12-2016

# Data Mining in Social Networks

Usha Singh
sius1401@stcloudstate.edu

**Data Mining in Social Networks**

by

Usha Rani Singh

A Starred Paper

Submitted to the Graduate Faculty of

St. Cloud State University

in Partial Fulfilment of the Requirements

for the Degree of

Master of Science

in Information Assurance

December, 2016

Starred Paper Committee:
Dr. Dennis Guster, Chairperson
Dr. Susantha Herath
Dr. Balasubramanian Kasi

**Abstract**

The objective of the study is to examine the idea of Big Data and its applications in data mining. The data in the universe is expanding step by step every year and turns into large data. These significant data can be determined to utilize a few data mining undertakings. In short, Big Data can be called as an “asset” and data mining is a technique that is employed to give useful results. This paper implements an HACE algorithm that analysis the structure of big data and presents an efficient data mining technique. This framework model incorporates a mixture of information sources, mining techniques, customer interest, security, and data protection system. The study also analyzes and presents the challenges and issues faced in the big data model.

***Keywords:*** *Big data, Data Mining, HACE algorithm, Security*

**Acknowledgements**

The successful fulfillment of this paper could not have been possible without the supervision of my respected professors, Dr. Dennis Guster and Dr. Susantha Herath.

I also would like to thank Dr. Balasubramanian Kasi for being part of the committee and finding the time to read the paper.

# Table of Contents

## List of Tables

## List of Figures

# Chapter I: Introduction

## Introduction

Big Data describes datasets that are so large, complex, or rapidly changing that they push the very limits of our analytical capability (Gurin, 2014). It contains organized and unorganized both sort of data. Data originates from all over the place. Sensors used to accumulate atmosphere data, presents via web-based networking media destinations, advanced pictures, recordings and so on. This data is known as Big Data. The most fundamental challenge for the Big Data applications is to explore the large volumes of data and extract useful information or knowledge for future actions (Rajaraman, & Ullman, 2011). Valuable data gets separated from this Big Data with the assistance of data mining. Data mining is a technique for finding appropriate data and additionally separating unimportant data from vast data scale. In this paper, Nobel data mining techniques and data protection models are discussed and analyzes the challenges and issues faced in the big data model.

The Debates between Democratic Nominee Hillary Clinton and Republican Nominee, Donald Trump in 2016 has generated vast amount of data in search engines like Google, Social Networking Sites like Twitter, Facebook, You-tube, and Instagram. The Huge Data which generated from the tweets through Social Networking Sites and comments in Facebook and You-Tube provides a way to know public opinion before the election and requires proper Data mining techniques to analyze this data. This can one of the best example for showing the importance and challenges of Big Data. Another example would be of the Noble Prize Winner of 2016 Dr. Bob Dylan. Dr. Bob Dylan won Nobel Prize in Literature for the year 2016. This news is apparently the most debatable Nobel Prize of this grouping. Looking on Google with “Bob Dylan Nobel

Prize" achieved a million web pointers on the Internet. Dylan is the first songwriter to win the Noble Prize and the first American to win the Noble Prize for literature after Toni Morrison in 1993. There are many discussions happening on this award given to Bod Dylan and the fact that he did not respond to the award yet. As comments keep proceeding the Internet and in various news media, would there be capacity to diagram an extensive variety of evaluations in different media in a continuous way including overhauled, cross-referenced examinations by reviewer. This example also shows characteristics of Big Data which is diverse, huge, complex, and evolving.

The term 'Big Data' showed up for the first time in 1998 on a Silicon Graphics (SGI) slide deck by John Mashey with the title of Big Data and the NextWave of InfraStress."

The foremost book that was published specifying 'Big Data' was in a book Data Mining by Weiss and Indurkhya in the year 1998. However, the first paper with the words 'Big Data' in the title showed up in a paper by Diebold in 2000. The starting point of the term 'Big Data' is because of the way the gigantic measure big data that is being developed consistently.

The above blueprints exhibit the ascending of Big Data applications where data gathering has ended up being enormous. Despite the way that analysts have authenticated that hypnotizing cases, for instance, transient radio irregularities can be found from the SKA data, existing strategies can just work in an isolates data and are unequipped for managing this Big Data condition consistently. Like this, the vast data volumes require modern methods to analyze and present the data accuracy and fraud prevention, security and protection data.

"Big data" is unavoidable, yet still the idea causes difficulties. Big data has been utilized to pass on a wide range of ideas, including gigantic amounts of data, social media content,

government data administration capacities, ongoing data, and significantly more. Whatever the name, researchers are beginning to comprehend and investigate how to prepare and break down an unlimited cluster of data in new ways.

**Problem Statement**

Designing an efficient data mining technique for Big Data is a challenging task since Big Data is huge, heterogeneous, complex, dynamic and evolving at very fast pace. It is a difficult task for many companies to design an efficient data mining technique which can not only handle the Big Data characteristics but also at same time these mining techniques should protect privacy, security and be economical. Designing such mining technique for Big Data is a challenge.

**Significance of the Study**

The proposed system develops an efficient data mining technique for a social networking site like Twitter whose data is vast, heterogeneous, complex, dynamic and evolving. The data mining technique protects privacy, security and is also economical. The data mining technique is designed in such a way that the client can analyze the data by giving input to the system which might be location based or key-based information. The client can retrieve all the information required by entering any of the keyword.

**Summary**

Chapter I discusses the importance and challenges of data mining in Big Data. Disadvantages in an existing system and the importance of the proposed datamining technique in achieving privacy and security of Big Data in social networking sites are discussed. The research and survey that is done to support implementation of this project is described in the next chapter.

# Chapter II: Background and Review of Literature

## Introduction

The Era of Big Data has arrived (Mervis, 2012). There are many uses and applications by analyzing the Big Data. Big Data plays a major role in making major decisions in almost all sectors like education, information technology, social media, sports, news, politics, weather report and many more. Such is the importance of data mining in Big Data, but still there is much to be done in developing more efficient data mining techniques in terms of handling Big Data characteristics like vastness, complexity, diversity, and dynamic, and, at the same time, the data mining techniques also need to provide privacy, security and needs to economical. In this project, data mining for social networking sites like Twitter is developed. This data mining technique can count the total number tweets in a particular location or total number locations that tweet is tweeted or total number of hash tags for a particular tweet or total discussions a tweet has generated. This data from the data mining technique will be very important to generate reports which are needed in critical decision making. 2.5 quintillion information is generated each year and 90% of the data was produced in last two years (IBM 2012). Data mining in Big Data is not as easy as said or written, there would be many obstacles in the process of design and Implementation.

## Characteristics of Big Data

Figure 1 explains the characteristics of Big Data. Big Data is compared to a giant elephant and our understanding or views on big data is compared with inputs given by blind men. Blind people here are trying to draw the picture of elephant according to the analysis that

he made by measuring the elephant. However, each blind man has different opinions on the elephant. The first one thinks of it as hose; the second blind man thinks it is a tree, a third blind man thinks it a wall, and the last one thinks it a rope. Each one of them is right in their opinion with the measurement they made. However, the output or their perceptions on the giant elephant are entirely wrong. Let us make it more complicated—to better understand Big Data, think the elephant is growing and is changing its position continuously. If the picture is drawn by taking inputs from each blind man (Xindong, Xingquan, Gong-Qing, & Wei, 2014), it would be quite different from original, and this would be one of the examples by which Big Data can be explained. From this explanation, three key features of Big Data came up.

**HACE THEOREM**


*Figure 1*. The Blind Men and Giant Elephant (Miller, 2012)

**Huge data with heterogeneous and diverse dimensionality.** The main features of Big Data are its volume, which is huge, heterogeneous with diverse dimensionalities. Information from different sources is collected and each source has their standards and schema.

For example, take a customer who has the account in bank. A single customer can have different accounts, and each customer is different from regarding benefits that he gets from a

bank, even when receiving information from the customer, each product information will be stored in various environment and to store all this information under ID, data needs to be retrieved from all these source.

**Autonomous source with distributed and decentralized control.** Individual information source with separate and distributed controls are also one of the key features of Big Data applications. When the information sources are different, each will collect the information without depending on the other or any grouped or common control. As an example, consider the World Wide Web. Every server generates information without depending on any standard control and each server can work properly without depending on other servers. The huge data also make the systems prone to attacks, if the system depends on a standard control unit. Major applications like Facebook, YouTube, and Twitter have servers spread across the world which depend on other servers. So, an independent source with shared and decentralized control makes sure that applications respond faster and serve the customers without any hurdles.

**Complex and evolving relationships.** As the capacity of Big Data increases, so will the complexity and connections under data. As the data and complexity increases, data should be connected by finding similarities. Social networking sites like Facebook, Twitter can be taken as an example. People are grouped based on their similarities. If the Facebook page is observed, it is designed in such a way that the advertisements on the page is displayed based on our likes. Few more examples would be Twitter, LinkedIn, and so forth.

## Big Data Processing Model

**Conceptual view of big data processing framework.** Figure 2 displays a conceptual view of the Big Data processing framework, which includes three layers from inside out with

thoughts on data retrieving and analyzing (Layer1), data secrecy and domain knowledge (Layer2), and Big Data mining algorithms (Layer3).

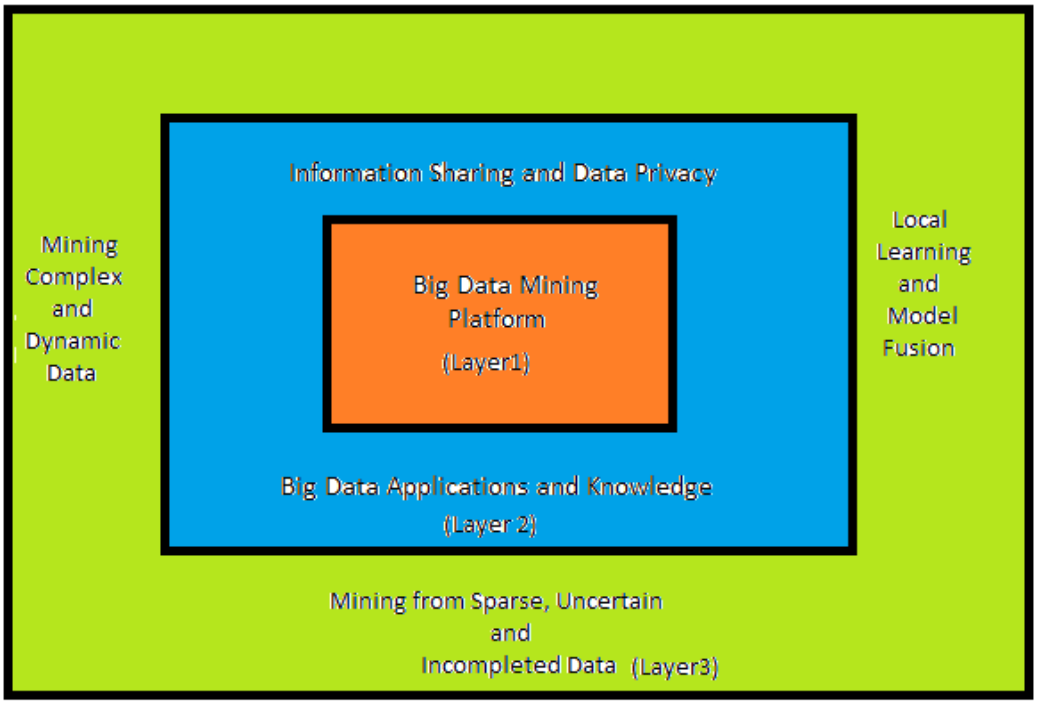



*Figure 2*. Big Data Processing Model (Xindong et al., 2014)

**Big data mining platform (Layer1).** The challenge at Layer1 concentrates on data retrieving and computing methods. As Big Data volume increases continuously and are stored in different places, an efficient computing method is needed. Many data mining algorithms need data to be loaded into the main memory, even if there is a super large main memory, moving data from one place to other makes it expensive.

**Big data semantics and application knowledge (Layer2).** The Challenge at Layer2 focuses on semantics and field knowledge for various Big Data applications. Such data can provide more benefits to the mining process (Layer1) and mining algorithms (Layer3). For example, depending on different field applications, the data privacy and data sharing

mechanisms between data producers and information consumers can be significantly different. Sharing sensor system data for applications like water quality monitoring is encouraged but sharing mobile users' location is not acceptable for majority applications. Another example is a social network; users are connected and share dependency structures.

**Algorithms design (Layer3).** At Layer3, for data mining challenges focuses is on algorithm designs to tackle the difficulties raised by the huge data volumes, distributed data, and by complicated and changing data characteristics. The circle at Layer3 contains three steps. First, inadequate, heterogeneous, vague, unfinished, and multi-source data are preprocessed by data fusion methods. Second complicated and changing data is mined after pre-processing. Thirdly, the information that is obtained by local learning and model fusion is examined, and appropriate data is supplied back to the pre-processing stage. Then the model and parameters are fixed according to the feedback. In the entire process, data sharing is not only a commitment of stable improvement at each layer but also a goal of Big Data processing.

## Big Data Costs and Numbers

Table 1

*Big Data Costs and Numbers*

| Year | Data Storage Capacity | Cost |
|---|---|---|
| 1980s | 10 MB | 3398$ |
| 2000 | 20 GB | 20$ |
| 2014 | 1 TB | 85$ or less |

(whatsthebigdata.com, 2014)

The ability to store data has multiplied every 40 months since the 1980s. Over the years, the storage costs reduced, but the data accumulation increased and lead to network overload.

*Figure 3*. Changes in Storage Space Over Time (pingdom, 2010)

Total data—the exact amount of data—is unknown because the volume of data is growing so fast that it is becoming difficult to keep track of the data. According to computer giant IBM, 2.5 Exabyte's—that is 2.5 billion gigabytes (GB)—of information is produced every day in 2012. That is big by anyone's standards. About 75% of data is unstructured, coming from sources such as text, voice, and video (BBC News, 2014).

Data storage for the primary computers came with memories marked in kilobytes, but the latest smartphones can now store 128GB and many laptops now have one terabyte (1,000GB) hard drives as standard. Storage is not an issue anymore.

*Figure 4*. Range International Information Hub (Computer Business Review)

Range International Information Group located in Langfang, China is the largest data center—as of November 2016—6.3 Million Sq. Ft and equivalent to nearly 110 football pitches. Total investment for this is 1 billion. It is collaborated with IBM to provide services to Chinese government services and private companies.

**Data on Big Data**

Table 2

*Data on Big Data*

| Social Media | Quantity of Data Generated |
|---|---|
| Email messages per day | >144.8 billion a day |
| Twitter | >340 million a day |
| Facebook | Share >684,000 bits of content a day |
| YouTube | Upload 72 hours of new video |
| Web shopping | Consumer spend 272,000$ a day |
| Google | Receives 2 million search queries a day |
| Apple | Receives around 47,000 app downloads a minute |
| Instagram | Photographers share 3,600 new photos a minute |
| Flicker | Photographers upload 3,125 new photos a minute. |
| WordPress | bloggers publish close to 350 new blog posts a minute |
| Mobile Web | Receives 217 new participants a minute. |
| Walmart | 25 Petabytes of data flow through Walmart transactional databases. |
| Big Data companies | 631 Number of enterprise big data companies expected to exist in 2017 (IBISWorld) |
| People working on Big Data | 340,000 projected number of people expected to work on Big Data in 2018. |

(Cukier, 2010)

**Data Mining**

First, what data mining exactly is should be understood. Data mining is "The practice of automatically searching large stores of (Big) data to discover patterns and trends that go beyond

simple analysis." This is done using mathematical algorithms to segment the data and evaluate the probability of future events or future data. What this all means is that Data Mining is the act of sifting through huge amounts of data, using sorting algorithms to make the amount of data sift through smaller, to identify connections and correlations within the data. The reason that Data Mining is featured in our report and report is that it features heavily in the interaction between Big Data.

