S. Gadekar, and R. Publications, "A review paper on big data and hadoop," vol. 4, no. 10, pp. 1-7, 2014.

[6] S. Salloum, R. Dautov, X. Chen, P. X. Peng, and J. Z. Huang, "Big data analytics on Apache Spark," *International Journal of Data Science and Analytics,* vol. 1, no. 3, pp. 145-164, 2016/11/01 2016, doi: 10.1007/s41060-016-0027-9.

[7] V. S. Jonnalagadda, P. Srikanth, K. Thumati, S. H. Nallamala, K. J. I. J. o. C. S. T. Dist, and Technology, "A review study of apache spark in big data processing ",vol. 4, no. 3, pp. 93-98, 2016.

[8] M. Parekh and B. Saleena, "Designing a Cloud Based Framework for HealthCare System and Applying Clustering Techniques for Region Wise Diagnosis," *Procedia Computer Science,* vol. 50, pp. 537-542, 2015/01/01/ 2015, doi: https://doi.org/10.1016/j.procs.2015.04.029.

[9] A. C. Sanders *et al.*, "Unmasking the conversation on masks: Natural language processing for topical sentiment analysis of COVID-19 Twitter discourse," *medRxiv,* p. 2020.08.28 ,2020 ,20183863.doi: 10.1101/2020.08.28.20183863.

[10] A. S. Imran, S. M. Daudpota, Z. Kastrati, and R. Batra, "Cross-Cultural Polarity and Emotion Detection Using Sentiment Analysis and Deep Learning on COVID-19 Related Tweets," *IEEE Access,* vol. 8, pp. 181074-181090, 2020, doi: 10.1109/ACCESS.2020.3027350.

[11] K. Chakraborty, S. Bhatia, S. Bhattacharyya, J. Platos, R. Bag, and A. E. Hassanien, "Sentiment Analysis of COVID-19 tweets by Deep Learning Classifiers—A study to show how popularity is affecting accuracy in social media," *Applied Soft Computing,* vol. 97, p. 1067 /01/12/2020 ,54 ,2020doi: https://doi.org/10.1016/j.asoc.2020.106754.

[12] E. Alomari, R. Mehmood, and I. Katib, “Road traffic event detection using Twitter data, Machine Learning, and Apache Spark,” 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), 2019.

[13] J. Samuel, G. Ali, M. Rahman, E. Esawi and Y. Samuel, "COVID-19 Public Sentiment Insights and Machine Learning for Tweets Classification", SSRN Electronic Journal, 2020. Available: 10.2139/ssrn.3584990.

[14] G. Barkur, Vibha and G. Kamath, "Sentiment analysis of nationwide lockdown due to COVID 19 outbreak: Evidence from India", Asian Journal of Psychiatry, vol. 51, p. 102089, 2020. Available: 10.1016/j.ajp.2020.102089.

[15] L. A. Cotfas, C. Delcea, I. Roxin, C. Ioanăş, D. S. Gherai, and F. Tajariol, "The Longest Month: Analyzing COVID-19 Vaccination Opinions Dynamics From Tweets in the Month Following the First Vaccine Announcement," *IEEE Access,* vol. 9, pp. 33203-33223, 2021, doi: 10.1109/ACCESS.2021.3059821.

[16] F. Rustam, M. Khalid, W. Aslam, V. Rupapara, A. Mehmood, and G. S. J. P. o. Choi, "A performance comparison of supervised machine learning models for Covid-19 tweets sentiment analysis," vol. 16, no ,2 .p. e0245909, 2021.

[17] H. Elzayady, K. Badran and G. Salama, "Sentiment Analysis on Twitter Data using Apache Spark Framework", 2018 13th International Conference on Computer Engineering and Systems (ICCES), 2018. Available: 10.1109/icces.2018.8639195 [Accessed 27 November 2021].

[18] M. Mohammed, M. B. Khan, and E. B. M. Bashier, *Machine learning: algorithms and applications*. Crc Press, 2016.

[19] D. Rao and D. Yarowsky, "Detecting latent user properties in social media," in *Proc. of the NIPS MLSN Workshop*, 2010: Citeseer, pp. 1-7 .

[20] R. Ardianto, T. Rivanie ,Y. Alkhalifi, F. S. Nugraha, and W. J. J. I. K. d. I. Gata, "Sentiment Analysis on E-Sports For Education Curriculum Using Naive Bayes and Support Vector Machine," vol. 13, no. 2, pp. 109-122, 2020.

[21] R. Sint, S. Stroka, S. Schaffert, and R. Ferstl, “Combining Unstructured, Fully Structured and Semi-Structured Information in Semantic Wikis.,” in SemWiki, 2009, vol. 464.

[22] T. N. Yogi, N. J. I. J. o. I. S. Paudel, Engineering, and Technology, "Comparative Analysis of Machine Learning Based Classification Algorithms for Sentiment Analysis," vol. 7, no. 6, pp. 1-9.

