**COVISHIELD REVIEW ANALYSIS**

**A Project Report Submitted**

**In Partial Fulfillment of the Requirements**

**for the Degree of**

**MASTER OF COMPUTER APPLICATION**

### by

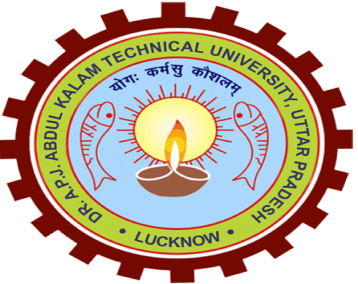
**BhanuPratap Singh**

**(1900290149034)**

**Under the Supervision of**

**Dr. Ajay Kumar Shrivastava**

**KIET Group of Institutions, Ghaziabad**

****

**to the**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**DR. APJ ABDUL KALAM TECHNICAL UNIVERSITY**

**(Formerly Uttar Pradesh Technical University) LUCKNOW**

**August 2021**

**DECLARATION**

I hereby declare that the work presented in this report entitled “COVISHIELD REVIEW ANALYSIS", was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name : Bhanu Pratap Singh

Roll. No. : 1900290149034

Department : Computer Applications

(Signature)



**CERTIFICATE**

Certified that **BhanuPratap Singh** (**1900290149034**) has carried out the project work presented in this report entitled “**Covishield Review Analysis**” for the award of **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University, Lucknow under my supervision. The report embodies result of original work, and studies are carried out by the student himself and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University.

**Dr. Ajay Kr. Shrivastava External Examiner**

Professor & Head

Dept. of Computer Applications

KIET Group of Institutions, Ghaziabad

Date:

**ABSTRACT**

The severe acute respiratory syndrome coronavirus-2 (SARS-CoV-2) outbreak that started in Wuhan, China, in 2019 resulted in a pandemic not seen for a century, and there is an urgent need to develop safe and efficacious vaccines.Covishield is a recombinant, replication-deficient chimpanzee adenovirus vector encoding the SARS-CoV-2 Spike (S) glycoprotein. Following administration, the genetic material of part of corona virus is expressed which stimulates an immune response.Clinical trials have shown that after 2 doses of the vaccine almost all recipients developed antibodies. The Centre for Disease Control (CDC), USA does not recommend antibody testing to assess for immunity to SARS-CoV-2 following COVID-19 vaccination. In this project as well as research I am going to analyse what are the reviews against one of the vaccinei.e, COVISHIELD. The data belongs to the Twitter website where people had posted reviews their views on Covishield Vaccination.

As we know now a day’s social media has been so popular among all, receiving more attention, time and people are using this as platform to express their views and other likings and disliking. Everybody posts many things like their love, sorrow, anger and opinions about products, places and etc. There are so many social media’s platform like Facebook, Twitter, Instagram, Whatsapp, Snapchat, Linkedin, Blogger, Quora and many more. People are spending lots of time with these sites and expressing their opinion on many issues. Twitter is one of the popular social media used for official statements as well as personal views, which allows users to publish micro-blogging short messages called tweets with limited length that are visible to your friends or followers. So in short tweets can be named positive, negative, in support or unbiased. In this project I amanalyzing sentiments/reviews of Covishield vaccination related to covid19 virus. The dataset is taken from website named Kaggle. The analysis on the dataset is done using Big Data Analytics tools like Apache Hadoop. In the end the result shows the user’s count of positive, negative and neutral tweets based on the words available in the dictionary file.

**ACKNOWLEDGEMENT**

Success in life is never attained single handedly. My deepest gratitude goes to my Project supervisor, **Dr. Ajay Kumar Shrivastava, Professor and Head, Department of Computer Applications**forhis guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions. Words are not enough to express my gratitude for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

**Bhanu Pratap Singh**

**1900290149034**

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
|  | Page No. |
| Declaration | ii |
| Certificate | iii |
| Abstract | iv |
| Acknowledgement | v |
| List of Figures | viii |
| List of Tables | ix |
| **CHAPTER 1: INTRODUCTION** | **1-8** |
| 1.1 Project Description | 1 |
| 1.2 Project Purpose | 2 |
| 1.3 Big Data and Its Characteristics | 2 |
| 1.4 Big Data Analysis Using Hadoop | 4 |
| **CHAPTER 2 :LITERATURE REVIEW** | **9-14** |
| 2.1 Sentiment Analysis | 9 |
| 2.2 Twitter and Covid19 Tweets | 10 |
| 2.3 Big Data | 11 |
| 2.4 Hadoop | 11 |
| 2.5 HDFS | 12 |
| 2.6 MapReduce | 13 |
| 2.7 Apache Hive | 13 |
| **CHAPTER 3 : REQUIREMENT SPECIFICATIONS** | **15-22** |
| 3.1 Hardware Requirements | 15 |
| 3.2 Software Requirements | 15 |
| 2.1 Data Requirements | 22 |
| **CHAPTER 4 : DESIGN** | **23-27** |
| 4.1 0 level Data Flow Diagram | 23 |
| 4.2 1 level Data Flow Diagram | 23 |
| 4.3 Data Dictionary | 24 |
| **CHAPTER 5 :BIG DATA TESTING** | **28-34** |
| 5.1 Big Data Testing Strategy | 28 |
| 5.2 How Big Data Testing Strategy Works | 29 |
| 5.3 How to Test Hadoop Applications | 30 |
| 5.4 Benefits of Big Data Testing Strategy | 32 |
| 5.5 Big Data Testing Challenges | 33 |
| **CHAPTER 6 :IMPLEMENTATION &WORKFLOW** | **35-49** |
| 6.1 Linux Machine Setup using Oracle VM and CentOS 7 | 35 |
| 6.2 Setting Up Single Node Hadoop Cluster | 35 |
| 6.3 Data Flow from Local Disk To HDFS using Apache Nifi | 42 |
| 6.4 Analysis of Tweets using HIVE | 44 |
| **CHAPTER 6 : CONCLUSION AND FUTURE SCOPE** | **50-51** |
| 6.1 Conclusion | 50 |
| 6.2 Future Scope | 51 |

**LIST OF FIGURES**

[1.1 HADOOP-HDFS 5](#_Toc76764665)

[1.2 HADOOP CLUSTER 8](#_Toc76764666)

[3.1 ORACLE VIRTUALBOX 16](#_Toc76764667)

[3.2 CENTOS 17](#_Toc76764668)

[3.3 APACHE HADOOP 19](#_Toc76764669)

[3.4 HIVE ARCHITECHTURE 20](#_Toc76764670)

[3.5 POWER BI SERVICE 21](#_Toc76764671)

[3.6 POWER BI FEATURES 21](#_Toc76764672)

[3.7POWER BI DESKTOP 22](#_Toc76764673)

[4.1 0-LEVEL DFD 23](#_Toc76764674)

[4.2 1-LEVEL DFD 24](#_Toc76764675)

[5.1 BIG DATA TESTING 31](#_Toc76764676)

[5.1 HADOOP SETUP 37](#_Toc76764677)

[5.2 HDFS UI WITH DIRECTORIES 42](#_Toc76764678)

[5.3 NIFI SETUP 43](#_Toc76764679)

[5.4 FLOW FILES IN NIFI 43](#_Toc76764680)

[5.5 HIVE SETUP 44](#_Toc76764681)

[5.6 RUNNING NODES 45](#_Toc76764682)

[5.7 HIVE TABLES 47](#_Toc76764683)

[5.7 POSITIVE, NEGATIVE AND NEUTRAL TWEETS 48](#_Toc76764684)

[5.8 FINAL RESULT 49](#_Toc76764685)

[5.9 VISUALIZATION 49](#_Toc76764686)

**LIST OF TABLES**

[4.1 Covitweets 25](#_Toc76764283)

[4.2 Dictionary 26](#_Toc76764284)

[4.3 Splitwords 26](#_Toc76764285)

[4.4 scoretable 26](#_Toc76764286)

[4.5 Covisentiment 27](#_Toc76764287)

CHAPTER 1

INTRODUCTION

**1.1 PROJECT DESCRIPTION**

Covishield is one of the Vaccine which is helpful in cure of amid Covid-19 disease. In this project I am working on the reviews of this vaccine i.e, **Covishield**from the Twitter Source by taking the tweets from the dataset. As we know dataset contains huge amount of data know as Big Data. With the help of the Analytics Tool the analysis will be done on the data and finally the data will produced the result as a sentiment of the tweets.

Big Data Analytics is the way of processing the large amount of data. Hadoop is a popular open-source software which is very useful in analysing the larger data. Hadoop provides several tools for this purpose like Hive, Pig, HBase, Cassandra etc. In this Project, I have used Hadoop framework. For the analysis of Covid tweets dataset, Hive tool is used with Hadoop framework. The aim is to analyse the sentiment of tweets and classify the users’ tweet in one of the 3 categories – positive, negative, neutral. The tweets are classified into these categories using a dictionary text file which contains English words with its score. At last, a graph is plotted showing total number of positive, negative and neutral tweets. Visualization of the sentiment counts will give better understanding of the dataset.

**1.2 PROJECT PURPOSE**

Now a day, there are millions of users on social media who are active daily and generate huge amount of data daily (In TB’s and PB’s). Therefore, it is difficult to store and manage that unstructured data hence now it has become a tedious task to do. We are going to use Hadoop technology to maintain this huge amount of data. This data will be stored in HDFS (Hadoop Distributed File System) format. Hadoop is a platform, which is used to store distributed and computational data. The main purpose of this project is to analyse large dataset of covid tweets and perform sentiment analysis using different Big Data Analytics tools like Hadoop, Hive, MapReduce. This sentiment analysis will result in classification of tweets in three categories – positive, negative and neutral. This project also aims towards deep learning of Big Data Analytics.

* 1. **BIG DATA AND ITS CHARACTERSTICS**

**Big Data** is a collection of data that is huge in volume yet growing exponentially with time. It is a data with so large size and complexity that none of traditional data management tools can store it or process it efficiently. Big data is also a data but with huge size.