Data mining has had a relatively bad reputation over the years and so had gained a series of pseudonyms that are less than complementary—Data Dredging, Data Fishing, Data Snooping. The official term for Data Mining, however, is the analysis phase of the Knowledge Discovery in Data (KDD) process. The term, Data Mining, is something of a misnomer as the goal of data mining is the extraction of patterns and correlations from within big data, not the extraction of data itself. These extracted patterns are, in turn, interpreted to identify trends and patterns that existed within the data unknowingly.

**Data Mining Types**

The key to understanding the function of Data Mining is to understand that it is a task-oriented process. It handles six classifications of tasks, all designed to extract the data to solve particular problems or questions.

The first of these tasks is Anomaly Detection. The best way of summing this task up is the saying "One of these things is not like the others . . ." It is a way of identifying erroneous data or behaviors in the data to be singled out and examined later by a human operator. Examples of this include statistical plot analysis, to identify erroneous data, or looking for an expected average of sales, month by month, for business, ignoring the unusual and unlikely

repeat sales. This can also be used for something as mundane as identifying if someone is cheating on paper or essay, by identifying the numbers of recognizable patterns that exactly match large contents found on the internet or other papers and essays (Karypis & Ahmed, 2012).

The second task of data mining is Association Rule Learning. This looks for correlations between data that might otherwise not be identified based upon one or more common points of data. For mundane users, this becomes most influential in their life when suggested items or products are indicated at check-out or in coupons. It also comes into effect through targeted marketing like political flyers and mailed advertisements. The idea is that if one or more points of data are shared, the likelihood of more points of data being shared is increased.

The third task of data mining is Clustering. This is used to identify shared commonality within data. It is used to find correlations between data that is otherwise not connected. From this extracted data, the association rule, above, could be performed to create more targeted data, or it could be looking for correlations that were unidentified before. A word of warning must go hand in hand with this task because correlation does not mean causation. Correlation DOES exist between increased youth gang violence and deaths and the sales of ice cream. This was identified by a clustering result requested by Kemps in 2007. This has become one of the classic examples used to show that correlation does not equate to causation.

The fourth task of data mining is Classification. In our day to day lives, this becomes most useful in winnowing out spam emails from our inbox. To describe it best, it is the means of sorting unknown data into known classes through identifying known traits and patterns. As a rule, this should not be subject-based search except in the broadest terms. The reason for this, is

that person defies logic based grouping, and this technique has been accused of stereotyping individuals.

The fifth task of data mining is Regression. This is how the programmer can feed the desired correlation or conclusion into the data mining program, and the program will identify the optimum pathway through the data to agree with that correlation or conclusion. Although this is a powerful tool, it can sometimes be used to defy majority data to create a pattern that does not truly exist. To give an example, if there is a case of the majority of individuals who commit violence, who own either a MacBook Air or an IBM ThinkPad, are owners of MacBook Air, a less than honorable IBM ThinkPad retailer could use Regression to prove that there exists a correlation between MacBook Air buyers and violence. Since this is not causal, there is no real connection between the two, but there is a proven correlation, and the wrong conclusions could be drawn from it.

The sixth and final task of Data Mining is Summarization. This is one of the most useful forms of data mining if the big data in question is mostly unknown. If the user is looking for statistical data on the United States Census, the summarization task would be of the most use. The reason for this is that there is no need for a connection between data, or further information based on known connections, or even looking for a conclusion based on the data. The sole output looked for is statistical data, or a summary of what the data tells us. For the sake of pure statistical, raw, summed up, data, summarization can be a powerful tool.

**Data Mining Uses/Applications**

Because of the types of data mining, it is used for a wide variety of applications. Data mining can be used in electronic gaming, economic predictions, CDC usage for disease

clustering, targeted advertising/politics, electronic searches for relevant data, classifying music by genre and finding new music based upon current listening trends, finding a job based upon resume, or by an employer finding the right candidate the opposite way. It can be used by the government to identify potential terrorist threats, human rights violations, illegal shipments, and criminal activities. It can be used by social media sites to help find friends, and to identify trending news stories based upon stories read in the past. The farmer's Almanac, the long-range weather forecast, the expectations of earthquakes, and a diagnosis of what is wrong based upon symptoms through an online clinic. It can be used to track spending trends to identify stocks that should increase in value. It can be used in any number of useful, day-to-day ways that impact each of us.

What is Subject-Based Data Mining? This is the method of searching for associations between individuals in data. This can be useful for identifying possible terrorist cell members, cult members, gang members, criminal associates, and more by identifying numerous connections between individuals. The goal, ultimately, is to determine which persons, financial transactions, movements, etc. are related to a known initiating datum. As mentioned above, this can lead to grouping non-related subjects to another subject erroneously because of shared or similar habits, movements, traits, and patterns. This is known, in legal terms as stereotyping (Tutorials point, n.d.).

**Data Mining Process**

Much like the waterfall programming cycle, Data Mining follows a very similar lifecycle pattern. It has six phases—Business Understanding, Data Understanding, Data Preparations, Modeling, Evaluation, and Deployment. Business Understanding and Data Understanding Phases

correlate with the Requirement/Analysis phase. In data mining, there is a lot of back and forth between these two phases before system moves on. The reasons for that is the need to understand the data that is being worked upon, and the understanding that that data is constantly being redefined by current and previous data mining efforts.

The Data Preparations phase and Modeling Phase would correlate with the System Design phase and the Implementation phase. In Data Mining, there is a great amount of interaction between Modeling and Data Preparation. The reason for this is that the modeling is dependent upon the Data. Every time the data changes, the Modeling must change with it. As Data Mining is performed on working and non-working databases, this requires that the modeling for the data mining process be equally fluid until it is run. Equally important is that, if the data mining is already occurring, the data fed into it, even the new data being collected must be of a nature that the data mining algorithm can identify and work with it.

The Evaluation phase would be similar to Testing/Verification, except that, sometimes, the extracted data from the data mining is impossible to interpret in any timely fashion. This is usually done as a means of testing the initial output of the program before it is run too long, creating erroneous data connections/extracting worthless data. If there is any problem with the criteria of that Data mining, it is back to square one, into the Business Understanding/Data Understanding phases.

The Deployment Phase for them is, simply put, the deployment of the program. The big difference between the two processes is that there is no maintenance phase in the Data Mining Cycle after Deployment without running through the whole cycle again. One of the big reasons for this is, like mentioned before, the understanding of the data is constantly evolving and

changing. Because of this evolution in understating, the business understanding (goal of the data mining) is changing and evolving, too.

**Online Social Networking**

Online social systems are broadly used to associate with our companions' circle. For instance, Facebook, Instagram, and Twitter inspire clients to stay in contact with their contacts, reconnect with old partners, and produce the new contacts with suggested clients in perspective of shared parts, for instance, bunches, recreation exercises, interests, and covers in cooperation circles. From the past couple of years, colossal addition in the usage of online social systems can be seen, with around 300 online social systems frameworks gather the information from more than a far-reaching part of a billion selected clients. In like manner, online social systems put away a tremendous extent of maybe fragile moreover, private information on customers and their affiliations. This information is ordinarily private and anticipated the general population are watching our gathered data. Another way, the reputation of online social systems keeps in steadfast customers as well as social occasions with rather adversarial interests as well. The upgrading and trouble of plan and use illustrations of online social systems unavoidably exhibit security interruption to all online social systems customers because of information exchange and openly sharing on the Internet. It is inside these lines not surprising that stories about security separated by Facebook and MySpace show on and on in standard media (Sandeep & Dhawas, 2016).

The Internet produces the different sort of data sharing frameworks, for example, the Web. In present days, online social systems expanded the solid prevalence and are these days turned into the one of most prevalent destinations on the Web. Online sharing system have been

got enormous development as of late, for example, Facebook, YouTube, Twitter, and LinkedIn, which is creates gigantic measure of social information which includes individual and private information about every client. Security and protection are primary key parameters of online social system which create some confinement on destinations. Be that as it may, clients need to make new companions to grow their social associations and in addition to get the data from a large number of individuals. Widely in the later past Companion proposal is a vital application in numerous online social systems. Online social systems are sorted out around users dislike the Web, which is, to a great extent, composed around content. Social system investigations have upgrade the exploration in creating different calculations.

Different specialists centered from various software engineering disciplines have endeavored to settle a portion of the issues that emerge in online social systems and propose an alternate scope of security arrangements, including programming devices and plan standards. Some old procedures need to ID-base suggestion, propose a friend by giving back a twofold reply, “yes” or “no,” which cut down the likelihood of finding companions of companions. Informal organizations are searching for; the larger part of this kind of strategies will disregard to open up kinship more than two trusts. The homophily idea, online social system clients have social connection with each other in little of their indistinguishable characteristics. From one point of view, particularly asking recommendations to untouchables or a non-dear companion reveals Alice’s character, and additionally uncovers her wellbeing condition and medicinal information. Shockingly more repulsive, customary proposal approaches advertising identity to prescribe to outcasts will demonstrates online social systems clients, social associations with the group, which prevent patients from utilizing it, besides diminished the shot of mastermind the

multi-jump trust chain on the off chance that one of online social system clients on the chain gives back a wrong outcome. On the other hand, current philosophies cannot manage the cost of the fine-grained and context aware comes about consequently, considering the way that online social network clients call for to choose the recommended mates in light of their own choices on the suggestion ask. As in our example, Alice would need to demand help from her associates who work in a healing center, yet not a truck driver. To overcome on this issue, an outline framework is designed which thinks about how conceivable it is of utilizing online social system users' social characteristic to implement the multi-bounce trust chain within the sight of every Context aware 1-bounce trust relationship, where the larger piece of trust associations are framed and reinforced by the common social trait.

Investigations of the principal well known social networking site, Friendster (Boyd, 2006) depict how individuals make their profile with the expectation of conveying news about themselves to others. Boyd (2006), utilizing an ethnographic approach, uncovers the likelihood of unintended outcomes. As in other social networking sites, Friendster individuals make a profile and make open connections to others. Imagine a scenario where there is something about companion's page that may bring about humiliation if saw outside of any relevant connection to the subject at hand. Although individuals can control what shows up on their profile, they cannot control what shows up on a companion's profile. Unrefined pictures on a companion's profile brought on sympathy toward an educator when her understudies asked to "companion" her. This put her in a clumsy position in light of the fact that permitting understudies access to her profile would likewise permit them to view her companions, who she knew to have racy pictures on their profile (Boyd, 2004). This episode shows that worries raised by exploring issues of security

and trust were obvious in the primary academic articles on social networking sites. Facebook, a social networking site that started with an attention on schools and colleges, yet now incorporates secondary schools and different associations, has been considered by (Gross & Acquisti, 2006, Lampe, Ellison, & Steinfield, 2007). These reviews have gathered profile data from Facebook using a web crawler, and through studies of individuals. They demonstrate that Facebook individuals uncover a great deal of data about themselves, and are not extremely mindful of protection choices or who can really see their profile (Gross & Acquisti, 2006).

Participation in social networking sites has significantly expanded as of late. Services such as Twitter, Instagram, or Facebook allow millions of people to create online profiles and share personal data with large systems of friends—and, regularly to large number of strangers. Data in social networks like Twitter and Facebook is an important resource for online users, researchers, and marketing companies. Users input has rendered in very rapid increase in online big data and which in turn becomes potential helpful data. Big data on social network has been valuable resources to online users, researchers, and marketing organizations. Input from users has contributed to the quick growth of online big data and in return the big data becomes potentially more useful. Web-based social networking has turned into a stage to serve enormous information administration, for example, client correspondence or data trade in a particular online group. Notwithstanding, to make a multi-reason interpersonal organization to be useful, concerns must be tended to with respect to information security, client protection, client and information dependability, and in addition how to advance client investment. While information security has been a consistent research subject for a considerable length of time, how to ensure client protection while guarantee client and information dependability in the meantime is still a

major test both practically speaking and in scholarly research on these interpersonal organizations. The capacity in holding clients is additionally one of the key variables to the accomplishment of informal communities. Different interpersonal organization stages, for example, Facebook and LinkedIn, have been effective in increasing significant ubiquity among interests by giving administrations, i.e., social associations and expert associations separately (Gao & Iwane, 2015).

In this area, a non-specific system model is exhibited as shown in figure 5, which can serve as social network stage. Utilizing this stage, a virtual group can be framed where clients participate in the exercises namelessly or by utilizing alias names. Be that as it may, this virtual group is not confined. It is associated with genuine elements which hold clients' actual characters and can in this way support clients' traits on the virtual group. Along these lines, because clients are mysterious in the social group, their embraced traits are reliable. Thus, clients or outsider specialists and advertisers can put trust on the focused on clients for data trade, research, or study in view of those trusted qualities. Clients can put trust on the stage with respect to their security on taking an interest in the virtual group exercises. This multi-trust informal organization display incorporates the accompanying segments.

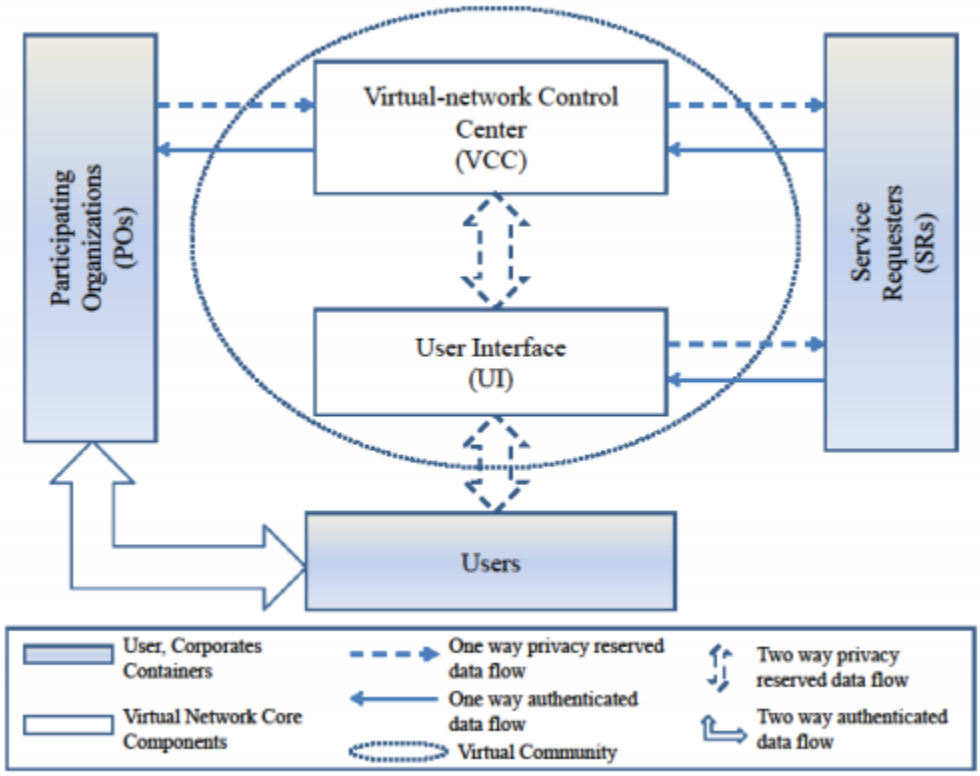



*Figure 5*. Generic Multi-Trust Social Network Model (Gao & Iwane, 2015)

VCC keeps up the focal database for alias records and trust control. The User Interface (UI) is to give a stage to the clients to initiate their virtual account on VCC and enlist on the virtual system, and in addition for the clients to recover data, collaborate with others, and partake in the exercises asked for by SRs, or for the SRs to demand benefits on VCC. POs are the trusted associations which hold legitimate client recognizable pieces of proof and authority client information. With a specific end goal to join the system and empower their substantial clients to take an interest, POs first make remarkable numbers (UNs) and transitory pass codes for their clients. POs then exchange the UNs and pass codes alongside trust-essential qualities TA (a1, a2, … , ai) to VCC. TAs (a1, a2, … , ai) are the fundamental credits to arrange clients to fill the need of the virtual system, yet utilizing these characteristics alone can't uncover clients' actual

characters. Clients then utilize their UN and pass code to actuate and enroll on VCC. While enrolling, clients will make their login name and secret word. They may likewise make their alias virtual profile as indicated by their inclination. Once enrolled, clients can specifically distribute their TAs. Fractional or entirety TA (a1, a2, … , ai) will get to be distinctly distributed properties PA (v1, v2, … , vj), which will be uncovered to open for different clients to distinguish comparative clients. Login names are one of a kind on VCC. Clients can change their login name, watchword, and other profile data whenever. Login names won't be uncovered, while pen names virtual profiles are freely shared. SRs may ask for review, backers or business benefits on VCC, focusing on a few particular clients in light of clients' TA (a1, a2, … , ai), which must be upgraded by POs, and ordered by VCC. VCC is dependable to give the focus on gatherings to SRs and give the alternatives to clients to decide on support. Clients won't uncover their genuine personalities by participating in such exercises. VCC is dependable to oversee accounting of clients' support exercises by utilizing UNs. The administration requesters may give motivating force prizes, for example, fiscal attributes or blessing cards to members. VCC is capable to pass these prizes to the clients through the UN guarantors, i.e., the journalist POs.