[23] V. Carletti, R. Di Lascio, P. Foggia and M. Vento, "A Semantic Reasoner Using Attributed Graphs Based on Intelligent Fusion of Security Multi-sources Information", Activity Monitoring by Multiple Distributed Sensing, pp. 73-86, 2014. Available: 10.1007/978-3-319-13323-2_7 [Accessed 19 November 2021.

[24] M. Bharati and M. Ramageri, "Data mining techniques and applications," 2010.

[25] W. Wu, B. Li, L. Chen, J. Gao, and C. Zhang, "A Review for Weighted MinHash Algorithms," *IEEE Transactions on Knowledge and Data Engineering* ,pp. 1-1, 2020, doi: 10.1109/TKDE.2020.3021067.

[26] *CS174 SP99, lecture 9 summary, John Canny*. [Online]. Available: https://people.eecs.berkeley.edu/~jfc/cs174lecs/lec9/lec9.html. [Accessed: 04-Dec-2021.

[27] M. Zaharia *et al.*, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *9th {USENIX}*

Symposium on Networked Systems Design and Implementation ({NSDI} 12) ,2012 ,pp. 15-28 .

[28] D. P. and K. Ahmed, “A survey on Big Data Analytics: Challenges, open research issues and Tools,” International Journal of Advanced Computer Science and Applications, vol. 7, no. 2, 2016.

[29] Karau, H. (2013).Fast Data Processing with Spark. Packt Publishing Ltd.

[30] A. Oussous, F.-Z. Benjelloun, A. Ait Lahcen, and S. Belfkih, “Big Data Technologies: A survey,” Journal of King Saud University - Computer and Information Sciences, vol. 30, no. 4, pp. 431–448, 2018.

[31] P. Raj, “The Hadoop Ecosystem Technologies and tools,” Advances in Computers, pp. 279–320, 2018.

[32] H. Karau, A. Konwinski, P. Wendell, M. Zaharia, Learning Spark:Lightning-Fast Big Data Analysis, O’Reilly Media, Inc., 2015.

[33] V. K. Vavilapalli, A. C. Murthy, C. Douglas, S. Agarwal, M. Konar,R. Evans, T. Graves, J. Lowe, H. Shah, S. Seth, B. Saha, C. Curino,O. O’Malley, S. Radia, B. Reed, E. Baldeschwieler, Apache HadoopYARN: Yet another resource negotiator, in: 4th Annual Symposiumon Cloud Computing, SOCC ’13, ACM, New York, USA, 2013, pp.5:1–5:16.doi:10.1145/2523616.2523633.

[34] B. Hindman, A. Konwinski, M. Zaharia, A. Ghodsi, A. D. Joseph,R. Katz, S. Shenker, I. Stoica, Mesos: a platform for fine-grained re-source sharing in the data center., in: NSDI, Vol. 11, 2011, pp. 22–22.

[35] "What is Apache Spark? The big data platform that crushed Hadoop." https://www.infoworld.com/article/3236869/what-is-apache-spark-the-big-data-platform-that-crushed-hadoop.html (accessed 2021).

[36] K. El Bouchefry and R. S. de Souza, "Chapter 12 - Learning in Big Data: Introduction to Machine Learning," in *Knowledge Discovery in Big Data from Astronomy and Earth Observation*, P. Škoda and F. Adam Eds.: Elsevier, 2020, pp. 225-249.

[37] G. Malewicz *et al.*, "Pregel: a system for large-scale graph processing," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, 2010, pp. 135-146 .

[38] P. Sharma, "Performance Comparison of Apache Spark MLlib," 2018.

[39] D. P. Timothy and A. K. Santra, "A hybrid cryptography algorithm for cloud computing security," in *2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS)*, 10-12 Aug. 2017 2017, pp. 1-5, doi: 10.1109/ICMDCS.2017.8211728 .

[40] "RDD Programming Guide - Spark 2.3.0 Documentation." https://spark.apache.org/docs/2.3.0/rdd-programming-guide.html#resilient-distributed-datasets-rdds (accessed 2019).

[41] "SparkContext (Spark 2.3.0 JavaDoc)." https://spark.apache.org/docs/2.3.0/api/java/org/apache/spark/SparkContext.html (accessed 2019.(

[42] K. Aziz, D. Zaidouni, and M. Bellafkih, “Leveraging Resource Management for efficient performance of Apache Spark,” Journal of Big Data, vol. 6, no. 1, 2019.

[43] "Cluster Mode Overview - Spark 3.1.2 Documentation." https://spark.apache.org/docs/latest/cluster-overview.html (accessed 2019.

[44] "Spark local (pseudo-cluster)." https://jaceklaskowski.gitbooks.io/mastering-apache-spark/spark-local.html. (accessed 2019).

