Automation system for Smart City to collect and Address Civic Issues Using Social Media

Naveen Kumar K N[1], Sunil M P[2], Darshan M U[3], Mallikarjuna K[4] and Mohan Kumar K N[5].

[1] nvn2311@gmail.com,[2]sunil_roni.1920@gmail.com,[3]darshandarshi26@gmail.com,[4] greatmallikarjun@gmail.com
[5]Assistant Professor, mohan4183@gmail.com Department of Computer Science and Engineering, SJB Institute of Technology

This work has been supported by the Karnataka state council for science and technology-Student Project Programme: 41st Series: 2017 - 18

***Abstract*— Due to wide spread area of the city and huge population, it becomes very difficult to reach the problems of the society[1], the busy schedule of the citizens makes them to not submit the problem details to respected authorities. Intern the authority will not know about the problems faced by the citizens. With the popularization of smartphones, universal coverage of the Internet and advancement of technology, social media[2] hits the main stream in citizens' daily life, and responsively the citizens' express their emotions about the civic-issues[3] that they encounter. It becomes important to connect to social media to get the civic problems of any area in the city. In this paper we are proposing an idea and the methods to measure the public emotions regarding the civic-issues[3].**

Keywords—society; social media; civic-issues;

INTRODUCTION

Citizens in a densely populated city are facing a lot of problems regarding the societal issues [1]; these societal issues can be classified based on their domains, they are healthcare issues, environmental monitoring issues, civic issues and many more [2]. In this paper we are concentrating on the civic issues. Civic issues are such as garbage issues, road issues, water issues and etc..., these issues has been a major drawback for a city development [3], so solving these issues by the governing authorities has become an important aspect for the city development and citizens welfare. For solving these issues it is important to know the problems faced by the citizens on reason of the civic issue. Due to the rapid growth in the population and wide spread area, it becomes difficult know the issues that are faced by the citizens [4]. The existing method for making the authorities to know about the civic issues is by raising a complaint in the respected authorities. But due to the busy life schedule and prioritized work environment, most of the citizens are not launching the complaints, intern the authorities will be not aware of these issues faced by the citizens [3].

Now days the social media has become an important aspect in day to day life. Due to its viral nature and responsive behaviours, it has become an emotion expressing tool for the people [5]. So most of the peoples will be posting their day to day happiness, sorrow, gratitude, problems and etc… mainly the problems related to the civic issues are also posted. If the problems are connected to most of the peoples, they'll become viral. So it becomes important to the authorities to connect to the social media to get the civic issues faced by the citizens.

In this paper we show how identify these issues from various locations by accumulating the information from the social media, processing and storing it and generating the report based on the issues. Intern this can be considered as a public sensing tool using the ICT (Information and Communication Technology) for smart city. This process give arise to the following key challenges.

1.Data acquisition from social media in Real Time.
2.Data Ingestion.
3.Data Transformation and Storage.
4.Quarrying and Report generation.

A. Data acquisition

Big data mainly came into existence because of the rapid growth of social media. Twitter has appeared to be the one of the most popular social media over the Internet [6]. Twitter receives tens of millions of tweets per day, creating huge data in unstructured form. A lot of research has been carried out to extract useful information from twitter raw data, so in this paper we are using the twitter as our data source. Data acquisition is done in real-time from the twitter. The tool used in our framework for acquiring data from the twitter is Apache NIFI [7]. Apache NIFI acts as a data flow manager for filtering the data streams from the twitter.

B. Data Ingestion

Our framework uses this message brokering system. To balance the incoming load, Topics are defined and each of these Topics is split into multiple partitions, each storing one or more of those partitions with ability to accept multiple formats. This is an essential requirement for Big Data systems to deal with unstructured data. Kafka [8] is suitable for building real-time streaming data routes that reliably pass data to systems or applications by running on a cluster of servers. The Kafka cluster stores and categorize streams of records in Topics, while these records consists of key, value, and timestamp.