SRs may ask for overview, promoters or business benefits on VCC, focusing on a few particular clients in light of clients' TA (a1, a2, … , ai), which must be upgraded by POs, and classified by VCC. VCC is capable to focus on gatherings to SRs and give the alternatives to clients to select support. Clients will uncover their genuine personalities by partaking in such exercises. VCC is capable to oversee accounting of clients' interest exercises by utilizing UNs. The administration requesters may give impetus prizes, for example, money related attributes or

blessing cards to members. VCC is capable to pass these prizes to the clients through the UN backers, i.e., the reporter POs.

In the model, multi-trust is occupied with terms of information security requirement, client protection control, member believability control, and compensating component. The principle worry on information security is with respect to communication channels among the system segments. This could be upheld through open key encryption so that the partaking groups and clients get appropriate confirmation and verification when vital. Client protection is completely enforced through various access controls. In the first place, clients need to accept so that their unidentifiable features might be exchanged to VCC from POs. Second, clients need to initiate their records on VCC and specifically distribute their qualities. Third, clients make their virtual profile uninhibitedly and may roll out any improvements at whatever time. Fourth, only POs can interface genuine client characters through UNs. Client validity is key for SRs and different clients to believe the members and related information input. This is tended to by requiring that members must be legitimate clients in POs, and the clients' TA (a1, a2, … , ai) must be redesigned through POs. Besides, clients can be evaluated by different clients in light of their cooperation's. Clients' change of pen names not influence their evaluations. Thusly, a client's dependability can be kept steady. To energize clients' cooperation's, motivating forces might be essential. A dependable reward instrument is through VCC's accounting of client support and reverse data passing by means of UNs to comparing POs. SRs pass the compensates through VCC to POs. The POs will dispense the reward to the genuine clients through true channels (Sandeep, & Dhawas, 2016).

**Data Mining Algorithms**

Dynamic system is considered as robust mechanism to appear and address the transient changes of the data of various underlying complicated systems (Karypis & Ahmed, 2012). Gigantic bits of learning concerning the steady social cases among the substances can be grabbed by analyzing the short-lived progression of the unusual component relations.

This can perceive the moves beginning with one spared state then onto the following and may offer confirmation to the nearness of external components that are accountable for changing the steady social cases in these frameworks. This paper demonstrates another data mining methodology that explores the time-constant relations or states between the components of the dynamic structures and gets all maximal non-abundance progression methods for the steady social states. Exploratory results, in light of different datasets from genuine applications, exhibit that the strategy is gainful and adaptable.

Web crawlers are vital to various web applications, for example, web searchers, web reports, and web lists, which keep up web pages in their adjacent documents (Lee, Ha, & Alam, 2012). In this document, a plan is proposed for crawling computations for convincing and capable crawl asking for by sorting out basic pages with the outstanding PageRank as the hugeness metric.

Remembering the true objective to score URLs, the proposed computations use distinctive components, including deficient association structure, between host joins, page titles, and subject significance. Broad scale test is coordinated by using straightforwardly open data sets to take a gander at the effect of each component on crawl asking for and evaluate the execution of various computations. The test occurs affirm the feasibility of our arrangements. In

particular, differentiated and the specialist Rank Mass crawler, the FPR-title-host estimation decreases computational overhead by a variable as unbelievable as three in running time while improving feasibility by 5% in total PageRank.

Perceiving social effect in frameworks is fundamental to perceiving how behavior spread (Walker & Aral, 2012). Strategy is displayed that is used in vivo randomized experimentation to perceive effect and shortcoming in frameworks while avoiding the inclinations natural in standard assessments of social contamination. Estimation in an agent trial of 1.3 million Facebook customers exhibited that more energetic customers are more frail to affect than more settled customers, men are more convincing than women, women sway men more than they affect different women, and married individuals are the base unprotected to affect the decision to get the thing publicized. Examination of effect and shortcoming together with framework structure revealed that convincing individuals are less helpless to affect than non-influential individuals and that they amass in the system while defenseless individuals do not, which recommends that capable people with enticing sidekicks may be instrumental in the spread of this thing in the system.

A large amount of measurable data about individuals, e.g., demographic information, web activity, energy use, correspondence illustrations and community interactions are being accumulated and inspected by various national real workplaces, audit affiliations, remedial centers, and Web and individual to individual correspondence associations (Reiter & Machanavajjhala, 2012).

Wide disseminating of microdata (data at the granularity of individuals) supports impels in science and open technique, helps subjects to get some answers concerning their social

requests, and enables understudies to make aptitudes at data examination. Routinely, regardless, data producers cannot release microdata as accumulated, in light of the fact that doing, all things considered, could reveal data subjects' identities or estimations of unstable properties. Fail to guarantee arrangement (when ensured) is deceitful and can make hurt data subjects and the data supplier. It even may be unlawful, especially in government and investigation settings.

For example, if one reveals mystery data secured by the United States Private Information Protection and Statistical Efficiency Act, an individual is subject to a most extraordinary of $250,000 in fines and a five-year prison term.

The prompt increase in opportunity and command of interpersonal and behavior rich sources, for example, sites and other web organizing avenues, rising open entryways and challenges develop as people now can, and do, adequately use computational learning to look out and fathom the sentiments of others (Agarwal & Banerjee, 2012).

The examination of the total behavior of individuals has recommendations to business understanding, insightful examination, customer relationship organization, and taking a gander at the total online action as appeared by changed burst swarms, the Arab Spring and other such events.

In this article, a nature-breathed life is familiarized into theory with total model behavior from the watched data on web diaries using swarm information, where the goal is to exactly exhibit and anticipate the future behavior of a considerable people resulting to viewing their joint efforts in the midst of an arrangement stage. Specifically, an underground bug state improvement model is set up with the behavioral example from the online diary data and is attempted over genuine sites. Promising results were gained in example estimate using underground bug region

based pheromone classier and CHI real measure. Accurate principles are given in selecting proper parameters for the model, complete up with captivating recognitions, and envision future investigation course**.**

**Java Overview**

Java is the programming language and it keeps running on the stage also called as platform called the Java Virtual Machine (JVM). The Java programming language is a high-level language that can be described by the greater part of the accompanying trendy expressions. Java can be understood easily, if the basics of Object Oriented Programming are known which makes it simple. Java can be expanded as it is dependent on the Object model. The application that is developed on one operating system can run on any other operating system, then that application is called Platform independent. Java is platform independent but JVM is platform dependent. Simple, Architecture neutral, Object oriented, Portable, Distributed, High performance, Interpreted, Multithreaded, Robust, Dynamic, Secure.

With maximum programming languages, either compile or interpret a program so that it can run on the computer. The Java programming language is unique in that a program is both compiled and interpreted. The compiler first translates a program into an intermediate language called Java bytecode- the platform-independent codes understood by the interpreter on the Java platform. The interpreter parses and runs each bytecode instruction on the computer. Compilation occurs just once; interpretation occurs each time the program runs.

Java bytecodes are the machine code instructions for the Java Virtual Machine. Every Java interpreter, whether it is a development tool or Web program that can run applets, is an execution of the Java VM. Java bytecodes help make “compose once, run anyplace” feasible.

The program can be compiled into bytecodes on any platform that has a Java compiler. The byte codes can then be keep running on any usage of the Java VM. That implies that as long as a computer has a Java VM, the same program written in the Java programming language can work on Windows 2000, a Solaris workstation, or on an iMac.

**The Java platform**. A platform is the equipment or programming environment in which a system runs. The most prominent platforms like Windows 2000, Linux, Solaris, and MacOS are already know. Most platforms can be depicted as a mix of the working system and equipment. The Java platform contrasts from most different platforms in that it is a product just platform that keeps running on top of other equipment based platforms.

The Java platform has two segments which are the java virtual machine and the java application programming interface (java API). Java VM has been acquainted by now. It is the base for the Java platform and is ported onto different equipment based platforms. The Java API is an extensive gathering of instant programming parts that give numerous helpful abilities, for example, graphical UI (GUI) gadgets. The Java API is assembled into libraries of related classes and interfaces; these libraries are known as packages. The following segment, What Can Java Technology Do? Highlights what usefulness a portion of the packages in the Java API give.
A Native code will be code that after it is assembled, the ordered code keeps running on a particular equipment platform. As a platform-autonomous environment, the Java platform can be a bit slower than local code. Nonetheless, savvy compilers, very much tuned mediators, and in the nick of time bytecode compilers can convey execution near that of local code without undermining convertibility.

**What can Java technology do**? The most widely recognized sorts of projects written in the Java programming language are applets and applications. Surfing the net one might be familiar with applets. An applet is a project that sticks to specific traditions that permit it to keep running inside a Java-empowered program.

Nevertheless, the Java programming language is not only to write charming, engaging applets for the Web. The universally useful, high-level state Java programming language is additionally a capable programming platform. Utilizing the library API, diverse kinds of projects can be created.

An application is a standalone program that works specifically on the Java platform. An exceptional sort of application known as a server serves and backings customers on a system. Samples of servers are Web servers, proxy servers, mail servers, and print servers. Another specific application is a servlet. A servlet can practically be considered as an applet that keeps running on the server side. Java Servlets are a famous decision for building intuitive web applications, replacing the utilization of CGI scripts. Servlets are similar to applets in that they are runtime enlargements of applications. Rather than working in programs, however, servlets keep running inside Java Web servers, arranging or fitting the server.

How does the API help all these kinds of programs? It does so with packages of software elements that gives a wide range of functionality. Every full usage of the Java platform gives the following features:

**The essentials:** Objects, strings, numbers, info and yield, information structures, system properties, date and time, and so forth.

**Applets:** The arrangement of traditions utilized by applets. A Java applet is a little application which is composed in Java and conveyed to clients as bytecode. The client dispatches the Java applet from a page, and the applet is then executed inside a Java Virtual Machine (JVM) in a procedure separate from the web program itself. A Java applet can show up on the edge of the website page, another application window, Sun's Applet Viewer, or a stand-alone instrument for testing applets. Java applets were presented in the principal variant of the Java language, which was discharged in 1995.

**Internationalization:** Help for composing programs that can be restricted for users around the world. Projects can naturally adjust to particular areas and be showed in the fitting language.

**Security:** Both low-level and high-level state, including electronic marks, Public and private key administration, access control, and documents.

**Software components:** Known as JavaBeans, can connect to existing component models.

**Object serialization:** Permits lightweight resolution and correspondence via Remote Method Invocation (RMI).

**Java Database Connectivity (JDBCTM):** Gives uniform access to a wide range of relational databases.

The Java platform additionally has APIs for 2D and 3D illustrations, openness, servers, cooperation, communication, discourse, liveliness, and that's just the beginning.

**In what manner will Java Technology Change My Life?** Distinction, fortune, or even an occupation cannot be guaranteed, even if Java programming language is used. Still, it is prone

to improve projects and requires less exertion than different languages. Java innovation can be trusted to help to do the accompanying.

**Get started quickly:** Despite the Java programming language is a powerful object-oriented language, it is easy to learn, especially for programmers already accustomed with C or C++.

**Write less code:** Comparisons of project measurements (class checks, technique numbers, et cetera) propose that a system written in the Java programming language can be four times littler than the same project in C++.

**Compose better code:** The Java programming language enables awesome coding practices, and its garbage gathering helps dodge memory spills. Its article presentation, its JavaBeans part building, and it is enormous, adequately extendible API helps reuse other people's attempted code and present fewer bugs.

**Avoid platform dependencies with 100% Pure Java:** Programs can portable by avoiding the use of libraries written in other languages. The 100% Pure JavaTM Product Certification Program has a repository of historical process manuals, white papers, brochures, and similar materials online.

**Write once, run anywhere:** Because 100% Pure Java programs are compiled into machine-independent bytecodes, they operate consistently on any Java platform.

**Distribute software more easily:** applets can be easily updated from the central server. Applets take advantage of the feature of allowing new classes to be loaded "on the fly," without recompiling the entire program.

## ODBC and JDBC

**Open database connectivity (ODBC).** Microsoft Open Database Connectivity (ODBC) is an official programming interface for application designers and database systems suppliers. Before ODBC turned into an actual standard for Windows projects to interface with database systems, software engineers needed to utilize restrictive languages for every database they needed to associate. Presently, ODBC has settled on the decision of the database system verging on insignificant from a coding point of view, which is as it ought to be. Application engineers have a great deal more essential things to stress over than the sentence structure that is expected to port their project starting with one database then onto the next when business needs abruptly change.

Through the ODBC Administrator in Control Panel, the specific database can be indicated that is connected to a data source that an ODBC application system is designed to use. Remember about an ODBC information source as an entryway with a name on it. Each entryway will lead to a specific database. For instance, the information source named Sales Figures may be a SQL Server database, though the Accounts Payable information source could allude to an Access database. The physical database alluded to by a data source can live anyplace on the LAN.

The ODBC system documents are not introduced into system by Windows 95. Alternatively, maybe, they are introduced when a different database application is set up, for example, SQL Server Client or Visual Basic 4.0. At the point when the ODBC symbol is introduced in Control Panel, it utilizes a document called ODBCINST.DLL. It is likewise conceivable to control ODBC information sources through a stand-alone program called

ODBCADM.EXE. There are a 16-bit and a 32-bit variant of this system, and each keeps up a different rundown of ODBC information sources.

From a programming viewpoint, the magnificence of ODBC is that the application can be composed to utilize the same arrangement of capacity calls to interface with any information source, paying little heed to the database merchant. The source code of the application does not change whether it converses with Oracle or SQL Server. These two are specified for instance. There are ODBC drivers available for a few dozen well-known database systems. Indeed, even Excel spreadsheets and plain content records can be transformed into information sources. The working system utilizes the Registry data composed of ODBC Administrator to figure out which low-level ODBC drivers are expected to converse with the information source, (for example, the interface to Oracle or SQL Server). The stacking of the ODBC drivers is straightforward to the ODBC application program. In a customer/server environment, the ODBC API even handles a large portion of the system issues for the application software engineer.

The main usefulness of this plan is numerous to the point that are probably thinking there must be some catch. The main detriment of ODBC is that it is not as proficient as talking specifically to the local database interface. ODBC has had various spoilers make the charge that it is too moderate. Microsoft has dependably asserted that the basic element in execution is the nature of the driver programming that is employed. The accessibility of good ODBC drivers has enhanced an incredible arrangement as of late. What’s more, at any rate, the feedback about execution is to some degree closely resembling the individuals who said that compilers could never coordinate the velocity of immaculate low-level computing construct. Possibly not, but

rather the compiler (or ODBC) gives the chance to compose cleaner programs, which implies to complete sooner. In the interim, PCs get speedier consistently.

**Java database connectivity (JDBC).** With an end goal to set a free database standard API for Java; Sun Microsystems created Java Database Connectivity or JDBC. JDBC offers a universal SQL database access system that gives a uniform interface to an assortment of RDBMSs. This steady interface is accomplished using "module" database availability modules, or drivers. On the other hand, if a database seller wishes to have JDBC provider, he or she should give the driver to every platform that the database and Java keep run on.