In recent years, Big Data was defined by the “*3Vs*” but now there is “*5Vs*” of Big Data which are also termed as the characteristics of Big Data as follows:

**1. Volume:**

* The name ‘Big Data’ itself is related to a size which is enormous.
* Volume is a huge amount of data.
* To determine the value of data, size of data plays a very crucial role. If the volume of data is very large then it is actually considered as a ‘Big
* Data’. This means whether a particular data can actually be considered as a Big Data or not, is dependent upon the volume of data.
* Hence while dealing with Big Data it is necessary to consider a characteristic ‘Volume’.
* *Example:* In the year 2016, the estimated global mobile traffic was 6.2 Exabytes(6.2 billion GB) per month. Also, by the year 2020 we will have almost 40000 ExaBytes of data.

**2. Velocity:**

* Velocity refers to the high speed of accumulation of data.
* In Big Data velocity data flows in from sources like machines, networks, social media, mobile phones etc.
* There is a massive and continuous flow of data. This determines the potential of data that how fast the data is generated and processed to meet the demands.
* Sampling data can help in dealing with the issue like ‘velocity’.
* *Example:* There are more than 3.5 billion searches per day are made on Google. Also, FaceBook users are increasing by 22%(Approx.) year by year.

**3. Variety:**

* It refers to nature of data that is structured, semi-structured and unstructured data.
* It also refers to heterogeneous sources.
* Variety is basically the arrival of data from new sources that are both inside and outside of an enterprise. It can be structured, semi-structured and unstructured.
  + **Structured data**: This data is basically an organized data. It generally refers to data that has defined the length and format of data.
  + **Semi- Structured data**: This data is basically a semi-organised data. It is generally a form of data that do not conform to the formal structure of data. Log files are the examples of this type of data.
  + **Unstructured data**: This data basically refers to unorganized data. It generally refers to data that doesn’t fit neatly into the traditional row and column structure of the relational database. Texts, pictures, videos etc. are the examples of unstructured data which can’t be stored in the form of rows and columns.

**4. Veracity:**

* It refers to inconsistencies and uncertainty in data, that is data which is available can sometimes get messy and quality and accuracy are difficult to control.
* Big Data is also variable because of the multitude of data dimensions resulting from multiple disparate data types and sources.
* *Example:* Data in bulk could create confusion whereas less amount of data could convey half or Incomplete Information.

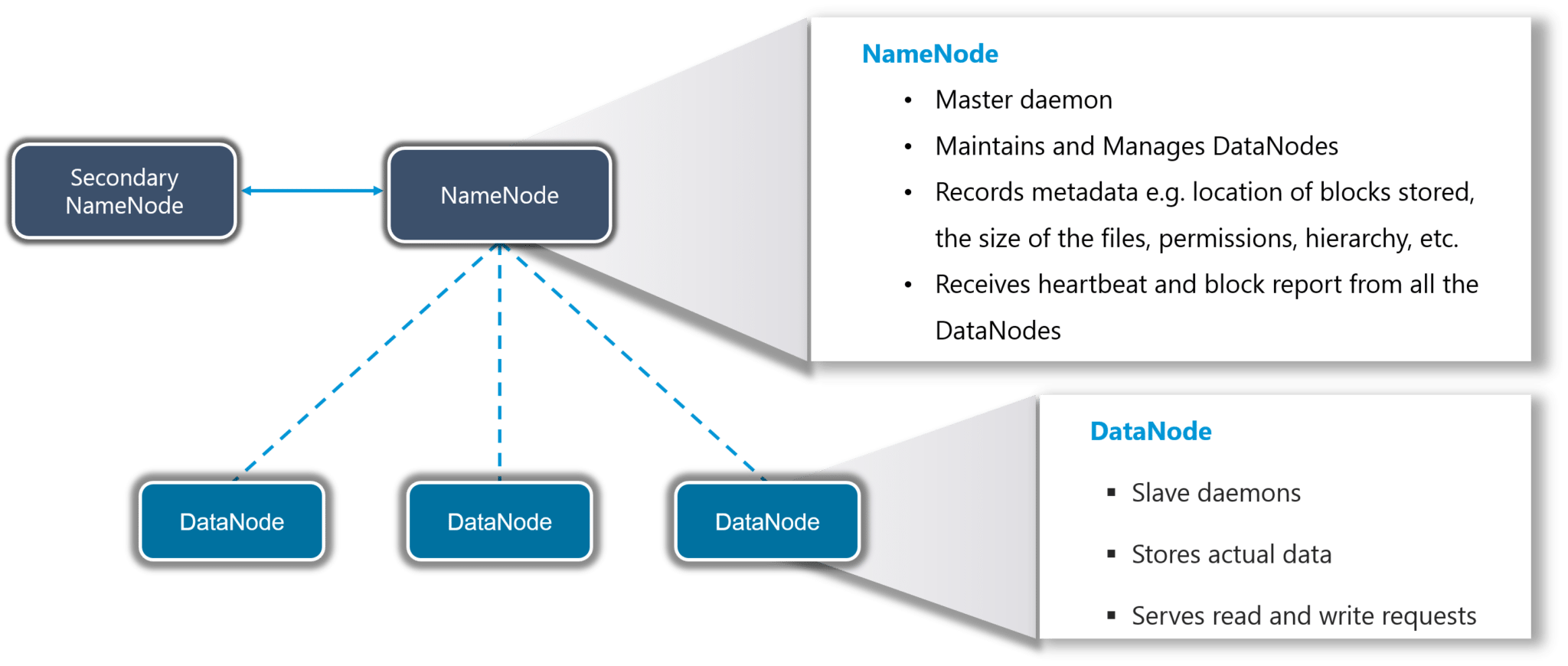
**5. Value:**

* After having the 4 V’s into account there comes one more V which stands for Value!. The bulk of Data having no Value is of no good to the company, unless you turn it into something useful.
* Data in itself is of no use or importance but it needs to be converted into something valuable to extract Information. Hence, you can state that Value! is the most important V of all the 5V’s.
  1. **BIG DATA ANALYSIS USING HADOOP**

Hadoop is a framework that allows you to first store Big Data in a distributed environment, so that, you can process it parallelly.  There are basically two components in Hadoop. The first one is ***HDFS*** for storage (Hadoop distributed File System), that allows you to store data of various formats across a

cluster. The second one is ***YARN***, for resource management in Hadoop. It allows parallel processing over the data, i.e. stored across HDFS.

* **HDFS:** HDFS creates an abstraction; let me simplify it for you. Similar as virtualization, you can see HDFS logically as a single unit for storing Big Data, but actually you are storing your data across multiple nodes in a distributed fashion. HDFS follows master-slave architecture. In HDFS, Namenode is the master node and Datanodes are the slaves. Namenode contains the metadata about the data stored in Data nodes, such as which data block is stored in which data node, where are the replications of the data block kept etc. The actual data is stored in Data Nodes. I also want to add, we actually replicate the data blocks present in Data Nodes, and the default replication factor is 3. Since we are using commodity hardware and we know the failure rate of these hardwares are pretty high, so if one of the DataNodes fails, HDFS will still have the copy of those lost data blocks.

****

1Fig. 1.1 HADOOP-HDFS

* **YARN**: **YARN** performs all your processing activities by allocating resources and scheduling tasks. It has two major components, i.e.ResourceManager and NodeManager.

ResourceManager is again a master node. It receives the processing requests and then passes the parts of requests to corresponding NodeManagers accordingly, where the actual processing takes place. NodeManagers are installed on every DataNode. It is responsible for the execution of the task on every single DataNode.

* **MapReduce** — [MapReduce](https://databricks.com/glossary/hadoop-ecosystem) is both a programming model and big data processing engine used for the parallel processing of large data sets. HadoopMapReduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.

A MapReduce job usually splits the input data-set into independent chunks which are processed by the map tasks in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the reduce tasks. Typically both the input and the output of the job are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.

* **Hadoop Common** — Hadoop Common provides a set of services across libraries and utilities to support the other Hadoop modules.

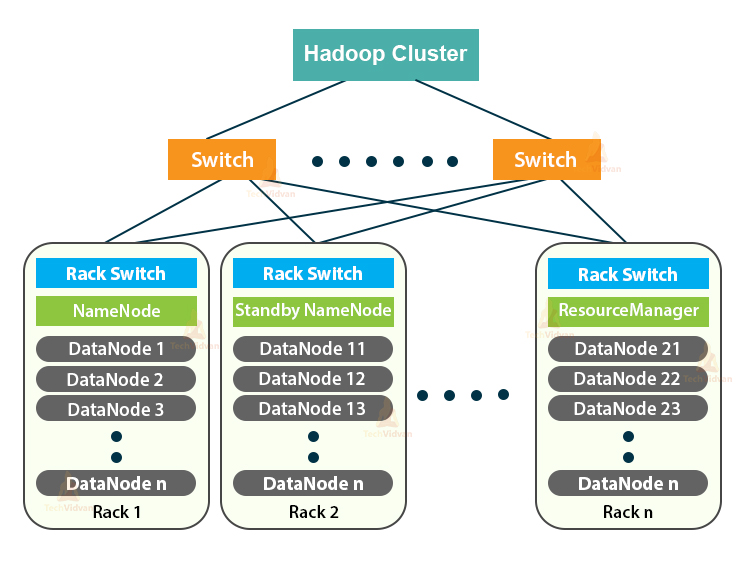
As big data grows exponentially, parallel processing capabilities of a [Hadoop cluster](https://www.dezyre.com/hadoop-wiki#hadoop-cluster) help in increasing the speed of analysis process. However, the processing power of a hadoop cluster might become inadequate with increasing

volume of data. In such scenarios, hadoop clusters can scaled out easily to keep up with speed of analysis by adding extra cluster nodes without having to make modifications to the application logic.

A hadoop cluster architecture consists of a data centre, rack and the node that actually executes the jobs. Data centre consists of the racks and racks consists of nodes. A medium to large cluster consists of a two or three level hadoop cluster architecture that is built with rack mounted servers. Every rack of servers is interconnected through 1 gigabyte of Ethernet (1 GigE). Each rack level switch in a hadoop cluster is connected to a cluster level switch which are in turn connected to other cluster level switches or they uplink to other switching infrastructure.