[45] S. V. Pashte, C. J. Awati, and S. S. Kharade, "Overcome Key Escrow Problem with Attribute-Based Data Access Policy & Efficient Cloud Environment," in *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, 17-18 Aug. 2017 2017, pp. 1-6, doi: 10.1109/ICCUBEA.2017.8463808 .

[46] R. S. Somula, R. J. S. C. P. Sasikala, and Experience, "A survey on mobile cloud computing: mobile computing+ cloud computing (MCC= MC+ CC)," vol. 19, no. 4, pp. 30.2018 ,337-9

[47] V. J. R. Winkler, "Cloud computing architecture," Securing the Cloud, pp. 29–53, 2011.

[48] Y. Al-Dhuraibi, F. Paraiso, N. Djarallah, and P. Merle, "Elasticity in Cloud Computing: State of the Art and Research Challenges," *IEEE Transactions on Services Computing,* vol. 11, no. 2, pp. 430-447, 2018, doi: 10.1109/TSC.2017.2711009.

[49] B. Calabrese and M. Cannataro, "Cloud computing in healthcare and biomedicine. Scalable Comput.: Pract. Exp. 16 (1), 1–18 (2015)," ed, 2014.

[50] A. Khajeh-Hosseini, I. Sommerville, and I. J. a. p. a. Sriram, "Research challenges for enterprise cloud computing," 2010.

[51] H. Xie, J. Li, and H. J. a. p. a. Xue, "A survey of dimensionality reduction techniques based on random projection," 2017.

[52] B. Liu, W. J. W. D. M. Mining, ser. Data-Centric Systems, and A. S. B. Heidelberg, "Exploring Hyperlinks, Contents, and Usage Data," 2011.

[53] A. Holzinger, "Machine learning for health informatics," in *Machine learning for health informatics*: Springer, 2016, pp. 1-24.

[54] K. Potdar, T. S. Pardawala, and C. D. J. I. j. o. c. a. Pai, "A comparative study of categorical variable encoding techniques for neural network classifiers," vol. 175, no. 4, pp. 7-9, 2017.

[55] M. Steinbach, G. Karypis, and V. Kumar, "A Comparison of Document Clustering Techniques," 2000 .

[56] A. Bahga and V. Madisetti, *Cloud computing: A hands-on approach*. San Bernardino, CA?: Arshdeep Bahga & Vijay Madisem, 2014.

[57] A. Bag, M. Sahoo, and A. K. Mohanty, “An assessment of Diagnosis and Prognosis of Breast Cancer Using Image Mining,” *Amalendu Bag,etal.Journal of Engineering Research and Application*, vol. 10, no. 2248-9622, pp. 16–25, 2020.

[58] L. Wang, C. A. J. I. J. o. M. Alexander, Engineering, and M. Sciences, "Machine learning in big data," vol ,1 .no. 2, pp. 52-61, 2016.

[59] Synced, “How random forest algorithm works in machine learning,” *Medium*, 25-Jun-2018. [Online]. Available: https://synced.medium.com/how-random-forest-algorithm-works-in-machine-learning-3c0fe15b6674. [Accessed: 05-Dec-2021].

[60] V. Kumar, "Evaluation of computationally intelligent techniques for breast cancer diagnosis", Neural Computing and Applications, vol. 33, no. 8, pp. 3195-3208, 2020. Available: 10.1007/s00521-020-05204-y.

[61] A. Burkov, The hundred-page machine learning book. [S.l.]: Andriy Burkov, 2019.

[62] A. Rajaraman and J. D. Ullman, *Mining of massive datasets*. Cambridge University Press, 2011.

Appendix A

Problems and Solutions

Problem 1: Failed to start the namenode: java.net.BindException: Address already in use
Solution:
Changing the port in core-site.xml from 9000 to 9001 as shown below:

```
<property>
  <name>fs.default.name</name>
 <value>hdfs://krb5.wafaa.io:9001</value>
</property>
```

hadoop namenode format
start-all.sh

Problem 2: SSH: connect to host localhost port 22: connection refused.
Solution:
Make password for root:
sudo passwd root

Problem 3: Internet stops working, "Failed to add /run/systemd/ask-password to directory watch: No space left on device"
Solution:
Edit the file /etc/sysctl.conf to include the line:
fs.inotify.max_user_watches=1048576

Problem 4: DataNode startup is unsuccessful - java.net.BindException: Port in use: localhost:0 Caused by: java.net.BindException: C
（1）java.net.BindException: Port in use: localhost:0
（2）Caused by: java.net.BindException: Cannot assign requested address
Solution:
1. Add the directory addresses of namenode and Datanode in core-site.xml as follows:

```
    <property>
    <name>dfs.namenode.name.dir</name>
<value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
</property>
    <property>
        <name>dfs.dataname.data.dir</name>
```

```
<value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
</property>
<property>
        <name>hadoop.tmp.dir</name>
        <value>//hadoop-2.7.3/data/tmp</value>
</property
```