**JDBC goals**. Few programming packages are outlined without objectives as a primary concern. JDBC is one that, because of its many objectives, drove the advancement of the API. These objectives, in conjunction with early commentator input, have settled the JDBC class library into a strong system for building database applications in Java. The objectives that were set for JDBC are critical. They will give some knowledge concerning why certain classes and functionalities carry on the way they do. The outlined objectives for JDBC are as per the following:

***SQL Level API***. The originators felt that their fundamental objective was to characterize a SQL interface for Java. Despite the fact that not the most minimal database interface level conceivable, it is at a sufficiently low level for more elevated level tools and APIs to be produced. Alternately, it is at a sufficiently high level for application software engineers to utilize it unhesitatingly. Achieving this objective takes into account future tool vendors to "produce" JDBC code and to conceal a large number of JDBC's complexities from the end client

***SQL Conformance***. SQL sentence structure differs as there is shift from database seller to database merchant. With an end goal to support a wide category of sellers, JDBC will permit any inquiry explanation to be passed through it to the fundamental database driver. This permits the availability module to handle non-standard functionality in a way that is appropriate for its clients.

***JDBC must be implemental on top of common database interfaces***. The JDBC SQL API must "sit" on top of other regular SQL level APIs. This objective permits JDBC to utilize existing ODBC level drivers by the utilization of a product interface. This interface would make an interpretation of JDBC calls to ODBC and the other way around.

***Provide a Java interface that is consistent with the rest of the Java system***. Because of Java's acceptance in the user community thus far, the architects feel that they should not stray from the current design of the core Java system.

***Keep it simple***. This objective likely shows up in all product outline objective postings. JDBC is no different. Sun felt that the configuration of JDBC ought to be exceptionally straightforward, taking into consideration one and only technique for finishing an assignment for each mechanism. Permitting duplicate functionality just serves to mislead the clients of the API.

***Use strong, static typing wherever possible***. Solid writing takes into account more error checking to be done at compile time; likewise, less blunder shows up at runtime.

***Keep the common cases simple***. Because more often than not, the usual SQL calls used by the programmer are simple SELECT's, INSERT's, DELETE's and UPDATE's, these

queries should be simple to perform with JDBC. However, more complex SQL statements should also be possible.

**Tomcat 6.0 Web Server**

Tomcat is an open source web server developed by Apache Group (Apache Tomcat 6.0, 2016). Apache Tomcat is the servlet container that is employed in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies. The Java Servlet and JavaServer Pages specifications are developed by Sun under the Java Community Process. Web Servers like Apache Tomcat provide only web components while an application server provides web components as well as business components (BEAs Weblogic, is one of the popular application servers). To develop a web application with JSP/servlet install any web server like JRun, Tomcat, etc. to run application.

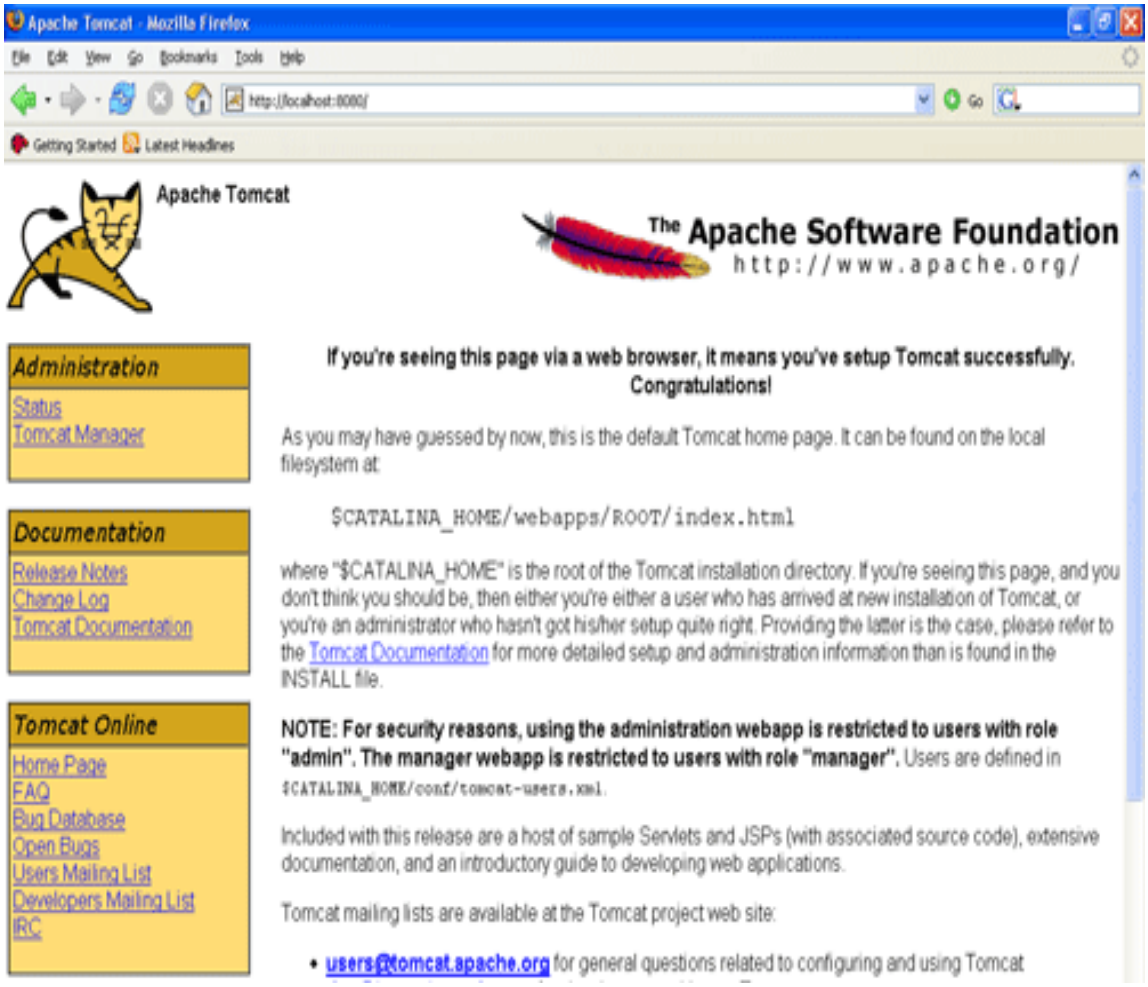



*Figure 6*. Screen Shot of Apache Tomcat Home Page (Apache Tomcat 6.0, 2016)

**Summary**

In this section, characteristics of Big Data are described through HACE THEOREM, Architecture of Big Data, some facts and figures about Big Data are described. In the second part of the section, datamining, types of datamining, applications of data mining, online social networking and few data mining algorithms which are helpful in the implementation of this project are discussed.

# Chapter III: System Analysis

## Existing System

Big Data is growly at a very fast pace and cannot be managed by usually used software to store or to process the data with reasonable speed. Challenges in existing systems include difficulty in developing a data mining technique which is economical and, at the same time, protect security and maintain data privacy. The main challenge in designing an efficient data mining technique is huge data with heterogeneous and diverse dimensionalities. Different data collectors use different schema for collecting the data and type of application also results in diverse presentation of data. For example, results of a CT scan of a person can be shown only in videos and images to doctors so that detailed examinations can be carried out. For DNA, microarray pictures are used to present the genetic code of data as this is the way our present techniques capture the information. The above two examples show how data can be heterogeneous (different types of presentation of data for same individual) and diverse dimensionality of data becomes major task when aggregating data from different sources.

Autonomous sources with distributed and decentralized access are other main feature of Big Data applications. As it is autonomous, it generates and collects information without any centralized data source. World Wide (WWW) can be compared with each server provides information and works independently without relying on any other web server. The advantage of this approach is the application will be less vulnerable to attacks as distributed volumes of data. The major Big Data applications can be considered like Google, Twitter, and Facebook servers are distributed across the world for efficient services and quick responses. Complex and evolving data is the other characteristic of Big Data, which is a challenge in designing an efficient data

mining technique. As the volume of data increases, the complexity of data and relationship between the data increases. Datamining technique that is designed should also be able handle this characteristic of Big Data which is complex and dynamic which means the data will be changing moment to moment. Datamining technique that is designed should provide information sharing (Howe et al., 2008) and at the same time maintain privacy and security (Huberman 2012).

**Proposed System**

The study offers an HACE algorithm to analyze the Big Data structure. The characteristics of HACE algorithm make it an astounding test for discovering significant gaining from the Big Data. The HACE theory speculation suggests that the key aspects of the Big Data are (a) enormous with heterogeneous and diverse data sources, (b) self-administering with appropriated and decentralized control, and (c) complicated and evolving networks.

The analysis of Big Data applications where data gathering has created tremendously and is past the limit of customarily used programming mechanical assemblies to get, regulate, and handle inside a “tolerable snuck past time.” The most significant test for Big Data applications is to examine the massive volumes of data and concentrate accommodating information or learning for future exercises. The noteworthy data volumes require a reasonable data examination and desire stage to fulfill quick response and consistent gathering of data.

In this study, the most relevant and most accurate social sensing feedback is provided to better understand our society at real-time. It is also very efficient and provides information sharing with privacy and security. Privacy of data is protected by restricting access to the information by adding encryption certificates or controlled access to data by providing access to sensitive information only to few people who needs it. The second way the data can be provided

privacy is by anonymizing the fields such that it can pinpoint to one record (Cormode & Srivastava, 2009). The task to achieve the data privacy by using above approaches can be obtained by carefully designing a secured/encrypted certification or access mechanisms so that confidential information can be protected from prohibited users or the users who do not have access to the information. Data anonymization can be introducing some random values into the data depending on the degree of privacy needed. Advantage of data anonymization is that data can be shared across any platform without the need of any controlled access. Data mining design which is developed should be able to predict the data of future, even if the prediction is just slightly better than random guess, it brings huge values to the designers (Bughin, Chui, & Manyika, 2010). For this to achieve the application and domain knowledge is needed.

**System Feasibility Study**

The practicability of the project is analyzed in this phase, and the business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis, the feasibility study of the proposed system is to be carried out. This is to assure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Figure 7 shows the three main key considerations involved in the feasibility analysis.

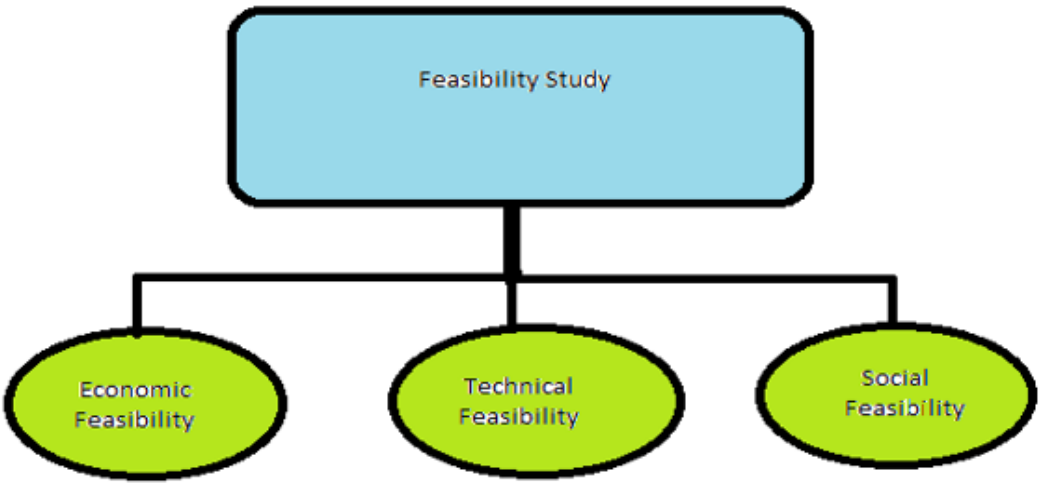



*Figure 7*. Feasibility Study

**Economic feasibility.** This research is carried out to check the financial consequence that the system will have on the company. The cost of capital that the company can pour into the research and expansion of the system is limited. The expenses must be supported. Thus, the developed system is within the funds, and this is achieved because most of the technologies used are freely available. Only the customized products had to be bought.

**Technical feasibility**. This study is carried out to check the technical practicability, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources and will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are needed for implementing this system.

**Social feasibility.** The phase of the study is to check the level of approval of the system by the client. This involves the method of preparing the user to use the system efficiently. The client must not feel scared by the method, rather must admit it as a need. The level of approval

by the clients only depends on the techniques that are applied to train the user about the operation and to make him easy with it. His level of belief must be increased so that he is also able to make some valuable analysis, which is embraced, as he is the final client of the system.

**System Architecture**

The Figure 8 illustrates the architecture of the application that is implemented in this project. The client wants to analyze the social networking site (which is similar to Twitter is taken as example), to make some important decisions by which the customer satisfaction is improved. Application is designed such that client can retrieve information that he needs for making decisions from the application that is implemented.

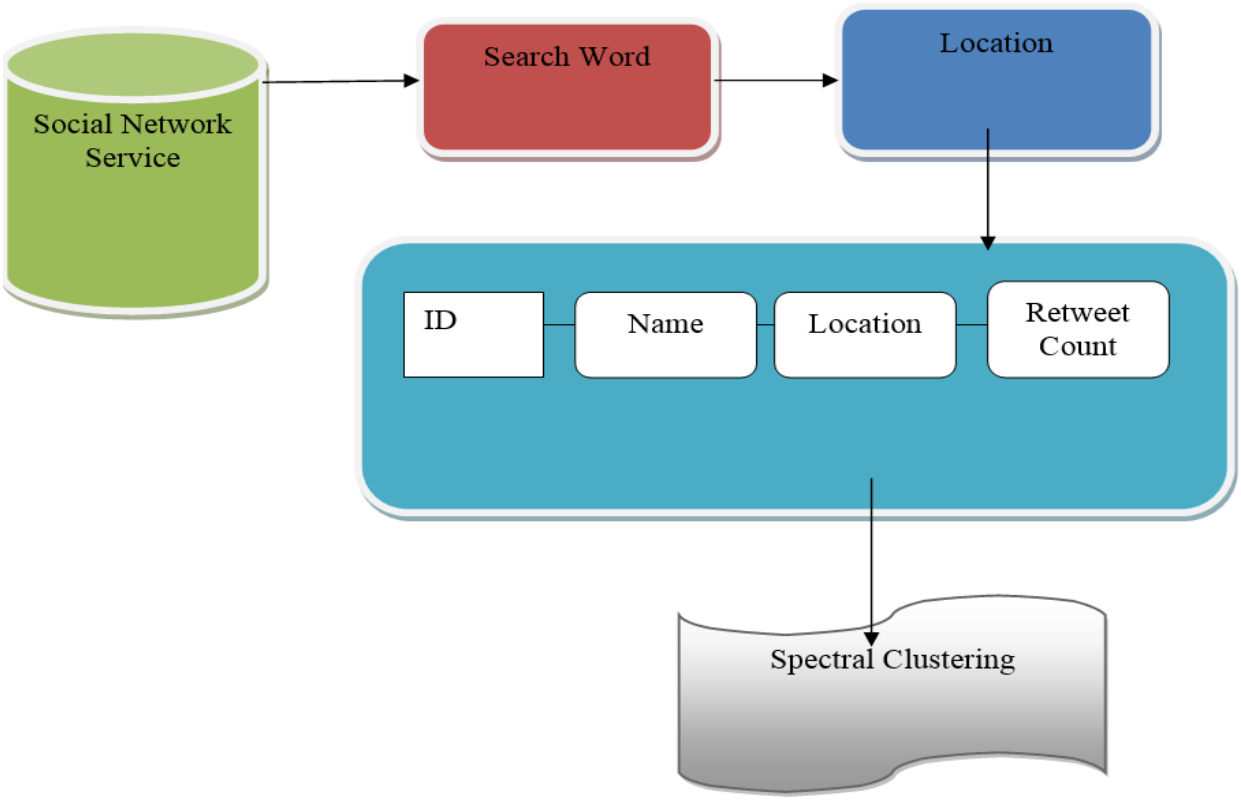



*Figure 8*. System Application Architecture

Activities or information that a client needs from the application might be the total number of users in a particular area, number of tweets in particular location, total tweets of a particular tweet, number of people who tweeted a particular tweet, number of hash tags for a particular tweet, responses to a particular tweet, and so forth. This information will be helpful to client in making some analysis and make decisions. The client can retrieve information from the application by entering two types of input, one is key based information like Hashtag or Location based information like the location of the user. The client enters information related to the data that he needs in the search word tab of the home page. The entered keyword by the client can be either key based information like a hashtag or location based information like the location of a particular user, but will still be able to get the results for both key based information and location-based information.