In a single node hadoop cluster, all the daemons i.e.DataNode, NameNode, TaskTracker and JobTracker run on the same machine/host. In a single node hadoop cluster setup everything runs on a single JVM instance. The hadoop user need not make any configuration settings except for setting the JAVA\_HOME variable. For any single node hadoop cluster setup the default replication factor is one.

In a multi-node hadoop cluster, all the essential daemons are up and run on different machines/hosts.  A multi-node  hadoop cluster setup has a master slave architecture where in one machine acts as a master that runs the NameNode daemon while the other machines acts as slave or worker nodes to run other hadoop daemons. Usually in a multi-node hadoop cluster there are cheaper machines (commodity computers) that run the TaskTracker and DataNode daemons while other services are run on powerful servers. For a multi-node hadoop cluster, machines or computers can be present in any location irrespective of the location of the physical server.



2Fig. 1.2 HADOOP CLUSTER

CHAPTER 2

LITERATURE REVIEW

* 1. **COVISHIELD SENTIMENT ANALYSIS**

Two vaccines are currently being administered in India to prevent the spread of severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). We assessed the humoral immune response after the first dose of two vaccines ChAdOx1-nCOV (Covishield™) and BBV-152 (Covaxin™) in Indian health care workers (HCW).Sentiment Analysis is a challenging research problem especially on social media. Users can freely express their views, opinions and feelings on different trending events, topics, etc. via social media posts. These posts need to be analysed to know what sentiment is conveyed through these posts. Sentiment Analysis, also referred as emotion AI, involves analyzing views from the written text so as to understand and gauge human emotions.

The primary goal of the exploratory data analysis phase of this project was to get acquainted with the columns of the data frame and start brainstorming research questions.

The social media allows world-wide users to connect and interact with each other and express the opinions on general topics. In recent years, the impact of social media websites on daily life has become so considerable that even information on large and small incidents or disasters is gathered via social media sites. The automated extraction of sentiment from these posts and classifying them into different polarities–positive, negative or neutral– has received extensive attention from researchers during the past decade.Twitter is one of the popular social media and boasts of a respectful 255 million active monthly users. Some of the challenges in analysing tweets are: use of informal language, short forms, abbreviations, heavy use of emoticons and slangs. Twitter, also known as

microblogging, has limited size of tweets that makes it difficult to compute the polarity.

Theoretical frameworks in psychology map the relationships between emotions and sentiments. In this paper, we study the role of such mapping for computational emotion detection from text (e.g., social media) with an aim to understand the usefulness of an emotion-rich corpus of documents (e.g., tweets related to covishield) to learn polarity lexicons for sentiment analysis.

The idea of drawing on social media sentiment for satisfaction monitoring aligns well with the broader move towards smart destinations and real-time information processing. Thus, this paper aims to examine whether the electronic word of mouth originating from Twitter posts offers a useful source for assessing destination sentiment for review of Covishield Vaccination.

* 1. **TWITTER AND COVISHIELD TWEETS**

Twitter is a social media platform with more than 500 million users worldwide. It has become a tool for spreading the news, discussing ideas and comments on world events. Twitter is also an important source of health-related information, given the amount of news, opinions and information that is shared by both citizens and official sources. It is a challenge identifying interesting and useful content from large text-streams in different languages, few works have explored languages other than English. It is important to understand public reactions, information dissemination and consensus building in all major forms, including social media in different countries. [11]

Undoubtedly, coronavirus (COVID-19) has caused one of the biggest challenges of all times. The ongoing COVID-19 pandemic has caused more than 150 million infected cases and one million deaths globally as of May 5, 2021. Understanding the sentiment of people expressed in their social media comments can help in monitoring, controlling, and ultimately eradicating the disease. This is

a sensitive matter as the threat of infectious disease significantly affects the way people think and behave in various ways.

Covishield vaccine, which was the first to be approved for use in India has a proven efficacy rate of 70%, which can be scaled up to 91% when both doses are administered 8-12 weeks apart. The vaccine also mounts a higher antibody response and prevents severe outcomes.

* 1. **BIG DATA**

Big Data is the dataset with 3 V's that are Volume, Variety and Velocity and difficult to store and process using traditional database management systems. Big Data Analytics is the way of processing the large amount of data. Hadoop is a popular open-source software which is very useful in analyzing the larger data. Hadoop provides several tools for this purpose like Hive, Pig, Hbase, Cassandra etc. [16]

Advances in information technology and its widespread growth in several areas of business, engineering, medical, and scientific studies are resulting in information/data explosion. Knowledge discovery and decision-making from such rapidly growing voluminous data are a challenging task in terms of data organization and processing, which is an emerging trend known as *big data computing*, a new paradigm that combines large-scale compute, new data-intensive techniques, and mathematical models to build data analytics. Big data computing demands a huge storage and computing for data curation and processing that could be delivered from on-premise or clouds infrastructures.[6]

With the development of Internet of Things (IoT), 5 G, and cloud computing technologies, the amount of data from manufacturing systems has been increasing rapidly. With massive industrial data, achievements beyond expectations have been made in the product design, manufacturing, and maintain process. [12]Many research works deal with big data platforms looking forward to

data science and analytics. These are complex and usually distributed environments, composed of several systems and tools. As expected, there is a need for a closer look at performance issues.[14]

* 1. **HADOOP**

Hadoop is Java based programming framework for distributed storage and processing of large data sets on commodity hardware. It is developed by Apache Software Foundation as open source framework. Hadoop basically has two main components. First one is Hadoop Distributed File System (HDFS) for distributed storage and second part is MapReduce for distributed processing. HDFS is a file system which builds on the existing file system. MapReduce is a programming model which is used for processing and generating large data sets with a parallel, distributed algorithm on a cluster. A MapReduce job generally splits the input data set into independent blocks which are processed by the map tasks in a completely parallel manner. First step is mapping of data set in MapReduce architecture. The framework sorts the outputs of the mapping process, which are then input to the second step is reduce task. Input and the output of the job are stored in a file-system. [3]

In Hadoop, users retain control over how data are being processed by writing their own algorithms in the Map and Reduce interfaces provided. Other issues that require addressing when dealing with parallel executions, such as failure detection, recovery and synchronization between tasks, are handled automatically by the Hadoop framework. In conjunction, the underlying file system supporting the Hadoop framework, the Hadoop Distributed File System, HDFS [2](https://onlinelibrary.wiley.com/doi/10.1002/spe.1082#spe1082-bib-0002) provides the framework with features such as file distribution balancing and file redundancy. The replicated files not only provide data redundancy in the event where one of the nodes fails, it also helps Reduce data transfer when performing workload balancing. In cases where the nodes are busy, new tasks can

be assigned to other nodes which hold the corresponding block's replicated copy, for processing. [5]

* 1. **HDFS**

[HDFS](https://www.sciencedirect.com/topics/computer-science/hadoop) faces several issues when it comes to handling a large number of small files. These issues are well addressed by [archive systems](https://www.sciencedirect.com/topics/engineering/archive-system), which combine small files into larger ones. They use index files to hold relevant information for retrieving a small file content from the big archive file. However, existing archive-based solutions require significant overheads when retrieving a file content since additional processing and I/Os are needed to acquire the [retrieval information](https://www.sciencedirect.com/topics/computer-science/information-retrieval) before accessing the actual file content, therefore, deteriorating the access efficiency.[9]

* 1. **MAPREDUCE**

[MapReduce](https://www.sciencedirect.com/topics/engineering/mapreduce) is an established computing paradigm for processing massive data, in which the input data is viewed as records of key-value pairs. In such a two-phase computation, a first-phase map task processes a portion of the data records to generate or update the key-value pairs; these key-value pairs are then shuffled and supplied to the second-phase reduce tasks, each processes a portion of them, typically with the same key, to produce the final output. [10]

Over the last decade, several advancements have happened in distributed and parallel computing. A lot of data is generated daily from various sources, and this speedy data proliferation led to the development of many more frameworks that are efficient to handle such huge data e.g. - Microsoft Dryad, Apache Hadoop, etc. Apache Hadoop is an open-source application of Google

MapReduce and is getting a lot of attention from various researchers. Proper scheduling of jobs needs to be done for better performance. [2]

[7] analyze the performance impact of JobTracker failure in Hadoop. A JobTracker failure is a serious problem that affects the overall job processing performance.

* 1. **APACHE HIVE**

SQL-on-Hadoop engines such as Hive provide a declarative interface for processing large-scale data over computing frameworks such as [Hadoop](https://www.sciencedirect.com/topics/computer-science/hadoop).The increasing need to process analytical queries over large-scale semi-structured data has led to the development of SQL-on-Hadoop engines. These systems evaluate SQL-like queries over data stored in distributed file systems such as the [Hadoop Distributed File System](https://www.sciencedirect.com/topics/computer-science/hadoop-distributed-file-system) (HDFS) . Hive was the first SQL-on-Hadoop system to provide an SQL-like query language, namely HiveQL, and can use MapReduce or Tez as its underlying framework for executing queries. [8]

CHAPTER 3

REQUIREMENT SPECIFICATIONS

1. **HARDWARE REQUIREMENTS**

* RAM: 8GB
* Operating system: Linux (32bit or 64 bit)
  1. **SOFTWARE REQUIREMENTS**
* **Oracle VM VirtualBox**

Oracle VM VirtualBox is cross-platform virtualization software that allows users to extend their existing computer to run multiple operating systems at the same time. Designed for IT professionals and developers, Oracle VM VirtualBox runs on Microsoft Windows, Mac OS X, Linux, and Oracle Solaris systems and is ideal for testing, developing, demonstrating, and deploying solutions across multiple platforms on one machine.

Oracle VM VirtualBox has been designed to take advantage of the innovations introduced in the x86 hardware platform, and it is lightweight and easy to install and use. Yet under the simple exterior lies an extremely fast and powerful virtualization engine. With a well-earned reputation for speed and agility, Oracle VM VirtualBox contains innovative features to deliver tangible business benefits: excellent performance; a powerful virtualization system; and a wide range of supported guest operating system platforms. VirtualBox is used to setup the virtual Hadoop servers.