2. Edit the '/etc/hosts' file of the server and all clients by adding the following lines:
127.0.0.1 localhost localhost.localdomain localhost4.localdomain4
::1 localhost localhost.localdomain localhost6.localdomain6

Problem 5: $ more /usr/local/hadoop/logs/hadoop-hduser-datanode-ubuntu.log
java.io.IOException: Incompatible clusterIDs in /usr/local/hadoop_tmp/hdfs/datanode: namenode clusterID = CID-e4c3fed0-c2ce-4d8b-8bf3-c6388689eb82; datanode clu
sterID = CID-2fcfefc7-c931-4cda-8f89-1a67346a9b7c
Solution:
Stop the cluster and issue the below command and then start the cluster again.
Stop-all.sh
sudo rm -rf /usr/local/hadoop_tmp/hdfs/datanode/*

Problem 6: Any of the service not working.
Solution:
Check the log file at /usr/local/hadoop/logs

Problem 7: Error: Could not load YARN classes. This copy of Spark may not have been compiled with YARN support.
Run with --help for usage help or --verbose for debug output.
Solution:
You need to compile spark against Yarn to use it.
build/sbt -Pyarn -Phadoop-2.x assembly

الخلاصة

تلعب وسائل التواصل الاجتماعي دورًا أساسيًا في توليد حجم كبير من البيانات المنظمة وشبه المهيكلة وغير المهيكلة. تعتبر أداة تسويق قوية ورؤية تجارية. ينطوي التحدي الرئيسي الذي يواجه هذه البيانات على الحاجة إلى بناء نموذج قوي باستخدام تقنيات فعالة لاستخراج معرفة جديدة واتخاذ القرار الصحيح في وقت أقل ، خاصة في البيانات الطبية. تحليل المشاعر هو أحد أساليب تحليل البيانات الموسعة المستخدمة لاستخراج الآراء العامة لبيانات اجتماعية كبيرة والتي عادة ما تشمل جمل قصيرة ، بشكل عام ، لم يتم إنشاؤها باستخدام قواعد نحوية مناسبة. تهدف هذه الدراسة إلى إجراء تصنيف المشاعر على التغريدات بطريقة فعالة وفي الوقت المناسب باستخدام نموذج هجين من Min-hash والتعلم الآلي للحصول على نتائج دقيقة للغاية ، وللتنفيذ المتوازي باستخدام إطار عمل برمجة Apache Spark. لإجراء تحليل المشاعر ، استخدمنا مجموعتي بيانات ، تغريدة COVID-19 مقدمة من منفذ بيانات IEEE ومجموعات بيانات لقاح COVID-19 التي تتضمن ثلاثة حقول – tweet Id ، وlabel ، و tweetنفسها.

يتكون نظام تحليل المشاعر المقترح من أربع خطوات: جمع البيانات ، وتنقية البيانات والمعالجة المسبقة ، منهاش وتصنيف المشاعر. يتيح النظام أداء عالي المستوى لتصنيف المشاعر مع الاستفادة من الجمع بين Minhash والمصنف القائم على التعلم. تتم مقارنة أداء Minhash مع التجزئة الحساسة للمنطقة (LSH) بأداء Minhash مع كل من الانحدار اللوجستي (LR) ، والغابة العشوائية (RF) ، و Naive Bayes (NB) ، وآلة المتجهات الداعمة (SVM) على التوازي و بطريقة موزعة. تُستخدم معلمات الأداء مثل الوقت المستغرق والدقة التي تم الحصول عليها من Confusion Matrix لتصنيف التغريدات إلى إيجابية وسلبية ومحايدة. تم تطبيق النتائج التي تم الحصول عليها في التحليل المقارن لتحليل سلوك المصنفات في مجموعة Amazon Web Services (AWS) ، ومجموعة spark local وفي النظام التقليدي. تشير النتائج إلى أن النماذج في بيئة Spark كانت فعالة للغاية في معالجة البيانات .تظهر النتائج التجريبية أن LR و RF يتفوقان على مصنفات SVM و NB .

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل
كلية تكنولوجيا المعلومات
قسم البرامجيات

تحليل البيانات باستخدام
Minhash و التعليم الالي في بيئة Apache Spark

رسالة مقدمة
الى مجلس كلية تكنولوجيا المعلومات للدراسات العليا بجامعة بابل
كجزء من متطلبات درجة الدكتوراه فلسفة في تكنولوجيا المعلومات – برامجيات

من قبل
وفاء شاكر محمد حسن

بإشـــراف
أ. د رفاه محمد كاظم
أ. م. د مهدي عبادي مانع مهدي

1443 A.H
2021 A.D