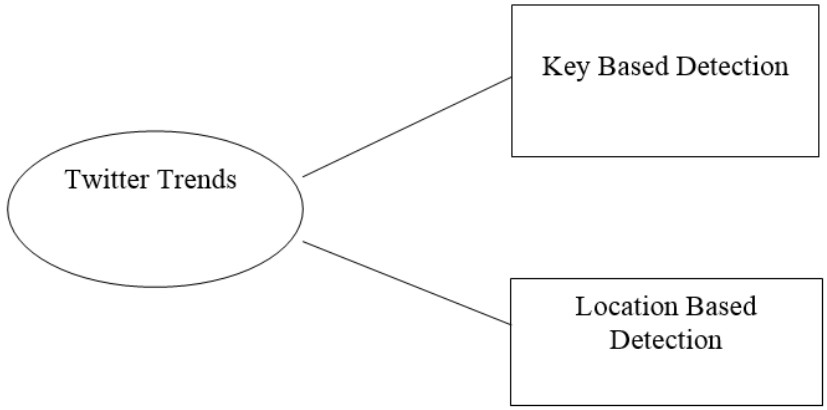
This application is designed such that it retrieves information from the database based on Spectral Clustering. The following sections in this chapter clearly explain the process of the designed application.

**Data Flow Diagram**

1. The DFD is called as bubble diagram. It is a straightforward graphical formalism that can be utilized to speak to a framework as far as information given to the framework, different processing took on this information and the result of the framework
2. The Data Flow Diagram (DFD) is a very powerful designing tool. It is utilized to display the framework segments. These segments are the framework procedure, the information utilized by the procedure, an outer element that collaborates with the framework and the data streams in the framework.

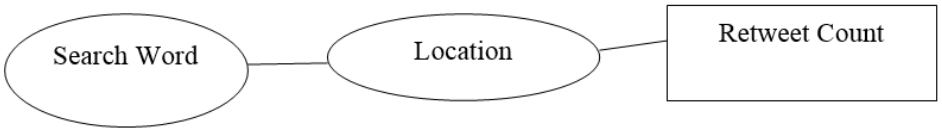
3. DFD shows how the data travels through the framework and how it is changed by a progression of changes. It is a graphical method that delineates data stream and the changes that are connected as information moves from contribution to yield.
4. DFD is otherwise called bubble outline. A DFD might be utilized to speak to a framework at any level of deliberation. DFD might be divided into levels that speak to expanding data stream and operative detail.

Level 0



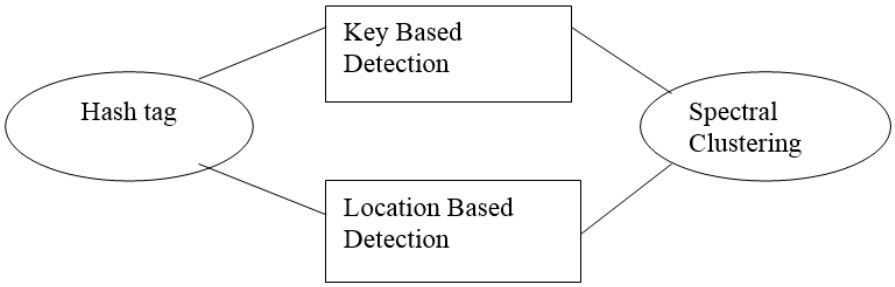

The Level 0 explains the Twitter trends, it can be divided into two sub-divisions one is key-based detection, and other is location-based detection. Key based keyword can be keyword like hashtag or part of tweet information. Location based keyword is location information. Client can give input either through key based or location based.

Level 1



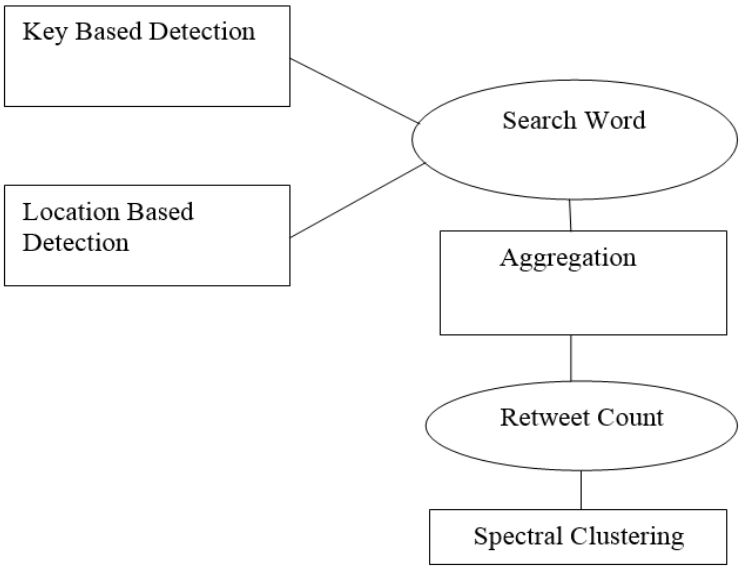
The Level 1 explains that when the Client searches for information with the location-based keyword. The Client just enters a keyword related to location in the search tab, now the Client will be able to retrieve complete data of the particular user that he is looking for like ID, location name, retweet count, hashtag.

Level 2




The Level 2 explains that when the client enters hashtag in search word tab in Home Scree, Client will be able to retrieve data related to key-based information and location-based information through processing called Spectral Clustering.

Level 3



The Level 3 explains the client can search with two methods Key based detection (hashtag) and location based detection (location name) to get the data. The data is aggregated and filters with Spectral Clustering to get the output. The key based input that client can enter would be Hashtag of a tweet and Location-based input that client can enter would be location name. Whatever might be the input is based on, that is it might be location-based or key-based he can retrieve.

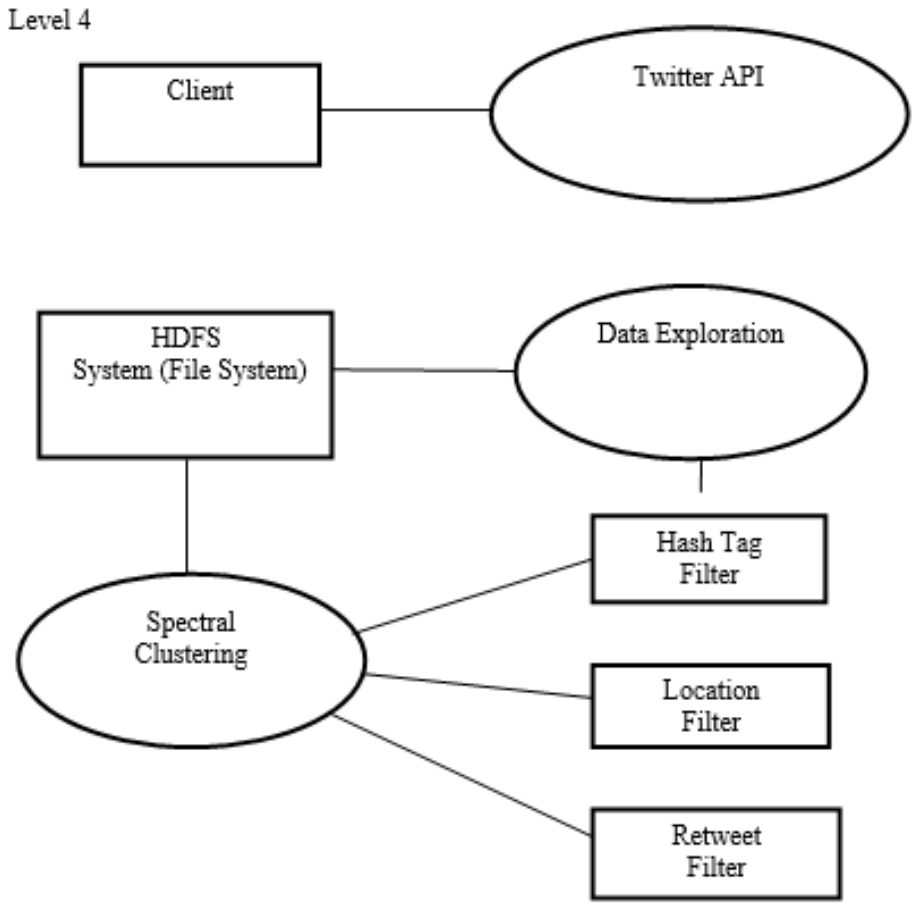



*Figure 9.* Data Flow Diagram

The Level 4 explains various stages that data flows to retrieve the information requested by the client. When the client first enters the keyword in search word tab on Home Page, the data

would be aggregated and picks information from the File System, and through Spectral Clustering filter the data based on division and displays the output.

**Use Case Diagram**

A utilization case graph in the Unified Modeling Language (UML) is a kind of behavioral outline characterized by and made from a Use-case investigation. Its motivation is to show a graphical review of the usefulness gave by a framework regarding performing artists, their objectives (spoke to as use cases), and any conditions between those utilization cases. The primary reason for a utilization case graph is to show what framework capacities are performed for which performer. Parts of the performing artists in the framework can be portrayed.

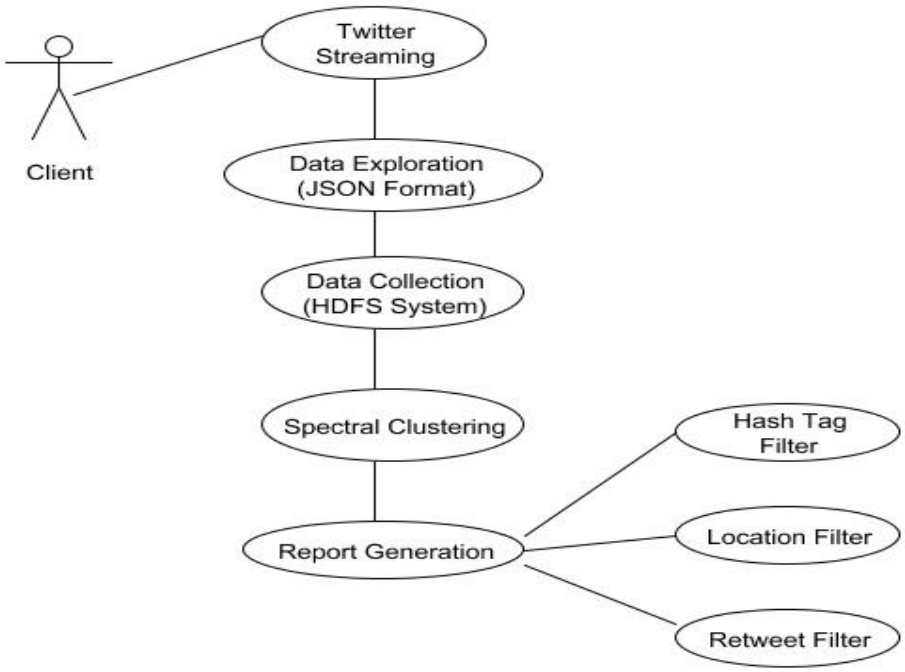



*Figure 10*. Use Case Diagram

The study explains the flow of the process starting from the client searching for data by entering keyword in search work tab of home screen to get the information that the client requested as output.

**Class Diagram**

In Software Engineering, a class outline in the Unified Modeling Language (UML) is a kind of static structure chart that portrays the structure of a framework by demonstrating the framework's classes, their qualities, operations (or techniques), and the connections among the classes. It clarifies which class contains data.

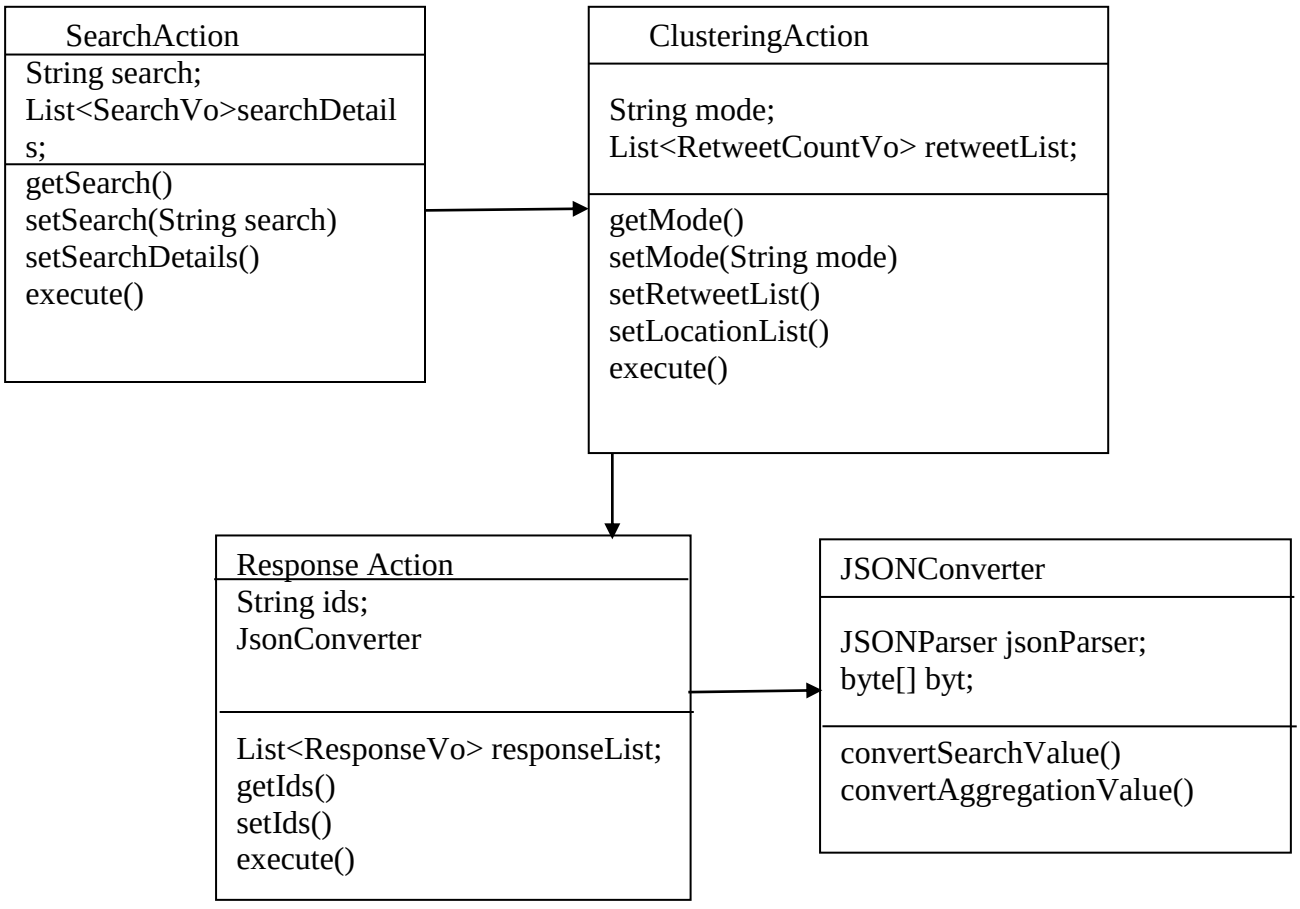



*Figure 11*. Class Diagram

The Class Diagram explains that the search action Class uses variables such as string search, List<SearchVo>searchDetails and methods getSearch(), setSearch(String search), setSearchDetails(), execute().

The second class is clustering action, this class uses variables String mode, List<RetweetCountvo>retweetList and methods getMode(), setMode(String mode), setRetweetList(), setLocationList(), execute().

The third class is Response Action, this class uses variables such as String ids, JsonConverter and methods List<ResponseVo> responseList, getIds(), setIds(), execute().

The final class name is JSONConvertor, the variables used are JSONParser jsonparser, byte[] byt and methods are convertSearchValue(),convertAggregationValue().