Graphical user interface, application, website

Description automatically generated

3Fig. 3.1 ORACLE VIRTUALBOX

* **CentOS 7**

CentOS is a community-driven free software effort that provides two Linux distribution (CentOS Linux and CentOS Stream) and a variety of Special Interest Groups releasing packages to run on those distributions. CentOS Linux provides a free, community-supported computing platform functionally compatible with its upstream source, Red Hat Enterprise Linux (RHEL).CentOS Stream is a continuously delivered distribution that tracks just ahead of RHEL and acts as an upstream for RHEL development.

Logo, company name

Description automatically generated

4Fig. 3.2CENTOS

* **Apache NiFi version-1.9.0**

Apache NiFi supports powerful and scalable directed graphs of data routing, transformation, and system mediation logic. Some of the high-level capabilities and objectives of Apache NiFi include:

1. Web-based user interface

* Seamless experience between design, control, feedback, and monitoring

1. Highly configurable

* Loss tolerant vs guaranteed delivery.
* Low latency vs high throughput
* Dynamic prioritization
* Flow can be modified at runtime.
* Back pressure

1. Data Provenance

* Track dataflow from beginning to end

1. Designed for extension.

* Build your own processors and more
* Enables rapid development and effective testing

1. Secure

* **Apache Hadoop**

Apache Hadoop is an open source, Java-based software platform that manages data processing and storage for big data applications. Hadoop works by distributing large data sets and analytics jobs across nodes in a computing cluster, breaking them down into smaller workloads that can be run in parallel. Hadoop can process structured and unstructured data and scale up reliably from a single server to thousands of machines. Hadoop clusters are gaining popularity for enhancing the speed of data analysis applications. Hadoop clusters are extremely scalable and highly efficient as they are resistant to failures.

Company name

Description automatically generated with low confidence

*5Fig. 3.3APACHE HADOOP*

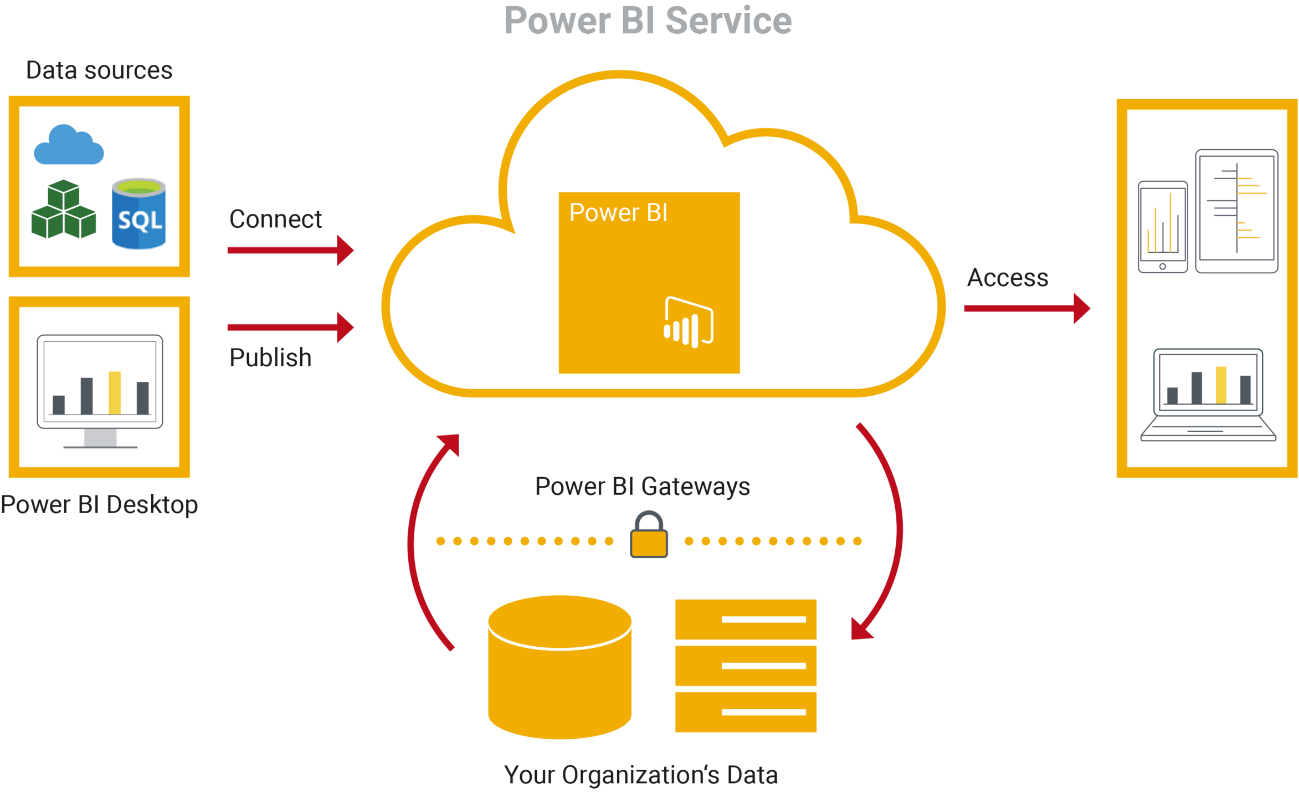
* **Apache Hive**
* Hive is a data warehouse system for Hadoop.
* It allows querying, data analysis utilizing HiveQL etc.
* Hive enables users to portray structure on huge unstructured data.
* Hive can understand organized and unorganized data which may include text files where fields are circumscribed by specific characters.

Graphical user interface, application

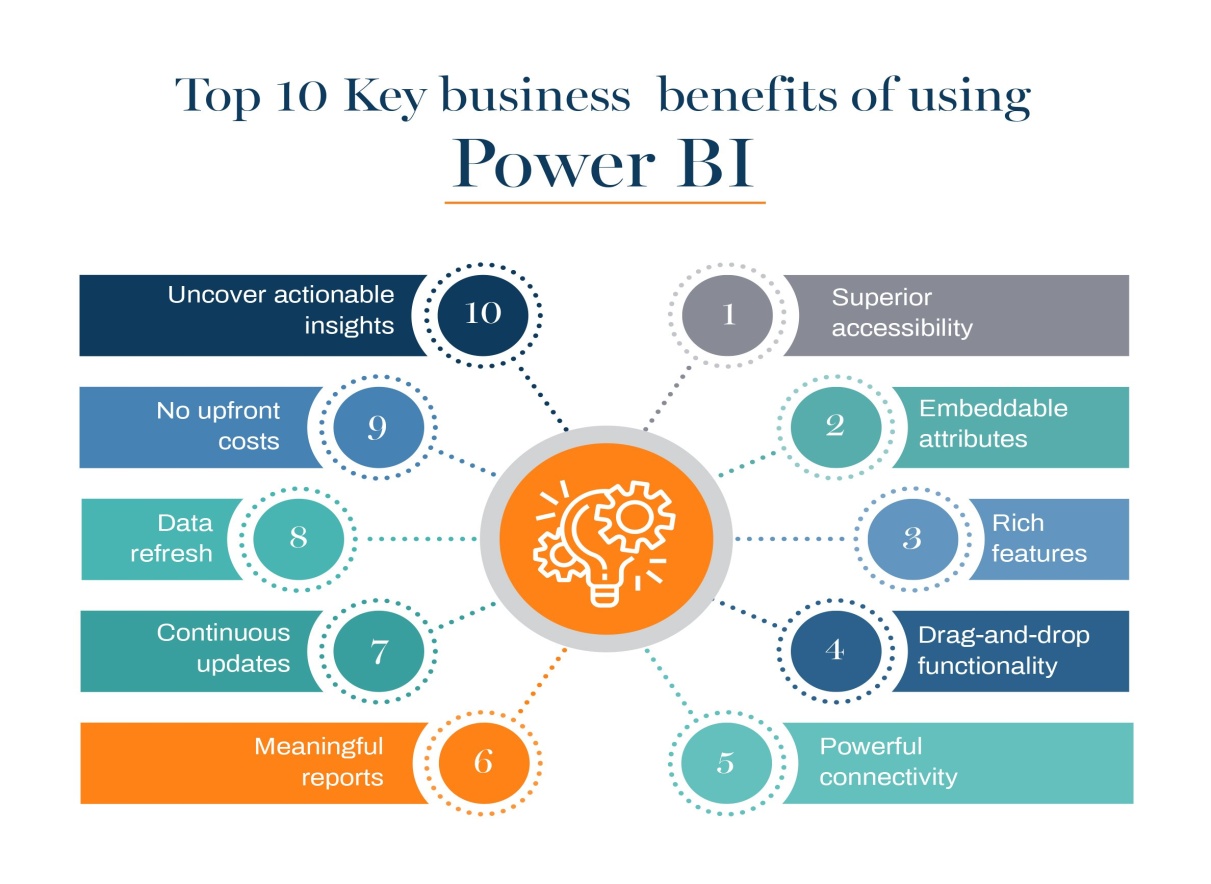
Description automatically generated

6Fig. 3.4HIVE ARCHITECHTURE

* **Putty**
* PuTTY is a free implementation of SSH and Telnet for Windows and Unix platforms, along with an xterm terminal emulator.
* **Microsoft Power BI**
* Microsoft Power BI is a suite that is a collection of business intelligence tools such as software services, apps and data connectors. It is a cloud-
* based platform used to consolidate data from varied sources into a single data set. These data sets are used for data visualization, evaluation, and analysis by making sharable reports, dashboards, and apps. **Power BI** is **Microsoft's** interactive data visualization and analytics tool for **business intelligence** (**BI**). It is used to visualize the final table of hive having count of each of positive, negative and neutral tweets.