**Sequence Diagram**

A Sequence Diagram in Unified Modeling Language (UML) is a sort of association graph that shows how forms work with each other and in what request. It is a development of a Message Sequence Chart. Succession graphs are now and again called occasion outlines, occasion situations, and timing charts.

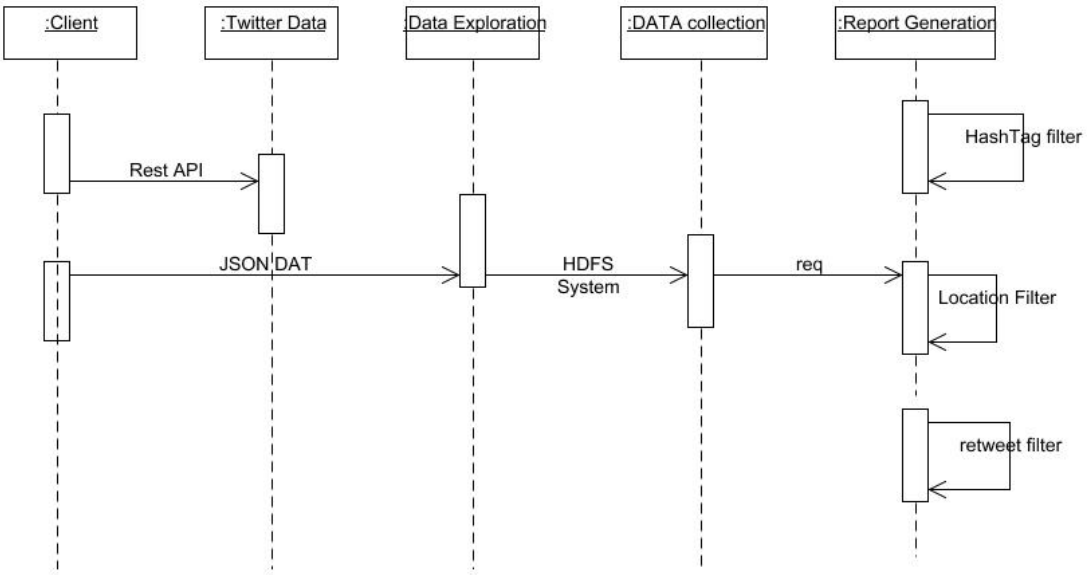



*Figure 12*. Sequence Diagram

The Sequence Diagram explains the flow process with respect to order of timing. Starts from the client searching for the data by using tweeter streaming (tweeter API) the received data will be move to explored (JSON Format) and it will be collected through data collection (HDFS System) after the data will be filtered using spectral clustering and the report generated by three options called location, hash tag and retweet count.

**System Requirements**

Table 3

*Hardware Requirements*

| Hardware | Requirements |
|---|---|
| System | Pentium IV 2.4 GHz |
| Hard Disk | 40 GB |
| Floppy Drive | 1.44 Mb |
| Monitor | 15 VGA Color |
| Ram | 512 Mb |

Table 4

*Software Requirements*

| Software | Requirements |
|---|---|
| Operating System | Windows XP/7. |
| Coding Language | JAVA/J2EE |
| IDE | NetBeans 7.4 |
| Database | MYSQL |

**Summary**

Architecture of the system, drawbacks in existing system, proposed system advantages, system feasibility are discussed. System Design is explained through data flow diagram, use case diagram, class diagram, and sequence diagram. System requirements both relating to hardware and software are also discussed. Implementation and test results are discussed in Chapter IV.

# Chapter IV: Implementation

## Main Modules

Main modules involved in implementing this project are integrating and mining data, Big Data fast response, pattern matching and mining, key technologies for integration and mining, group influence and interactions.

## Modules Description

**Integrating and mining data.** Data from different sources is composed and mined to unwind and utilize the structure of natural frameworks to shed new bits of information on the components of characteristic systems. The speculative underpinnings are addressed, present and future engaging advances for consolidating and mining natural frameworks. The procedures and methods in information obtainment, transmission, and get ready for information frameworks are augmented and consolidated. Systems for semantic-based data integration, mechanized hypothesis period from mined data, and robotized versatile investigative gadgets to evaluate diversion and refine models are designed.

**Big data fast response.** To gather a stream-based Big Data investigative framework for fast response and continuous essential authority is proposed. Designing Big Data assessing parts to decrease Big Data volumes to a sensible size for taking care of, Building desire models from Big Data streams. Such models can adaptively adjust to the component changing of the data, and moreover exactly foresee the example of the data later and a learning requesting framework to ensure continuous data watching and course of action for Big Data applications.

**Pattern matching and mining.** A proficient examination on case organizing, plan mining with unique cases, and application issues are designed. Exploration of the NP-hard

diverse nature of the organizing and mining issues, Multiple plans organizing with wild cards, Approximate plan organizing and mining, and Application of our examination onto unavoidable modified information taking care of and bioinformatics.

**Key technologies for integration and mining.** An investigation on the availability and mathematical regularities of multisource, massive and dynamic information, including cross-media search based on data extraction, sampling, random information querying, and cross-domain and cross-platform information polymerization is conducted. To break through the barriers of traditional data mining techniques, an independent data discovery and mining in aggregate inline data, mining in data streams, multi-granularity knowledge discovery from huge multisource data (Xindong et al., 2014) are studied.

**Group influence and interactions.** Employing group effect and information scattering models, and contemplating gathering participation rules in casual associations using dynamic redirection speculation

Studying clever individual assurance and effect appraisals under interpersonal associations affected by social affair feeling, and inspecting energetic affiliations and effect among individuals and totals, and establishing an astute effect model and its figuring systems for casual group social events, to reveal the instinctive effect and advancement of interpersonal associations.

**Detail Design**

**Twitter data generation.** Twitter is an extremely widespread platform for information exchange, can be used as a data-mining source which could support in the aforementioned hurdles which is received by sensor nodes. Precisely, using a large data set of collected tweets,

sensor nodes join with sink to transport the dataset to HDFS system. The REST APIs grants the programmatic path to read and write Twitter data. Author a new Tweet, read author profile and follower data, and more. The REST API recognizes Twitter applications and users using OAuth; responses are available in JSON.

**Exploration.** This step customarily starts with data development which may require scrubbing data, data conversions, and choosing subsets of records and in a case of data sets with enormous numbers of variables ("fields") performing some preparatory feature selection methods to bring the number of variables to a manageable. Making use of heterogeneous data is a major challenge for Big Data applications because any two individuals in a complex network are probably interested to each other with a social relationship. Such a relationship is quadratic on the number of nodes in the network, so a million-node links may be subject to one trillion links.

**Data collection.** In this, accumulated data is transferred to HDFS system and spectral clustering is implemented to conduct data analytics meant on the Hash name, Location and retweet number. As Big Data applications are indicated by independent sources and decentralized controls, summing dispersed data sources to a centralized site for mining is systematically restrictive due to the implied transportation cost and secrecy concerns. On the other hand, though, mining actions can be regularly carried out at each distributed site, the biased view of the data collected at each site often leads to biased decisions or models.

**Spectral clustering.** For this analysis, various user lists on Twitter as the ground truth data for a group of users is used. All the tweets from the users who were enrolled in the lists and then tried to obtain clusters by using various relationship metrics utilizing the spectral clustering

algorithm are collected. Despite this, various relationships between users in addition to just the social relationships to find out other characteristics that influence the users being placed together are examined. Effects of implementing spectral clustering algorithm using the modularity matrix and the symmetric normalized Laplacian matrix are displayed. The effects of these approaches while using various complex input matrices formed by the different order of the above similarity measures are analyzed.

**System Testing**

The reason for testing is to find errors. Testing is the way toward attempting to find each possible flaw or shortcoming in a work product. It gives an approach to check the usefulness of segments, sub-congregations, gatherings and a completed product It is the way of practicing software with the aim of guaranteeing that the ssoftware system lives up to its necessities and client desires and does not fall flat in an unacceptable way. There are different sorts of test. Every test sort addresses a particular testing necessity.

**Types of Tests**

- **Unit testing**. Unit testing involves the design of test cases that certify that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at the component level and test a specific business process, application, and system configuration. Unit

tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

- **Integration testing**. Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.
- **System Test**. System testing assures that the complete combined software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.
- **White Box Testing**. In White Box testing, the tester should have an understanding of inner functionalities, format, and language of the tool or at least is purpose. It is used to test where black box testing cannot be tested.
- **Test Results**. All test results are documented and are successfully executed and 'Passed.' All requirements reflect a "passed" status.
- **User Acceptance Testing**. Confirmation that the data can be utilized by the business groups for the intended purpose.

**Results/Output Screenshots**

Figure 13 is the input page or the first page of the application. Client can enter a keyword in search word tab. This keyword helps in retrieving information from the database which the client can use in his analysis for improving services to the customer. The keyword that client enters in search word tab can either be key-based information like hashtag of the tweet or location-based information like location. Even though the client enters either the key-based information or location-based information, the whole data can be retrieved which would simply the task of the client in searching the information. As said this information reterieved by client from database can be used to improve services offered to the customer like increasing the privacy or security or improving the customer exeperience in using the application by enhancing its features for easy and better operation. The information will be retrieved from the data source by process called spectral clustering. Spectral clustering is implemented to conduct data analytics meant on the hash name, location and retweet count. Effects of implementing spectral clustering algorithm is displayed by using modularity matrix and the symmetric normalized Laplacian matrix.

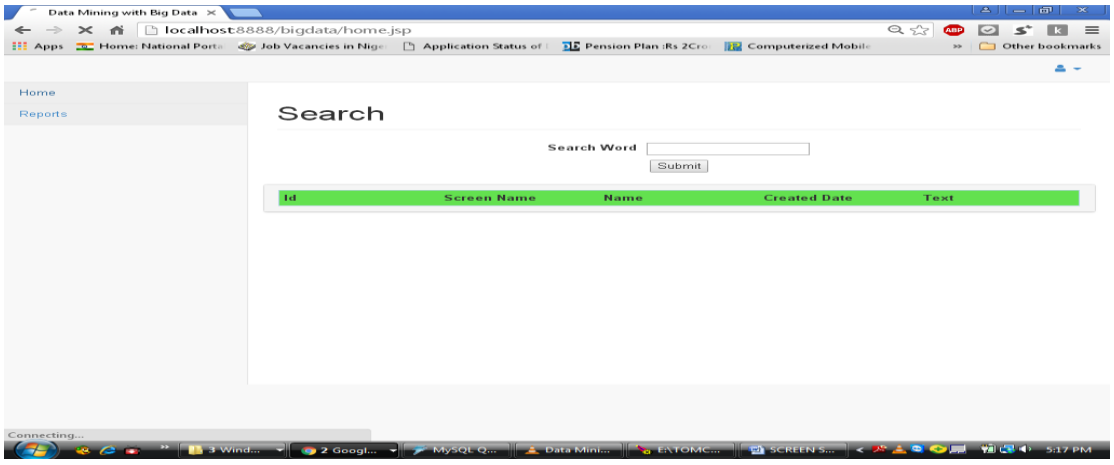



*Figure 13*. Screen Shot of Home Page

Figure 14 shows that the client has entered location name as keyword for searching in search word tab. Location name that he entered is France.

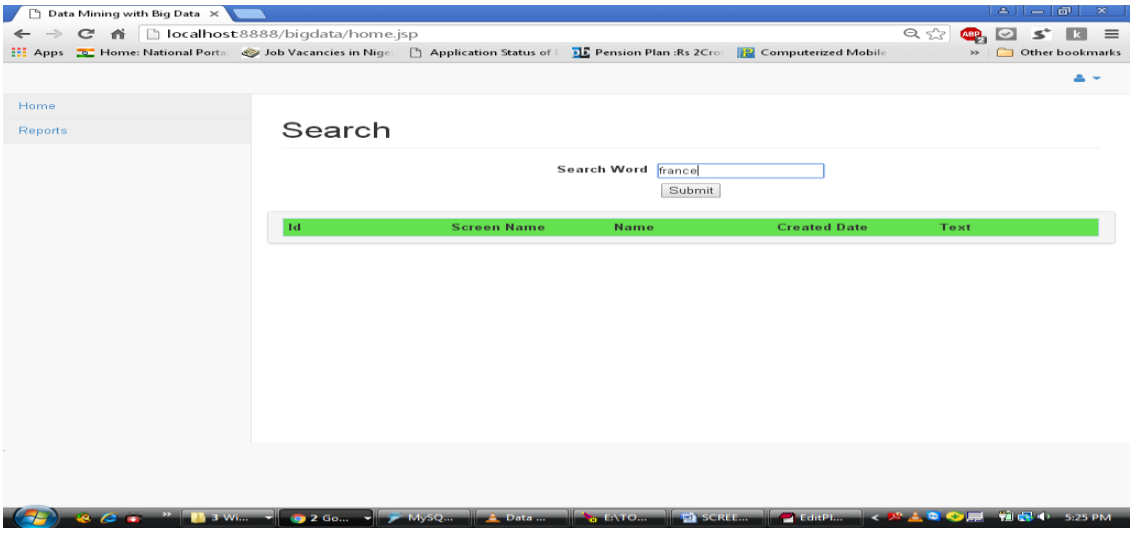



*Figure 14*. Screen Shot of Search by Location Name

Figure 15 shows the results after entering keyword Location name in Search word tab. All the ID's, customer names, tweets of customer in that particular location are displayed as in the Figure 15.

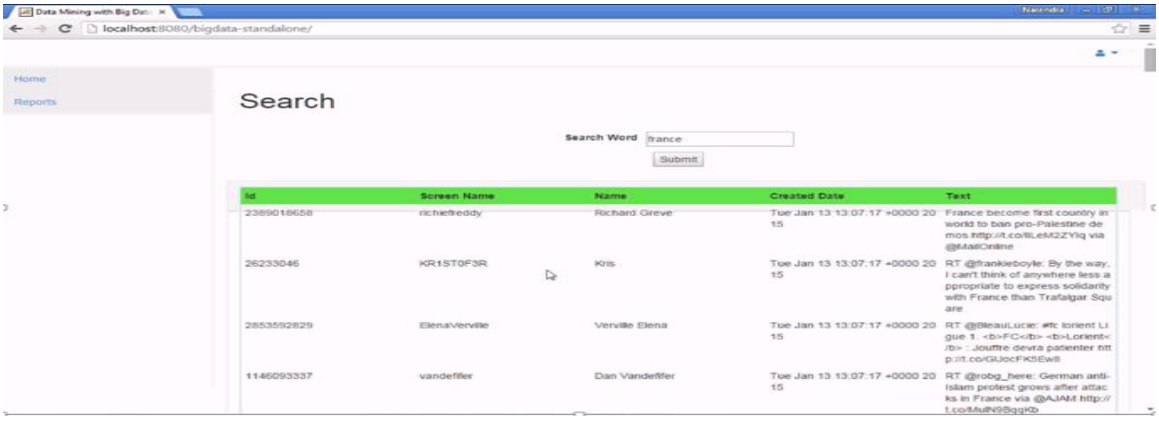



*Figure 15*. Screen Shot of Search by Location

Figure 16 shows that reports can be generated for the results by clicking on Reports button on the left side of the window and is below the Home tab.

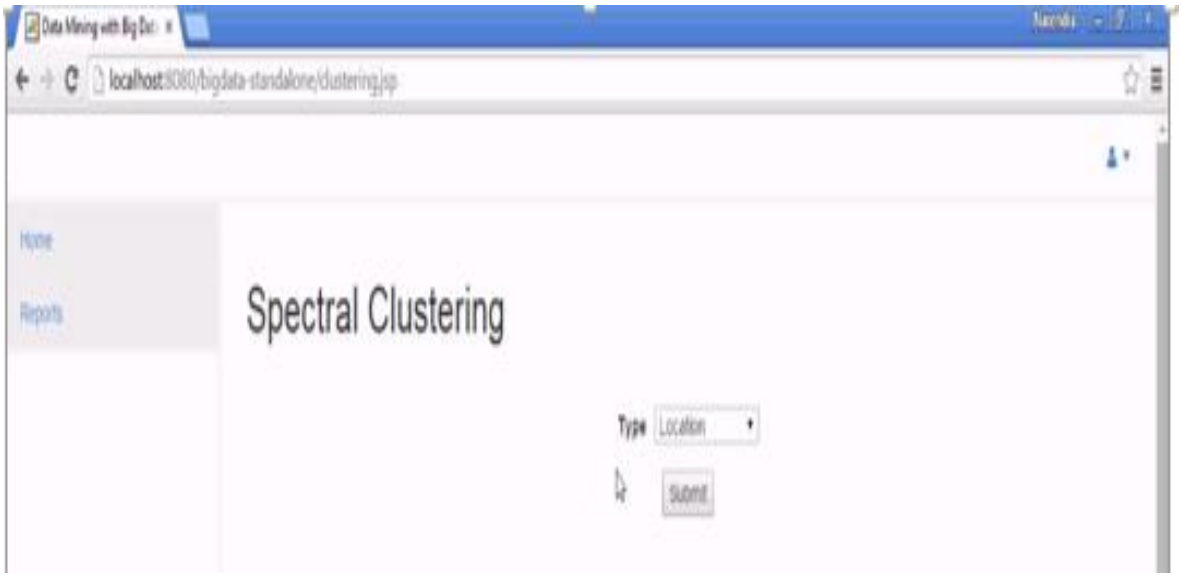



*Figure 16*. Screen Shot of Reports Page

Figure 17 shows the results of spectral clustering by entering the keyword Location name in the search word tab. Overall locations are displayed as show above in the figure.

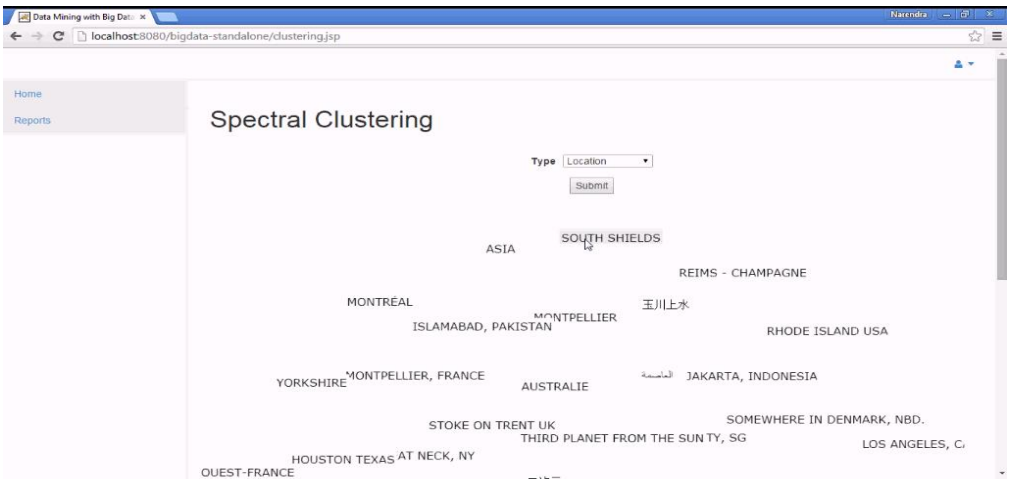



*Figure 17*. Screen Shot of Spectral Clustering by Location Option

Figure 18 shows the result when key based input is entered in search word tab which is Hash tag. Overall hash tags would be displayed on the page which would be a video image and hash tag would be scrolling which displays all the hash tags.

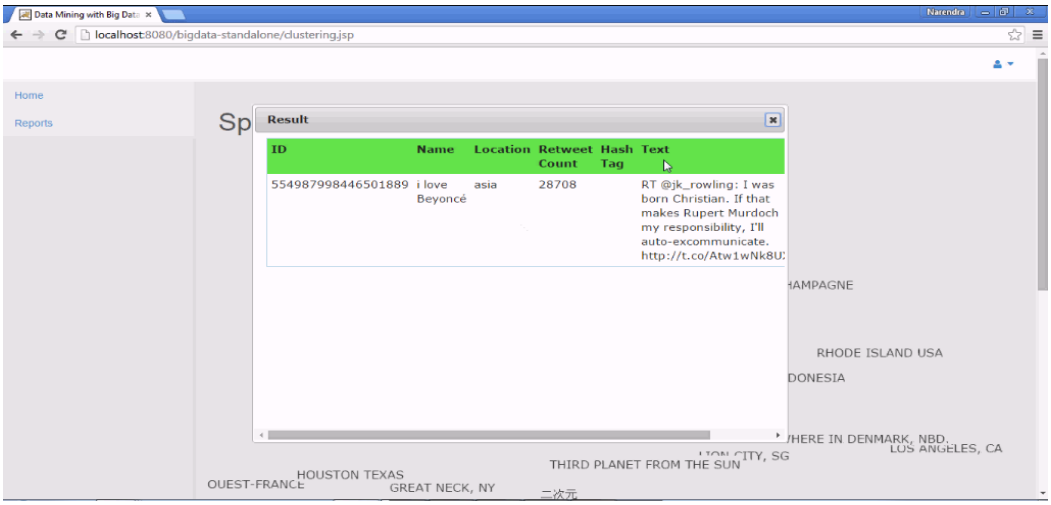



*Figure 18*. Screen Shot When Hash-Tag Is Selected for Key-Based Input

Figure 19 shows the output for spectral clustering for hash tag option. In this scenario, the client enters the input which is key-based input hash tag.

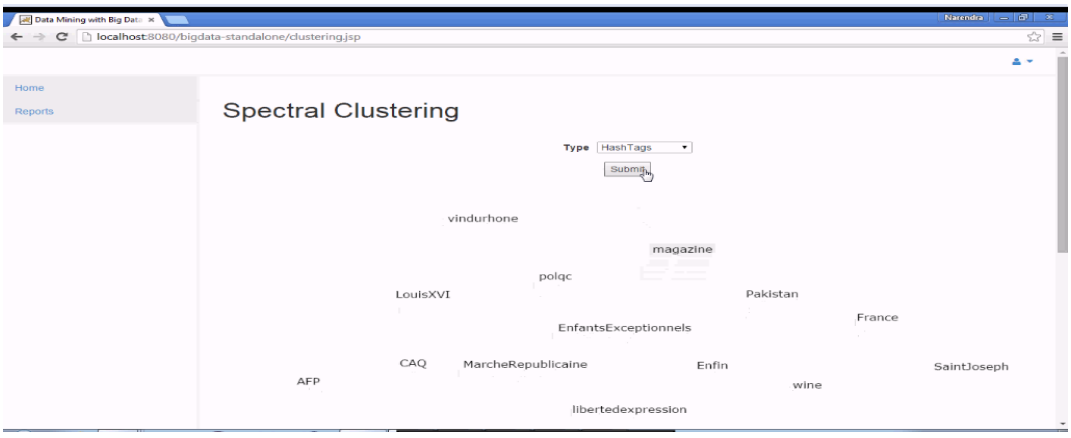



*Figure 19*. Screen Shot for Spectral Clustering by Hash Tags Option

Figure 20 shows client selecting option Retweet-count in search word tab. Here the client after selecting the Retweet-count option will enter the Retweet-count.

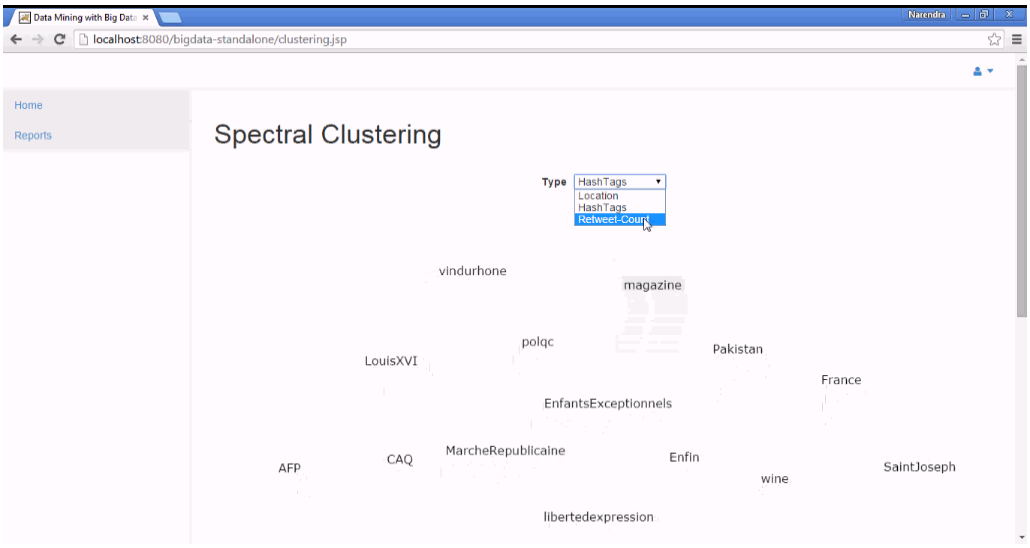



*Figure 20*. Screen Shot for Spectral Clustering by Select Retweet-Count option

Figure 21 shows the output in the form of spectral clustering for input retweet-count. When the client enters input retweet-count which is key-based input. Client will be able to get information which is total number of tweets of particular tweet-based on hashtag or based on location or based on the user.


*Figure 21*. Screen Shot when Retweet-Count is Selected for Key-Based Input

Figure 22 shows the output in the form of report for the input which is key-based that is retweet count value. When the client enters the input, which is retweet count, the output is displayed which is the total number of tweet with that count, total people tagged in that particular tweet, ID and location of tweet.

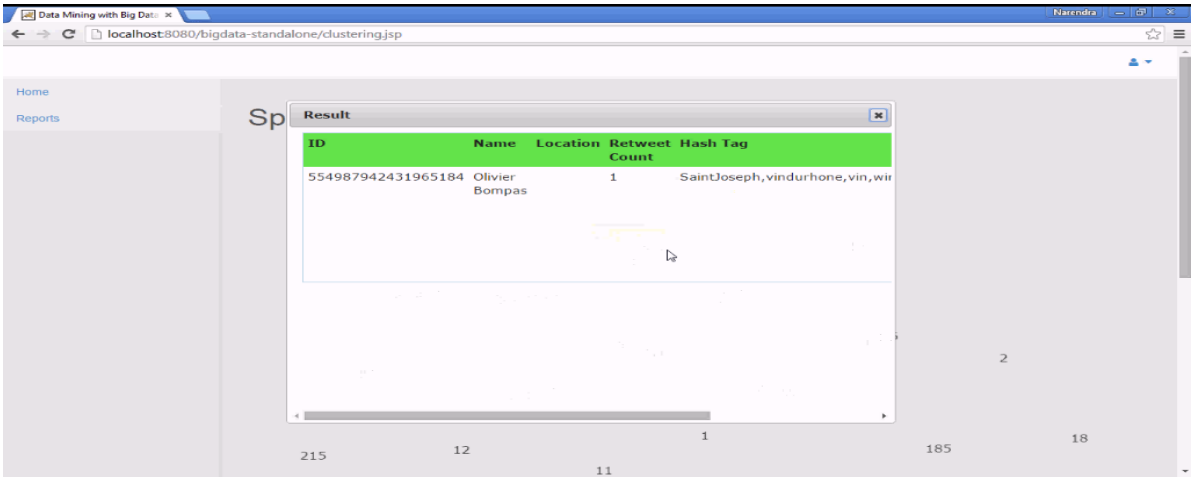



*Figure 22*. Screen Shot of Results Page When Clicked on Retweet-Count.

**Summary**

Various phases in implementation of this project are discussed clearly. System testing is performed to check if the code is working correctly. Results are explained step by step with the help of out-put screens. The study is concluded and future work is discussed in the Chapter V.

# Chapter V: Conclusion and Future Work

## Conclusion

In this project, an efficient and cost effective model is implemented in which required data can be retrieved from data source which has the characteristics of Big Data. A data source similar to that of Twitter is taken, and this data source has properties which are huge, heterogeneous, and complex and dynamic, which shows that it is correct match for Big Data source as it has characteristics of Big Data. System architecture is designed in such a way that required data can be retrieved from the data source by entering the keyword related to the data to be retrieved in search tab. The keyword entered in search word tab can be either location-based input or key-based input. Even though the client enters the key-based keyword in search tab, both the key-based information and location-based information will be retrieved. The same with location-based keyword as input, both key-based and location-based information is retrieved. Client access to retrieve the information from the data source through input page is access controlled so that privacy and security requirements are meet. This information retrieved by client from social network site is valuable resource for the client. The client uses this information retrieved from the social network to analyze the data, do research to make important decisions by which the customer experience can be improved by increasing the privacy and security of customer's data. This data can also be used for applications like financial data analysis, retail industry, and marketing industry. Marketing industry uses the data to populate the customized advertisements as per the customer's activity online.

**Future Work**

The amounts of data are growing exponentially worldwide due to the explosion of social networking sites, search engines, communications sharing sites, stock exchanging sites, news sources and so on. Big Data is becoming the new area for scientific information research and marketing applications. Big data mining is a hopeful research area, still in its origin. Despite the limited work done on big data mining so far, much more effort is needed to overcome is hurdles related to heterogeneity, scalability, speed, efficiency, trust, and privacy. The research can be extended by adding more measures to increase the privacy and security in social networking site to increase the trust among the customers so that they can share information on social network without concerns.

## References


Agarwal, N., & Banerjee, S. (2012). Analyzing collective behavior from blogs using swarm intelligence. *Knowledge and Information Systems, 33*(3), 523-547.

Apache Tomcat 6.0. (2016). *Documentation index*. Retrieved from http://tomcat.apache.org/tomcat-6.0-doc/index.html

BBC News. (2014). Big data: Are you ready for the blast-off? Retrieved from http://www.bbc.com/news/business-26383058

Boyd, D. (2004). Friendster and publicly articulated social networks. *Proceedings of the SIGCHI Conference on Human Factors and Computing Systems,* Vienna, Austria.

Boyd, D. (2006). Friends, friendsters, and top 8: Writing community into being on social network sites. *First Monday, 8*, 11-12. Retrieved from http://firstmonday.org/article/view/1418/1336

Bughin, J., Chui, M., & Manyika, J. (2010). *Clouds, big data, and smart assets: Ten tech-enabled business trends to watch*. Retrieved from http://www.mckinsey.com/industries/high-tech/our-insights/clouds-big-data-and-smart-assets-ten-tech-enabled-business-trends-to-watch

Computer Business Review. (2015). *Top 10 biggest data centers from around the world*. Retrieved from http://www.cbronline.com/news/data-centre/top-10-biggest-data-centres-from-around-the-world-4545356/

Cormode G., & Srivastava D. (2009). Anonymized data: Generation, models, usage. *Proceedings of ACM SIGMOD International Conference on Management Data,* 1015-1018.

Cukier, K. (2010). *Data, data everywhere*. Retrieved from http://www.economist.com/node/15557443?story_id=15557443

Diebold, F. (2000). *Big Data dynamic factor models for macroeconomic measurement and forecasting*. Discussion read to the 8th World Congress of the Econometric Society.

Gao, C., & Iwane, N. (2015). A social network model with privacy preserving and reliability assurance and its applications in health care. *International Journal of Energy, Information and Communications, 6*(1).

Gross, R., & Acquisti, A. (2006). Imagined communities: Awareness, information sharing and privacy on the Facebook. *Proceedings of the 6th Workshop on Privacy Enhancing Technologies*, Cambridge, UK.

Gurin, J. (2014). *What is big data*. Retrieved from https://datascience.berkeley.edu/what-is-big-data/#JoelGurin

Howe, D., Costanzo, M., Pey, P., Gojobori, T., Hannick, L. . . . . & Rhee, S. Y. (2008). Big data: The future of biocuration. *Nature, 455*, 47-50.

Huberman B. (2012). *Sociology of science: Big data deserve a bigger audience*. Retrieved from http://www.nature.com/nature/journal/v482/n7385/full/482308d.html

IBM. (2012). Spark, an alternative for fast data analytics. Retrieved from http://www.ibm.com/developerworks/library/os-spark/.