7Fig. 3.5POWER BI SERVICE



8Fig. 3.6POWER BI FEATURES/BENEFITS

Graphical user interface, text

Description automatically generated

9Fig. 3.7POWER BI DESKTOP

**3.3 DATA REQUIREMENTS**

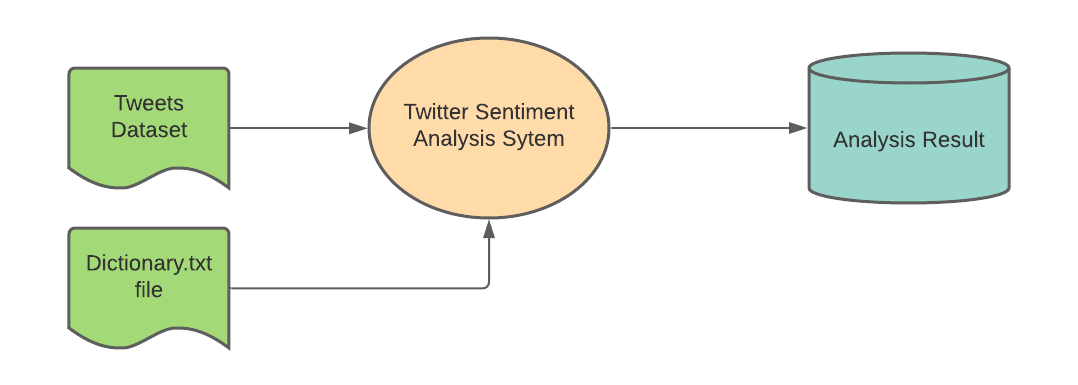
For the analysis purpose we needed a large amount of data as we are using Hadoop which is a tool for analysing Big Data. We found twitter dataset on Kaggle Website. This data set contains 13 columns. All the entries and Tweet’s text in data set are related to Covid19 situation. Kaggle.com is a website that provides dataset for free for its users. Thus, I got dataset for free of cost.

Link: [*https://www.kaggle.com/chaitanyahivlekar/covishield-vaccine-tweets*](https://www.kaggle.com/gpreda/covid19-tweets)

CHAPTER 4

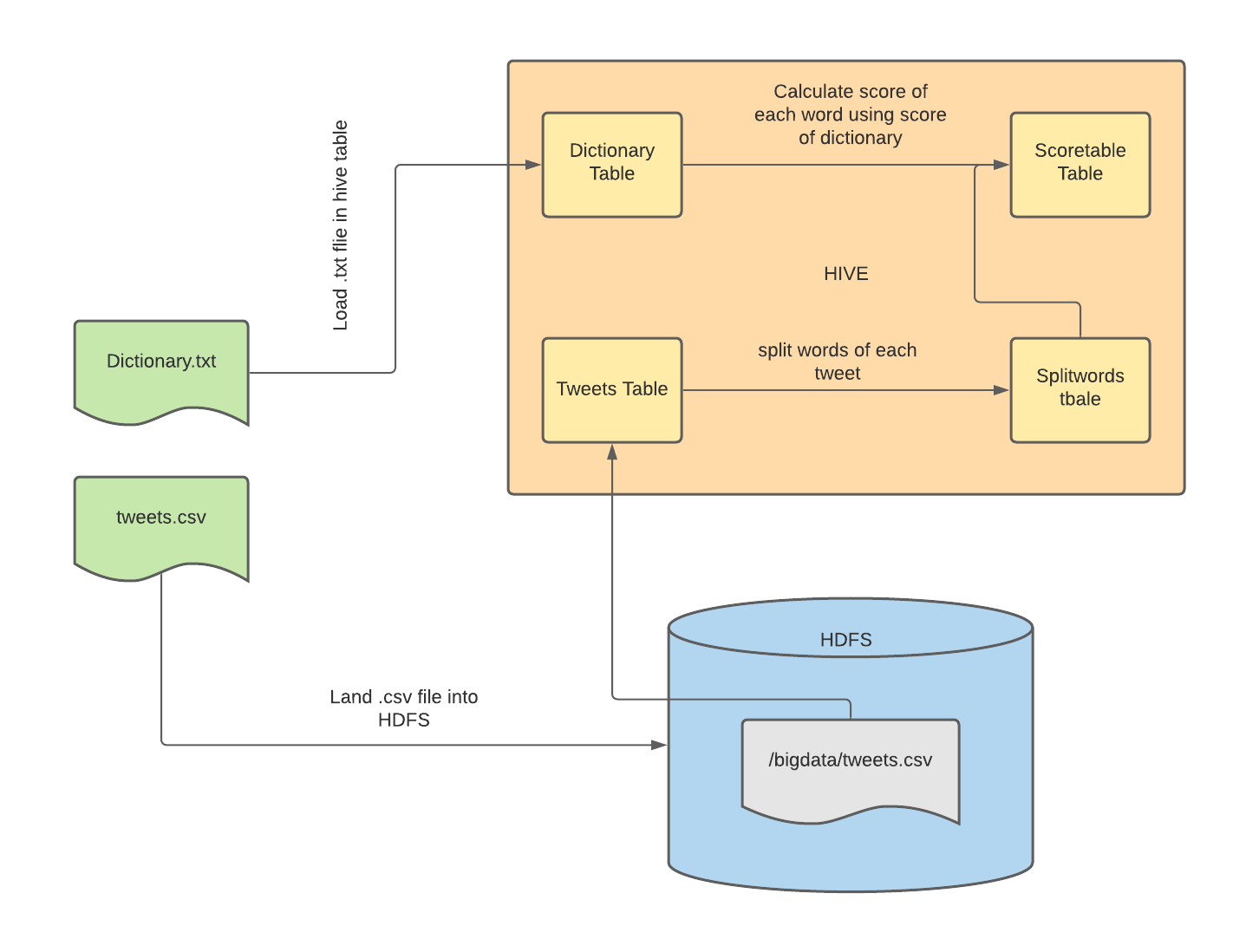
DESIGN

**4.1 0-LEVEL DATA FLOW DIAGRAM**



10Fig. 4.1 0-LEVEL DFD

**4.2 1-LEVEL DATA FLOW DIAGRAM**



11Fig. 4.2 1-LEVEL DFD

* 1. **DATA DICTIONARY**

**4.3.1 Tables Used**

* Covitweets Table stores the actual dataset and it contains all the fields that has been specified in .csv file.

|  |  |
| --- | --- |
| Covitweets | |
| sno | INT |
| time | STRING |
| FromUser | STRING |
| FromUserId | STRING |
| Text | STRING |
| RetweetCount | INT |

**Table 1Table 4.1 Covitweets**

* Dictionary Table contains 2 fields word and score same as in
* dictionary.txt file. Dictionary.txt file has scores associated with each word. The scores are given in such way that, the more the word is positively expressed the higher will be its value and for negative words, more the word is negative more negative value is specified in the score. If the score of any word is 0 then it means that word is Neutral in nature.

|  |  |
| --- | --- |
| Dictionary | |
| word | STRING |
| score | INT |

**Table 2Table 4.2 Dictionary**

* Splitwords Table contains user\_name field and each splitted word of his/her tweet. These split words will help in determining each word’s score independently.

|  |  |
| --- | --- |
| Splitwords | |
| user\_name | STRING |
| word | STRING |

**Table 3Table 4.3 Splitwords**

* Scoretable is generated to analyze each person’s tweet text. It contains the score of each word in tweet corresponding to the word’s value in dictionary table.

|  |  |
| --- | --- |
| Scoretable | |
| t.user\_name | STRING |

|  |  |
| --- | --- |
| t.word | STRING |
| d.score | STRING |

**Table 4Table 4.4 scoretable**

* Covisentiment table is the final table having 2 columns – sentiment and count.

|  |  |
| --- | --- |
| Covisentiment | |
| sentiment | STRING |
| counts | INT |

**Table 5Table 4.5 Covisentiment**

CHAPTER 5

TESTING

* 1. BIG DATA TESTING STRATEGY

Big Data Testing is a testing process of a big data application to ensure that all the functionalities of a big data application work as expected. The goal of big data testing is to make sure that the big data system runs smoothly and error-free while maintaining the performance and security. Since big data is a collection of large datasets that cannot be processed using traditional computing techniques, traditional data testing methods do not apply to big data. This means your big data testing strategy should include big data testing techniques, big data testing methods and big data automation tools, such as Apache’s Hadoop. There are several areas in Big Data where big data testing strategy is required. There are various types of testing in Big Data projects such as Database testing, Infrastructure, and Performance Testing, and Functional testing. Big Data defined as a large volume of data structured or unstructured. Data may exist in any format like flat files, images, videos, etc. The primary Big Data characteristics are three V's - Volume, Velocity, and Variety where volume represents the size of the data collected from various sources like sensors, transactions, velocity described as the speed (handle and process rates) and variety represents the formats of data. Some primary examples of Big Data are Social Networking sites like Twitter and Facebook and E-commerce sites such as Amazon, Flipkart, Snapdeal and any other E-commerce site which have millions of visitors and products.

Big Data Testing plays a vital role in Big Data Systems. If Big Data systems not appropriately tested, then it will affect business, and it will also become tough to understand the error, cause of the failure and where it occurs. Due to which finding the solution for the problem also becomes difficult. If Big Data Testing performed correctly, then it will prevent the wastage of resources in the future.

* 1. HOW BIG DATA TESTING STRATEGY WORKS
* Data Ingestion Testing: In this, data collected from multiple sources such as CSV, sensors, logs, social media, etc. and further, store it into [HDFS](https://www.xenonstack.com/blog/data-serialization-hadoop/). In this testing, the primary motive is to verify that the data adequately extracted and correctly loaded into HDFS or not. Tester must ensure that the data properly ingests according to the defined schema and also have to verify that there is no data corruption. The tester validates the correctness of data by taking some little sample source data, and after ingestion, compares both source data and ingested data with each other. And further, data loaded into HDFS into desired locations.
* **Data Processing Testing:** In this type of testing, the primary focus is on aggregated data. Whenever the ingested data processes, validate whether the business logic is implemented correctly or not. And further, validate it by comparing the output files with input files. **Tools -** Hadoop, [Hive](https://www.xenonstack.com/insights/apache-hive/), Pig, Oozie
* **Data Storage Testing:** The output stored in HDFS or any other warehouse. The tester verifies the output data correctly loaded into the warehouse by comparing the output data with the warehouse data. **Tools -**HDFS, [HBase](https://www.xenonstack.com/insights/apache-hbase/)
* **Data Migration Testing:** Majorly, the need for Data Migration is only when an application moved to a different server or if there is any technology change. So basically, data migration is a process where the entire data of the user migrated from the old system to the new system. Data Migration testing is a process of migration from the old system to the new system with minimal downtime, with no data loss. For smooth migration (elimination defects), it is essential to carry out Data Migration testing.
  1. HOW TO TEST HADOOP APPLICATIONS

Big Data Testing or Hadoop Testing can be broadly divided into three steps.