Karypis, G., & Ahmed, R. (2012). Algorithms for mining the evolution of conserved relational states in dynamic networks. *Knowledge and Information Systems, 33*(3), 603-630.

Lampe, C., Ellison, N., & Steinfield, C. (2007). A face(book) in the crowd: Social searching versus social browsing. *Proceedings of the 20th Anniversary Conference on Computer Supported Cooperative Work,* Banff, Alberta, Canada, pp. 167-170.

Lee, S. K., Ha, J. W., & Alam, H. (2012). Novel approaches to crawling important pages early. *Knowledge and Information Systems, 33*(3), 707-734.

Mervis, J. (2012). U.S. science policy: Agencies rally to tackle big data. *Science, 6*(6077), 22.

Miller, M. S. (2012). *Seven blind men and the giant elephant*. Retrieved from https://nsjour.wordpress.com/2012/10/21/seven-blind-men-and-the-elephant/

pingdom. (2010). *Changes in storage space*. Retrieved from http://royal.pingdom.com/2010/02/18/amazing-facts-and-figures-about-the-evolution-of-hard-disk-drives/

Rajaraman, A., & Ullman, J. (2011). *Mining of massive datasets*. Retrieved from http://infolab.stanford.edu/~ullman/mmds/book.pdf

Reiter, J. P., & Machanavajjhala, A. (2012) "Big privacy: Protecting confidentiality in big data. *ACM Crossroads, 19*(1), 20-23.

Sandeep, K., & Dhawas, V. N. (2016). Online social network for recommendation system using SVM. *International Journal of Computer Applications, 147*(14).

Tutorials Point. (n.d.). *Data mining: Applications and trends*. Retrieved from https://www.tutorialspoint.com/data_mining/dm_applications_trends.htm

Walker, D., & Aral, S. (2012). Identifying influential and susceptible members of social networks. *Science, 337*, 337-334.

Weiss, S. M., & Indurkhya, N. (1998). *Predictive data mining: A practical guide*. San Francisco, CA: Morgan Kaufmann Publishers.

Whatsthebigdata.com. (2014). *From 10MB hard drive to big data*. Retrieved from https://whatsthebigdata.com/2014/05/16/from-10mb-hard-drive-to-big-data/

Xindong, W., Xingquan, Z., Gong-Qing, W., & Wei, D. (2014). *Data mining with big data. IEEE Transactions on Knowledge and Data Engineering*, 26(1).

# Appendix

**Index.Html**

```
<html>
<head>
<title>Efficient Data</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
 <div class="header">
<ul><li><a href="index.html">HOME </a></li>
<li><a href="user.jsp">OWNER </a></li>
<li><a href="consumer.jsp">CLIENT </a></li>
<li><a href="admin.jsp">CLOUD </a></li>
</ul>
</div>
 <div class="header1">
  </div>
  <p class="piu"></p>
 <div class="content">
  <h2 align="center">&amp;nbsp; </h2>
```

```
  <h2 align="center"><marquee behavior="alternate"><font color="#3366FF">GENERAL
FLOW OF SECURE DATA ACCESS IN CLOUD</font></marquee></h2>
  <h1 align="center"><img src="images/flow.png"></h1>
 </div>
  <div class="fooder">
 </div>
</body>
</html>
```

**Register Page**

```
<%@ page language="java" import="java.sql.*,databaseconnection.*;" errorPage="" %>
<%
Connection con = null;
Statement st = null;
ResultSet rs = null;
String a = request.getParameter("fname");
session.setAttribute("fname",a);
System.out.println(a);
String b = request.getParameter("lname");
String c = request.getParameter("email");
session.setAttribute("email",c);
System.out.println(c);
```

```
String d = request.getParameter(“mno”);
String e = request.getParameter(“gender”);
String f = request.getParameter(“access”);
String g = request.getParameter(“uname”);
try{
con=databasecon.getconnection();
st=con.createStatement();
st.executeUpdate(“insert into reg(fname,lname,email,mno,gender,access,uname)
values(‘“+a+”‘,’“+b+”‘,’“+c+”‘,’“+d+”‘,’“+e+”‘,’“+f+”‘,’“+g+”‘)”);
String pass=(String)session.getAttribute(“pass”);
System.out.println(“pass is”+pass);
%>
<script>
alert(‘Kindly check mail and get access’);
window.location=‘pass.jsp’;
</script>
<%
}
catch(Exception ex)
{       out.println(ex);
}
%>
```

**Login**

```
<html>
<body bgcolor="#FF9900">
<%@ page contentType="text/html; charset=utf-8" language="java" import="java.sql.*,databaseconnection.*" errorPage="" %>
<%
String uname=request.getParameter("uname");
String pass=request.getParameter("pass");
 try{
                Connection con = databasecon.getconnection();
      PreparedStatement ps=con.prepareStatement("select id,fname,lname, email,mno,gender from reg where uname='"+uname+"' and pass='"+pass+"'");
            ResultSet rs=ps.executeQuery();
                if(rs.next())
                {
                int idw=rs.getInt("id");
                String wid = Integer.toString(idw);
                session.setAttribute("owid",wid);
                String f=rs.getString("fname");
                session.setAttribute("aa",f);
                System.out.println("firstname is"+f);
                 String l=rs.getString("lname");
```

```
                session.setAttribute("bb",l);
                System.out.println("lastname is"+l);
                String em=rs.getString("email");
                session.setAttribute("cc",em);
                System.out.println("email is"+em);
                String mno=rs.getString("mno");
                session.setAttribute("dd",mno);
                System.out.println("mobile no"+mno);
                String gender=rs.getString("gender");
                session.setAttribute("ee",gender);
                System.out.println("Gender"+gender);
               response.sendRedirect("details.jsp");
                System.out.println(uname);
               }
               }
        catch(Exception e2){
               out.println(e2.getMessage());
        }
%>
</body>
</html>
```

**Encryption:**

```
<%@ page import="java.io.*,java.util.*" %>
<html>
<head>
<%
String k=(String)session.getAttribute("aa");
System.out.println("firstname is"+k);
String l=(String)session.getAttribute("bb");
System.out.println("lastname is"+l);
String m=(String)session.getAttribute("cc");
System.out.println("email id"+m);
String n=(String)session.getAttribute("dd");
System.out.println("firstname is"+n);
String o=(String)session.getAttribute("ee");
System.out.println("firstname is"+o);
String p=(String)session.getAttribute("fid");
System.out.println("file id is:"+p);
%>
<title>Efficient Data</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
```

```
<body>
<div class="header">
<ul>
<li><a href="details.jsp">OWNER DETAILS </a></li>
<li><a href="upload.jsp">FILEUPLOAD</a></li>
<li><a href="viewencrypt2.jsp">VIEWDB</a></li>
<li><a href="user.jsp">LOGOUT </a></li>
</ul>
</div>
 <div class="header1">
  </div>
  <p class="piu"></p>
 <div class="content">
  <h2 align="center"> </h2>
  <h2 align="center"> </h2>
  <h2 align="center"> </h2>
  <h2 align="center"> </h2>
  <h1 align="center">OWNER FILE SUCCESSFULLY UPLOADED</h1>
  </div>
  <script>
  alert("Encrypted Formated Data Uploaded");
  window.location='viewencrypt2.jsp';
```

```
  </script>
 <div class="fooder">
```

**Clustering Action**

```
package com.mycompany.action;
import java.util.List;
import javax.servlet.ServletContext;
import org.apache.struts2.ServletActionContext;
import com.mycompany.logic.ClusteringLogic;
import com.mycompany.logic.HiveClientCall;
import com.mycompany.vo.HashTagVo;
import com.mycompany.vo.LocationVo;
import com.mycompany.vo.RetweetCountVo;
import com.opensymphony.xwork2.ActionSupport;
public class ClusteringAction {
	private List<RetweetCountVo> retweetList;
	private String mode;
	public String getMode() {
		return mode;
	}
	public void setMode(String mode) {
		this.mode = mode;
	}
```

```
public List<RetweetCountVo> getRetweetList() {
	return retweetList;
}

public void setRetweetList(List<RetweetCountVo> retweetList) {
	this.retweetList = retweetList;
}
ClusteringLogic clusteringLogic = new ClusteringLogic();
private List<LocationVo> locationList;
private List<HashTagVo> hashTagList;
public List<HashTagVo> getHashTagList() {
	return hashTagList;
}
public void setHashTagList(List<HashTagVo> hashTagList) {
	this.hashTagList = hashTagList;
}
public List<LocationVo> getLocationList() {
	return locationList;
}
public void setLocationList(List<LocationVo> locationList) {
	this.locationList = locationList;
}
```

```
	public String execute() {
		ServletContext context = ServletActionContext.getServletContext();
		String filePath = context.getRealPath(“”)+”/result”;
		if (mode.equals(“Location”)) {
			locationList = clusteringLogic.LocationServices(filePath);
		}
		else if(mode.equals(“HashTags”)) {
			hashTagList = clusteringLogic.HashTagServices(filePath);
		}
		else if(mode.equals(“retweet”)) {
			retweetList=clusteringLogic.retweetCountServices(filePath);
		}
		return ActionSupport.SUCCESS;
	}
}
```

**Search Action**

```
package com.mycompany.action;
import java.util.List;
import javax.servlet.ServletContext;
import org.apache.struts2.ServletActionContext;
import com.mycompany.logic.TwitterRestCall;
```

```
import com.mycompany.vo.SearchVo;
import com.opensymphony.xwork2.ActionSupport;
public class SearchAction {
	private List<SearchVo> searchDetails;
	private String search;
	public String getSearch() {
		return search;
	}
	public void setSearch(String search) {
		this.search = search;
	}
	public List<SearchVo> getSearchDetails() {
		return searchDetails;
	}
	public void setSearchDetails(List<SearchVo> searchDetails) {
		this.searchDetails = searchDetails;
	}
	private TwitterRestCall twitterRestCall;
	public String execute() {
		ServletContext context = ServletActionContext.getServletContext();
		String filePath = context.getRealPath(““) + “/” + “config.properties”;
		twitterRestCall = new TwitterRestCall(filePath);
```

```
searchDetails = twitterRestCall.getTweetSearch(filePath, search);
/*try {
filePath = context.getRealPath(“”) + “/” + “config.properties”;
FileInputStream fileInputStream=new FileInputStream(new File(filePath));
Properties properties=new Properties();
properties.load(fileInputStream);
Configuration conf = new Configuration();
conf.addResource(new Path(
properties.getProperty(“hadoopLoc”)));
FileSystem fs = FileSystem.get(conf);
Path pt = new Path(properties.getProperty(“hdfsLoc”));
FileStatus[] status = fs.listStatus(pt);
//fs.delete(pt, true);
if(status !=null)
{
for (int i = 0; i < status.length; i++) {
fs.delete(status[i].getPath(), true);
}
fileInputStream.close();
fs.close();
}
```

```
                filePath = context.getRealPath(““) + “/result”;
                File = new File(filePath);
                String[] file_list = file.list();
                for (String fileAbsolutePath : file_list) {
                        String fileAbsPath = filePath + “/” + fileAbsolutePath;
                        File file_abs = new File(fileAbsPath);
                        fileInputStream = new FileInputStream(file_abs);
                        byte[] byt = new byte[fileInputStream.available()];
                        fileInputStream.read(byt);
                        fileInputStream.close();
                        String outPutDir = new File(
                                        New
File(file_abs.getParent()).getParent()).getParent()+ “/result”;
                        FileOutputStream fileOutputStream2 = new
FileOutputStream(new File(outPutDir + “/”+ new File(fileAbsPath).getName()));
                        fileOutputStream2.write(byt);
                        fileOutputStream2.close();
                        file_abs.delete();
                }
        } catch (Exception e) {
                e.printStackTrace();
        }*/
```

```
            return ActionSupport.SUCCESS;
        }
}
```

**Response Action**

```
package com.mycompany.action;
import java.util.List;
import javax.servlet.ServletContext;
import org.apache.struts2.ServletActionContext;
import com.mycompany.logic.JsonConverter;
import com.mycompany.vo.ResponseVo;
import com.opensymphony.xwork2.ActionSupport;
public class ResponseAction {
        private JsonConverter;
        private String ids;
        private List<ResponseVo> responseList;
        public List<ResponseVo> getResponseList() {
            return responseList;
        }
        public void setResponseList(List<ResponseVo> responseList) {
            this.responseList = responseList;
        }
        public String getIds() {
```

```
		return ids;
	}
	public void setIds(String ids) {
		this.ids = ids;
	}
	public String execute() {
		jsonConverter=new JsonConverter();
		ServletContext context = ServletActionContext.getServletContext();
		String filePath = context.getRealPath(“”)+”/result/result.json”;
		System.out.println(“ids ::” + ids);
		responseList=jsonConverter.getResponseVos(filePath, ids);
		return ActionSupport.SUCCESS;
	}
}
```

**TwitterResetCall**

```
package com.mycompany.logic;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.util.List;
import java.util.Properties;
import oauth.signpost.OAuthConsumer;
```

```
import oauth.signpost.commonshttp.CommonsHttpOAuthConsumer;
import org.apache.commons.fileupload.FileUpload;
import org.apache.commons.io.IOUtils;
import org.apache.http.HttpHost;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.conn.params.ConnRoutePNames;
import org.apache.http.impl.client.DefaultHttpClient;
import com.mycompany.vo.SearchVo;
public class TwitterRestCall {
        static String consumerKeyStr = ““;
        static String consumerSecretStr = ““;
        static String accessTokenStr = ““;
        static String accessTokenSecretStr = ““;
        OAuthConsumer consumer;
        private JsonConverter;
        public TwitterRestCall(String filePath) {
                jsonConverter = new JsonConverter();
                consumerKeyStr = readProperties(filePath, “consumerKey”);
                consumerSecretStr = readProperties(filePath, “consumerSecret”);
                accessTokenStr = readProperties(filePath, “accessToken”);
```

```
        accessTokenSecretStr = readProperties(filePath, “accessTokenSecret”);
        consumer = new CommonsHttpOAuthConsumer(consumerKeyStr,
                        consumerSecretStr);
        consumer.setTokenWithSecret(accessTokenStr, accessTokenSecretStr);
}
private String readProperties(String filePath, String key) {
        String url = ““;
        try {
                Properties = new Properties();
                FileInputStream fileInputStream = new FileInputStream(new File(
                                filePath));
                properties.load(fileInputStream);
                url = properties.getProperty(key);
        } catch (Exception e) {
                e.printStackTrace();
        }
        return url;
}
public List<SearchVo> getTweetSearch(String filePath, String queryName) {
        try {
                String url = readProperties(filePath, “search”);
                queryName = queryName.replace(“#”, ““);
```

```
        queryName = queryName.replace(“ “, “+”);
        url += “=“ + queryName + “&count=1000”;
        System.out.println(url);
        HttpGet request = new HttpGet(url);
        consumer.sign(request);
        HttpClient client = new DefaultHttpClient();
        HttpResponse response = client.execute(request);
        int statusCode = response.getStatusLine().getStatusCode();
        System.out.println(statusCode + “:”
                        + response.getStatusLine().getReasonPhrase());
        filePath = filePath.replaceAll(“config.properties”,
                        “result/result.json”);
        File file=new File(
                        filePath);
        FileOutputStream = new FileOutputStream(file);
        fileOutputStream.write(IOUtils.toString(
                        response.getEntity().getContent()).getBytes());
        List<SearchVo> searchDetails = jsonConverter
                        .convertSearchValue(filePath);
        fileOutputStream.flush();
fileOutputStream.close();
```

```
/*FileInputStream fileInputStream=new FileInputStream(new File(filePath));
byte[] byt=new byte[fileInputStream.available()];
fileInputStream.read(byt);
fileInputStream.close();
String outPutDir=new File(new File(file.getParent()).getParent()).getParent()+"/result";
FileOutputStream fileOutputStream2=new FileOutputStream(new File(outPutDir+"/"+new File(filePath).getName()));
fileOutputStream2.write(byt);
fileOutputStream2.close();*/
return searchDetails;
} catch (Exception e) {
e.printStackTrace(); } retrun null; } }
```