Step 1 : Data Staging Validation

The first step in this big data testing tutorial is referred as pre-Hadoop stage involves process validation.

* Data from various source like RDBMS, weblogs, social media, etc. should be validated to make sure that correct data is pulled into the system
* Comparing source data with the data pushed into the Hadoop system to make sure they match
* Verify the right data is extracted and loaded into the correct HDFS location

Step 2 : “MapReduce” Validation

The second step is a validation of "MapReduce". In this stage, the Big Data tester verifies the business logic validation on every node and then validating them after running against multiple nodes, ensuring that the

* Map Reduce process works correctly
* Data aggregation or segregation rules are implemented on the data
* Key value pairs are generated
* Validating the data after the Map-Reduce process

Step 3 : Output Validation Phase

The final or third stage of Hadoop testing is the output validation process. The output data files are generated and ready to be moved to an EDW (Enterprise Data Warehouse) or any other system based on the requirement.

Activities in the third stage include

* To check the transformation rules are correctly applied
* To check the data integrity and successful data load into the target system
* To check that there is no data corruption by comparing the target data with the HDFS file system data

Diagram, schematic

Description automatically generated

12Fig. 5.1 BIG DATA TESTING

* 1. BENEFITS OF BIG DATA TESTING STRATEGY
* **Data Validation Testing:** Every organization strives for accurate data for business planning, forecasting and decision-making. This data needs to be validated for its correctness in any big data application. This validation process should confirm that:
* the data injection process is error-free
* complete and correct data is loaded to the big data framework
* the data process validation is working properly based on the designed logic
* the data output in the data access tools is accurate as per the requirement
* **Improved Business Decisions:** Accurate data is the pillar for crucial business decisions. When the right data goes in the hands of genuine people, it becomes a positive feature. It helps in analysing all kinds of risks and only the data that contribute to the decision-making process comes into the picture, and ultimately becomes a great aid to make sound decisions.
* **Cost-Effective Storage:** Behind every big data application, there are multiple machines that are used to store the data injected from different servers into the big data framework. Every data requires storage-and storage doesn't come cheap. That’s why it’s important to thoroughly validate if the injected data is properly stored in different nodes based on the configuration, such as data replication factor and data block size.  
  Keep in mind that any data that is not well structured or in bad shape requires more storage. Once that data is tested and is structured, the less storage it consumes, thus ultimately becoming more cost-effective.
* **Right Data at the Right Time:** Big data framework consists of multiple components. Any component can lead to bad performance in data loading or processing. No matter how accurate the data may be, it is of no use if it is not available at the right time. Applications that undergo load testing with different volumes and varieties of data can quickly process a large amount of data and make the information available when required.
* **Reduces Deficit and Boosts Profits:** Indigent big data becomes a major loophole for the business as it is difficult to determine the cause and location of errors. On the other hand, accurate data improves the overall business, including the decision-making process. Testing such data isolates the useful data from the unstructured or bad data, which will enhance customer services and boost business revenue.
  1. BIG DATA TESTING CHALLENGES

Challenges faced when testing unstructured data are expected, especially when new to implementing tools used in big data scenarios.

* Heterogeneity and Incompleteness of Data
  + **Problem**: Many businesses today are storing exabytes of data in order to conduct daily business. Testers must audit this voluminous data to confirm its accuracy and relevance for the business. Manual testing this level of data, even with hundreds of QA testers, is impossible.
  + **Solution**: Automation in big data is essential to your big data testing strategy. In fact, data automation tools are designed to review the validity of this volume of data. Make sure to assign QA engineers skilled in creating and executing automated tests for big data applications.

### High Scalability

* + **Problem**: A significant increase in workload volume can drastically impact database accessibility, processing and networking for the big data application. Even though big data applications are designed to handle enormous amounts of data, it may not be able to handle immense workload demands.
  + **Solution**: Your data testing methods should include the following testing approaches:
    - **Clustering Techniques**: Distribute large amounts of data equally among all nodes of a cluster. These large data files can then be easily split into different chunks and stored in different nodes of a cluster. By replicating file chunks and storing within different nodes, machine dependency is reduced.
    - **Data Partitioning**: This automation in big data approach is less complex and is easier to execute. Your QA testers can conduct parallelism at the CPU level through data partitioning.

### Test Data Management

* + **Problem:** It is not easy to manage test data when it’s not understood by your QA testers. Tools used in big data scenarios can only carry your team so far when it comes to migrating, processing and storing test data-that is, if your QA team doesn’t understand the components within the big data system.
  + **Solution:**First, your QA team should coordinate with both your marketing and development teams in order to understand data extraction from different resources and data filtering as well as pre and post-processing algorithms. Provide proper training to your QA engineers designated to run test cases through your big data automation tools so that test data is always properly managed.

CHAPTER 6

IMPLEMENTATION & WORKFLOW

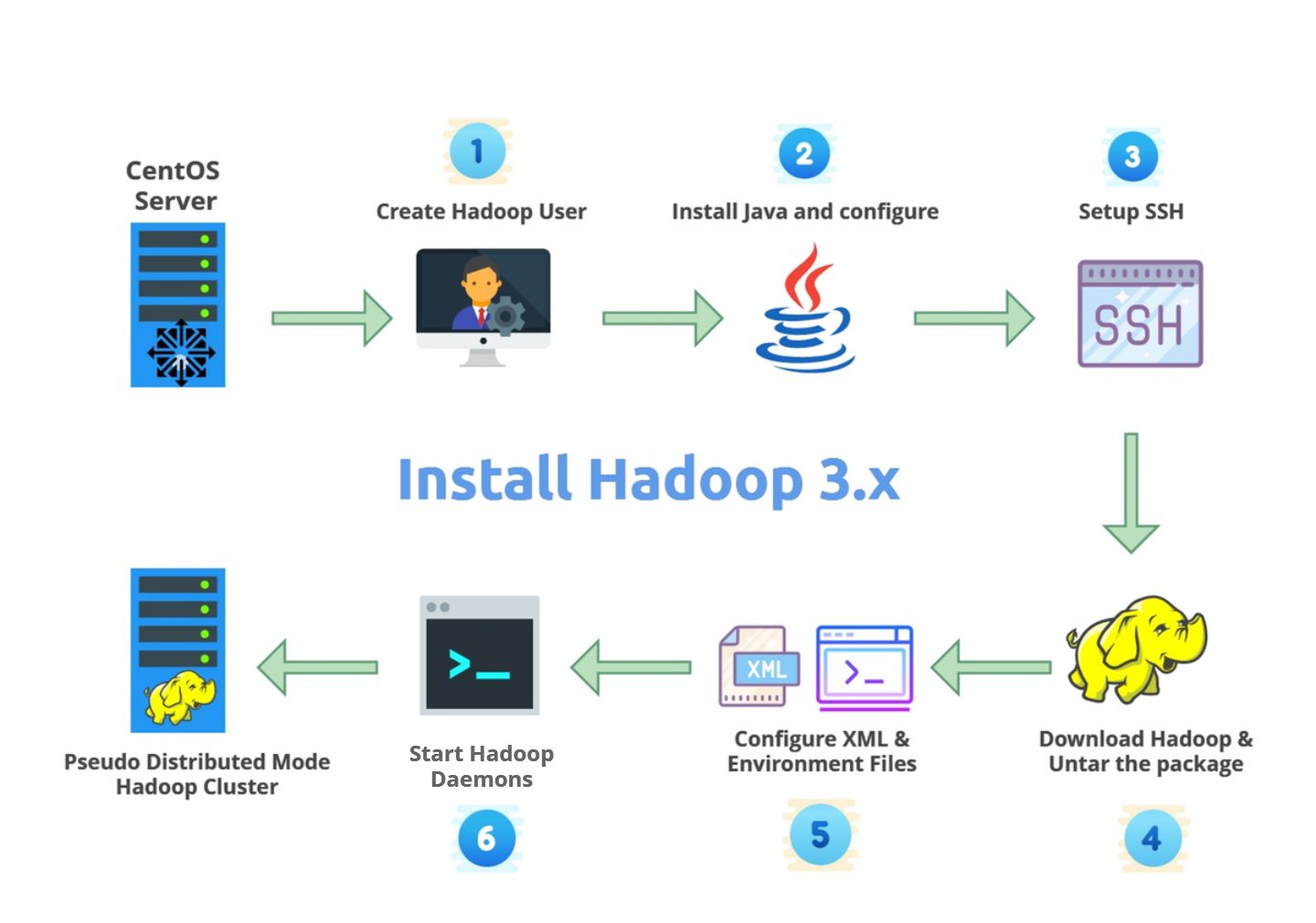
* 1. **LINUX MACHINE SETUP USING ORACLE VIRTUALBOX AND CENTOS**
* Download Oracle VM Virtualbox for windows and install.
* Download CentOS 7
* Setup Linux 64-bit Machine.
  1. **SETTING UP SINGLE NODE CLUSTER**

**Single node cluster** means only one DataNode running and setting up all the NameNode, DataNode, ResourceManager and NodeManager on a single machine.It can easily and efficiently test the sequential workflow in a smaller environment as compared to large environments which contains terabytes of data distributed across hundreds of machines.

Hadoop cluster consists of three components -

* Master Node – Master node in a hadoop cluster is responsible for storing data in HDFS and executing parallel computation the stored data using MapReduce. Master Node has 3 nodes – NameNode, Secondary NameNode and JobTracker. JobTracker monitors the parallel processing of data using MapReduce while the NameNode handles the data storage function with HDFS. NameNode keeps a track of all the information on files (i.e. the metadata on files) such as the access time of the file, which user is accessing a file on current time and which file is saved in which hadoop cluster. The secondary NameNode keeps a backup of the NameNode data.
* Slave/Worker Node- This component in a hadoop cluster is responsible for storing the data and performing computations. Every slave/worker node runs both a TaskTracker and a DataNode service to communicate with the Master node in the cluster. The DataNode service is secondary to the NameNode and the TaskTracker service is secondary to the JobTracker.
* Client Nodes – Client node has hadoop installed with all the required cluster configuration settings and is responsible for loading all the data into the hadoop cluster. Client node submits mapreduce jobs describing on how data needs to be processed and then the output is retrieved by the client node once the job processing is completed.

## **Install Hadoop**



13Fig. 5.1 HADOOP SETUP

### **Step 1:** Download the Java 8 Package. Save this file in your home directory.

### **Step 2:** Extract the Java Tar File.

**Command:** tar -xvf jdk-8u101-linux-i586.tar.gz

**Step 3: Download the Hadoop 2.7.3 Package.**

**Command:** wgethttps://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop-2.7.3.tar.gz

### **Step 4:**Extract the Hadoop tar File.

**Command**: tar -xvf hadoop-2.7.3.tar.gz

### **Step 5:**Add the Hadoop and Java paths in the bash file (.bashrc).

Open**.** **bashrc** file. Now, add Hadoop and Java Path as shown below.

**Command:**  vi .bashrc

Then, save the bash file and close it.

For applying all these changes to the current Terminal, execute the source command.

**Command:** source .bashrc

To make sure that Java and Hadoop have been properly installed on your system and can be accessed through the Terminal, execute the java -version and hadoop version commands.

**Command:**java –version

**Command:**hadoop version

**Step 6:**Edit the [**Hadoop Configuration files**](https://www.edureka.co/blog/explaining-hadoop-configuration/).

***Command:*** cd hadoop-2.7.3/etc/hadoop/

***Command:*** ls

All the Hadoop configuration files are in **hadoop-2.7.3/etc/hadoop** directory.

### **Step 7:**Open core-site.xml and edit the property mentioned below inside configuration tag:

core-site.xml informs Hadoop daemon where NameNode runs in the cluster. It contains configuration settings of Hadoop core such as I/O settings that are common to HDFS &MapReduce.

**Command:** vi core-site.xml

### **Step 8:** Edit hdfs-site.xml and edit the property mentioned below inside configuration tag:

hdfs-site.xml contains configuration settings of HDFS daemons (i.e.NameNode, DataNode, Secondary NameNode). It also includes the replication factor and block size of HDFS.

**Command:** vi hdfs-site.xml

### **Step 9:**Edit the mapred-site.xml file and edit the property mentioned below inside configuration tag:

mapred-site.xml contains configuration settings of MapReduce application like number of JVM that can run in parallel, the size of the mapper and the reducer process, CPU cores available for a process, etc.

In some cases, mapred-site.xml file is not available. So, we have to create the mapred-site.xml file using mapred-site.xml template.

**Command:** cp mapred-site.xml.template mapred-site.xml

**Command:**vi mapred-site.xml.

### **Step 10:** Edit yarn-site.xml and edit the property mentioned below inside configuration tag:

yarn-site.xml contains configuration settings of ResourceManager and NodeManager like application memory management size, the operation needed on program & algorithm, etc.

**Command:** vi yarn-site.xml

**Step 11:**Edit hadoop-env.sh and add the Java Path as mentioned below:

hadoop-env.sh contains the environment variables that are used in the script to run Hadoop like Java home path, etc.

**Command:** vi hadoop–env.sh

### **Step 12:** Go to Hadoop home directory and format the NameNode.

**Command:** cd

**Command:** cd hadoop-2.7.3

**Command:** bin/hadoopnamenode -format

This formats the HDFS via NameNode. This command is only executed for the first time. Formatting the file system means initializing the directory specified by the dfs.name.dir variable.

Never format, up and running Hadoopfilesystem. You will lose all your data stored in the HDFS.

### **Step 13:** Once the NameNode is formatted, go to hadoop-2.7.3/sbin directory and start all the daemons.

**Command:**cd hadoop-2.7.3/sbin

Either you can start all daemons with a single command or do it individually.

***Command:*** ./start-all.sh

The above command is a combination of**start-dfs.sh, start-yarn.sh** & **mr-jobhistory-daemon.sh**

Or you can run all the services individually as below:

### **Start NameNode:**

The NameNode is the centerpiece of an HDFS file system. It keeps the directory tree of all files stored in the HDFS and tracks all the file stored across the cluster.

**Command:**./hadoop-daemon.sh start namenode

### **Start DataNode:**

On startup, a DataNode connects to the Namenode and it responds to the requests from the Namenode for different operations.

**Command:**./hadoop-daemon.sh start datanode

### **Start ResourceManager:**

ResourceManager is the master that arbitrates all the available cluster resources and thus helps in managing the distributed applications running on the YARN system. Its work is to manage each NodeManagers and the each application’s ApplicationMaster.

**Command:**./yarn-daemon.sh start resourcemanager

### **Start NodeManager:**

The NodeManager in each machine framework is the agent which is responsible for managing containers, monitoring their resource usage and reporting the same to the ResourceManager.

**Command:**./yarn-daemon.sh start nodemanager

### **Start JobHistoryServer:**

JobHistoryServer is responsible for servicing all job history related requests from client.

**Command:** ./mr-jobhistory-daemon.sh start historyserver

### **Step 14:** To check that all the Hadoop services are up and running, run the below command.

**Command:**jps

### **Step 15:** Now open the Mozilla browser and go to **localhost**:**50070/dfshealth.html** to check the NameNode interface.

### 2-utilities-browse_directory-the-directories-created.PNG

14Fig. 5.2 HDFS UI WITH DIRECTORIES

* 1. **DATA FLOW FROM LOCAL DISK TO HDFS USING APACHE NIFI**

### Apache Nifi

### Apache NiFi is an open source data ingestion platform. It was developed by NSA and is now being maintained and further development is supported by Apache foundation. It is based on Java, and runs in Jetty server.

### We are using Nifi for ingesting our dataset from local disk into our HDFS. Nifi has a user friendly interface so it becomes very easy to perform these file flows.

### https://documents.lucid.app/documents/7bc934f6-9d8e-4c5d-a052-1eaa2997c6e1/pages/0_0?a=344&x=14&y=39&w=1012&h=462&store=1&accept=image%2F*&auth=LCA%205047a6423c02cf2ba9ece5bb034e9bd56ce26d19-ts%3D1612721249

15Fig. 5.3 NIFI SETUP

* After successful installation of Nifi, open nifi interface inside browser. Here create 2 Processors. First is GetFile. It will fetch our dataset from local disk. Second One is the PutHDFS processor. This processor will fetch dataset from GetFile and ingest dataset into the HDFS directory.

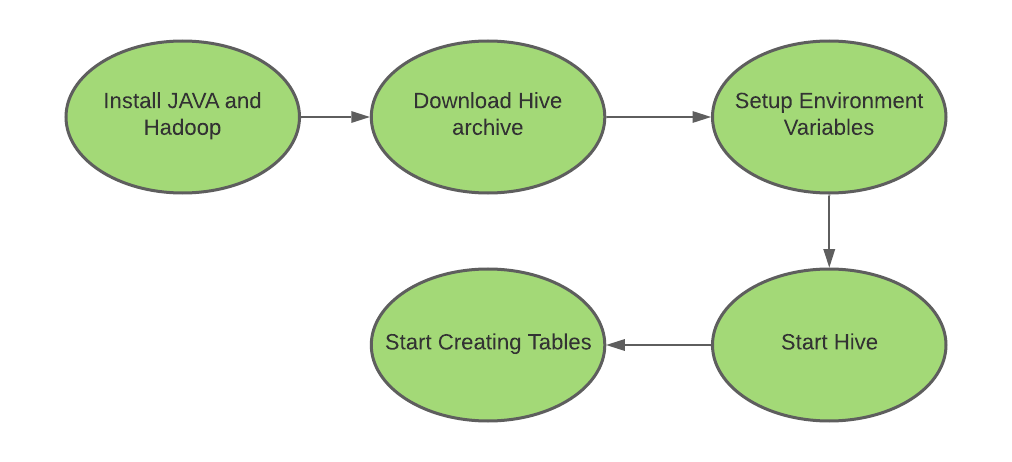
Diagram

Description automatically generated

16Fig. 5.4 FLOW FILES IN NIFI

* 1. **ANALYSIS OF TWEETS USING HIVE**

The Apache Hive data warehouse software facilitates querying and managing large datasets residing in distributed storage. Hive provides a mechanism to project structure onto this data and query the data using an SQL-like language called HiveQL.



17Fig. 5.5 HIVE SETUP

1. Start Hadoop. I have used ./start-all.sh command for starting the Hadoop.

Used jps command to view the running process at present.

**Graphical user interface

Description automatically generated**

18Fig. 5.6 RUNNING NODES

1. We have sample Twitter Data to perform a few basic analysis of the data.

Twitter.csv — Twitter feeds.

* + 1. This .csv file we have stored in HDFS inside directory /covitweets.

Dictionary.txt — Contains words with a rating for each word.

* + 1. Dictionary.txt file is in local disk.

1. To analyze what tweets the user has posted and figure out whether the tweets posted were positive or negative based on the words available in the Dictionary file.
2. Start Hive.

Let’s start creating Tables:

1. Load Dataset from HDFS and into Hive Table.

CREATE EXTERNAL TABLE IF NOT EXISTS covitweets

(sno INT,

time STRING,

FromUser STRING,

FromUserId STRING,

Text STRING,

RetweetCount INT

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION '/dataset';

1. Now, creating a table for dictionary and load data using the given dictionary file

CREATE TABLE dictionary(word string, score int) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';

LOAD DATA local INPATH 'Desktop/Dictionary.txt' into table dictionary;

1. Split up each tweet (called text in the twitter table) into individual words for comparison.

create table splitwords as

select FromUser, word from covitweets raw lateral view explode(split(Text, " ")) text\_ex as word;

select \* from splitwords;

1. Compare the words with the dictionary to get the “scores”.

create table scoretable as

select

t.FromUser,

t.word,

d.score

fromsplitwordst join dictionary d where t.word = d.word;

1. All Schemas and tables can be seen in the following Snapshot of hive shell.

**Graphical user interface

Description automatically generated**

19Fig. 5.7 HIVE TABLES

1. Group the scores again by date, id, user\_name and this is how I have attempted to do that:

SELECT   
user\_name,   
SUM(score),   
 case   
 when sum( score ) > 0 then 'positive'   
 when sum( score ) < 0 then 'negative'   
 else 'neutral'   
 end as sentiment   
 FROM scoretable  
GROUP BY user\_name ;

1. Final output using hive and select query is used to show the name, the tweets whether they are positive or negative. If score is >0 it is positive.

### Text Description automatically generated

20Fig. 5.7 POSITIVE, NEGATIVE AND NEUTRAL TWEETS

1. Insert above results into new table.

INSERT into table covi\_sentiment

select \* from (SELECT

FromUser,

SUM(score),

case

when sum( score ) > 0 then 'positive'

when sum( score ) < 0 then 'negative'

else 'neutral'

end as sentiment

FROM scoretable

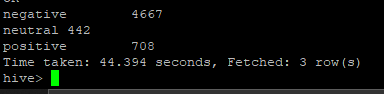
GROUP BY FromUser)

tmp;

1. Display resultof sentiment counts.

SELECT COUNT(sentiment),sentiment FROM covi\_sentiment

GROUP BY sentiment;

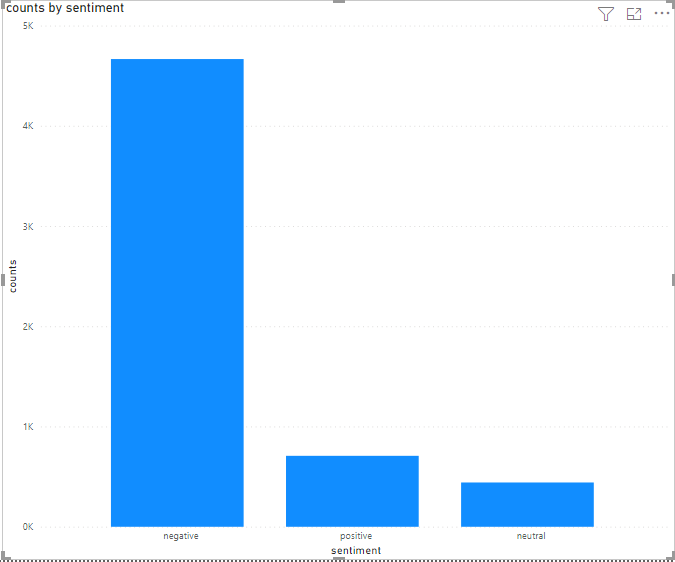


21Fig. 5.8 FINAL RESULT

Load this table as csv file using following command:

*hive -e 'set hive.cli.print.header=true; select \* from final\_count2' | sed 's/[\t]/,/g' > /root/final\_table2.csv*

1. Visualize final\_table.csv using Microsoft Power BI.



22Fig. 5.9 VISUALIZATION

**CHAPTER 7**

CONCLUSION AND FUTURE SCOPE

**7.1 CONCLUSION**

The main agenda of making this project is to understand the concept of Big Data and different tools that are used to handle Big Data. It can fetch the large datasets and after converting and visualizing the data we can easily perform analysis to make predictions and evaluate a person’s opinion according to his tweets.The primary goal of the exploratory data analysis phase of this project was to get acquainted with the columns of the data frame and start brainstorming research questions.

I have learnt toinstall Hadoop and Setting up a Single Node Hadoop Cluster and gained knowledge of data node and name node. The analysis was performed on covishield tweets dataset. Using Dictionary text file, the sentiment analysis is done on tweets dataset and classified tweets into positive, negative, and neutral tweets. For better understanding of the result, visualization is also done on the result in the form of bar graph. I have performed analysis on our datasets and our project is completed using these.

* Nifi
* Hadoop
* Map Reduce
* HDFS
* Hive
* Microsoft Power BI

**7.2 FUTURE SCOPE**

With the rapid development of the Internet, more and more users expressed their views on the Internet. Therefore, the big data of texts are generated on the Internet. In the era of big data, mining the sentiment tendencies contained in massive texts on the Internet through natural language processing technology has become an important way of public opinion supervision.

Sentiment Analysis has been more than just a social analytic tool. It has been an interesting field of study. But it is a field that is still being studied, although not at great lengths due to the intricacy of this analysis. That is this field has functions that are too complicated for machines to understand. The ability to understand sarcasm, hyperbole, positive feelings, or negative feelings has been difficult, for machines that lack feelings. Algorithms have not been able to predict with more than 60% accuracy the feelings portrayed by people. Yet with so many limitations this is one field which is growing at great pace within many industries. Sentiment Analysis using Big Data Analytics can be very helpful in business development and many other fields by analyzing and making predictions for large datasets.

REFRENCES

1. Costa, RogérioLuís de C., José Moreira, Paulo Pintor, Veronica dos Santos, and SérgioLifschitz. "A Survey on Data-driven Performance Tuning for Big Data Analytics Platforms." Big Data Research 25 (2021): 100206.
2. Vanam, Harika. "Analysis of twitter data through big data based sentiment analysis approaches." Materials Today: Proceedings (2021).
3. Tyagi, Kanika, AmertaGhosh, Dipti Nair, KoelDutta, Prakash Singh Bhandari, Irshad Ahmed Ansari, and AnoopMisra. "Breakthrough COVID19 infections after vaccinations in healthcare and other workers in a chronic care medical facility in New Delhi, India." Diabetes & Metabolic Syndrome: Clinical Research & Reviews 15, no. 3 (2021): 1007-1008.
4. Bhuyan, Anoo. "India begins COVID-19 vaccination amid trial allegations." The Lancet 397, no. 10271 (2021): 264.
5. Dutta, Ajitava, RitwikGhosh, Dwaipayan Bhattacharya, SaumenBhat, Adrija Ray, AlakPandit, Shambaditya Das, and SouvikDubey. "Anti-PF4 antibody negative cerebral venous sinus thrombosis without thrombocytopenia following immunization with COVID-19 vaccine in an elderly non-comorbid Indian male, managed with conventional heparin-warfarin based anticoagulation." Diabetes & Metabolic Syndrome 15, no. 4 (2021): 102184.
6. Kalia, Khushboo, and Neeraj Gupta. "Analysis of hadoopMapReduce scheduling in heterogeneous environment." Ain Shams Engineering Journal (2020).
7. Ghazi, MohdRehan, and DurgaprasadGangodkar. "Hadoop, MapReduce and HDFS: a developers perspective." Procedia Computer Science 48 (2015): 45-50.
8. Marichamy, V. Santhana, and V. Natarajan. "Big Data Performance Analysis on a Hadoop Distributed File System Based on Geometric Data Perturbation Technique." Procedia Computer Science 165 (2019): 415-420.
9. Pellakuri, Vidyullatha, P. VenkateswaraRao, Haran Pellakuri, and NikhatParveen. "Big data sentimental analytics on social media using Rhadoop-Hive." Materials Today: Proceedings (2021).
10. Vashishtha, Srishti, and Seba Susan. "Fuzzy rule based unsupervised sentiment analysis from social media posts." Expert Systems with Applications 138 (2019): 112834.
11. Pyne, Saumyadipta, BLS PrakasaRao, and SiddaniBhaskaraRao, eds. "Big data analytics: Methods and applications." Springer, 2016.
12. Emara, Tamer Z., and Joshua Zhexue Huang. "RRPlib: A spark library for representing HDFS blocks as a set of random sample data blocks." Science of Computer Programming 184 (2019): 102301.
13. Sah, Ranjit, Sunil Shrestha, Rachana Mehta, Sohan Kumar Sah, Ali R. Raaban, Kuldeep Dharma, and Alfonso J. Rodriguez-Morales. "AZD1222 (Covishield) vaccination for COVID-19: Experiences, challenges and solutions in Nepal." Travel medicine and infectious disease (2021).
14. Gao, Feng. "Frameworks for Big Data Integration, Warehousing, and Analytics." In Big data application in power systems, pp. 57-73. Elsevier, 2018.
15. Uzunkaya, Can, TolgaEnsari, and Yusuf Kavurucu. "Hadoop ecosystem and its analysis on tweets." Procedia-Social and Behavioral Sciences 195 (2015): 1890-1897.
16. Ghosh, Subhadeep, Subramanian Shankar, Kaustuv Chatterjee, Kaushik Chatterjee, Arun Kumar Yadav, Kapil Pandya, Vani Suryam et al. "COVISHIELD (AZD1222) VaccINe effectiveness among healthcare and frontline Workers of INdian Armed Forces: Interim results of VIN-WIN cohort study." *Medical Journal Armed Forces India* 77 (2021): S264-S270.
17. Mathew, Thomas, and Ms Saji K. John. "COVID-19 vaccine (ChAdOx1 nCoV-19 Corona Virus Vaccine (Recombinant)–COVISHIELD related MS relapse." *Neuroimmunology Reports* (2021): 100006.
18. Srinivasan, K. N., I. Sathyamurthy, and ManodayaNeelagandan. "Relation between COVID-19 vaccination and myocardial infarction–Casual or coincidental?" *IHJ Cardiovascular Case Reports (CVCR)* (2021).
19. Hou, Zhitao, Hongyu Zhang, Haidong Zhang, and Dongmei Zhang. "Visual analytics for software engineering data." In *Perspectives on Data Science for Software Engineering*, pp. 77-80. Morgan Kaufmann, 2016.
20. Bora, Dibya Jyoti. "Big data analytics in healthcare: A critical analysis." In *Big Data Analytics for Intelligent Healthcare Management*, pp. 43-57. Academic Press, 2019.