C. Data Transformation and Storage

The data that has been streamed from the twitter will be in JSON format, so that data has to be modified and transformed to be fit into the HDFS file system. Transformation is done using the Apache SPARK and Apache HADOOP [9].

Apache spark is an open source project designed to perform in-memory cluster computation. Also, it provides an integrated framework to support for a different type of data processing requirements which include graph data, text data, and other sources such as batch and real-time data. In memory cluster computing feature of Spark increases the processing speed of Spark applications. Apache Spark framework supports wide range of workloads such as Machine-learning algorithms, Batch and Streaming applications, Interactive SQL queries etc. It supports multiple programming languages such as Java, Python, R and Scala. Also, it contains many built-in API's in all languages. Apache Spark with a whole package of features that support Real-time Streaming of data, Graph processing, Map Reduce, SQL queries, and Machine Learning libraries help to do advanced analytics. It is 100X faster in in-memory processing and 10X faster on disk processing services.

Hadoop is an open source project of apache foundation appropriate for storing and processing large volume and variety of data in cluster mode. Hadoop framework designed to store in distributed manner and cluster computation. It is written in java for management of huge data sets. While enormous organizations are using Hadoop to store and process large data sets in cluster mode, for example Google, Amazon etc. It consist three modules they are [9]:

1) HDFS: This component guarantees high performance and quick access to the information. It can store all sort of information like organized, semi organized and unstructured information

2) YARN framework: A Resource administration framework for handling resource requests and job scheduling in the cluster.

3) Map Reduce framework: A software programming model for processing large data sets in parallel manner. Development

D. Quarrying and Report generation.

The data that is stored in the HBase[10] is now acts as a resource for the report generation. The HBase contains the tweets data that has been stored by the real time streaming, now that data should be represented as the report for the authority. For generating report the data should be classified based on the issues and their location. We can rather say this process as quarrying, this can done by the powerful quarrying tool that can be made compatible with the hadoop and HBase is Apache phoenix [11]

Apache Phoenix is the open source trusted data platform for OLTP and operational analytics for Hadoop through well-defined industry standard API[phoenix documentation].Apache phoenix takes the SQL query, compiles it into a series of HBase scans, and orchestrates the running of those scans to produce regular JDBC result sets. Direct use of the HBase API, along with coprocessors and custom filters results in performance on the order of milliseconds for small queries, or seconds for tens of millions of rows. Apache Phoenix supports all standard SQL query constructs including SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY and etc. [11].

RELATED WORK

Understanding the problem and their solution we have exponentially worked in the repository's to find existing solution to chase their facility. We have come across many ideas, trends and technologies. The objectives of this paper will be contributing as a part of the smart city plan.

The Smart city is an urban area that uses different types of data collection mechanisms for managing the city assets in real time. The data collection from the citizens and sensing their emotion is important to know their needs and essentials. The City Pulse a large scale data analytics proposed by the Dan.et.al [12] on 2016 gives the answers to why and how the city should connect with the citizens.

Haewoon.et.al [13] on 2010 gives an idea for using the social media as a news media to express the social issues. And Rashid.et.al [14] on 2017 proposes how the social media can be used as a public emotion sensing tool. Trupthi.et.al [15] for discovering the trends in the public

The big data technologies play an important role in real time data acquisition from the social media. The big data management becomes a key challenge; Bijesh.et.al [16] on 2014 gives a better description of using reliable and row-bust technologies for managing the big data, he also suggest that Apache Hadoop is the most reliable big data technology

Farzana.et.al [21] and wu.et.al [20] has a method of implementing a Social media analytics on the big data for a demographic representation for different topics like Education, Politics, Entertainment and many more, which was helpful for us to aware of the analytics methods.

DESIGN AND IMPLEMENTATION

In this paper, we have built complete solution of streaming, processing, and analysing Twitter data which include many tools of Big Data technologies. The NIFI is used as the data injection tool. The NIFI architecture contains web server, Flow controller, Flow File Repository, content Repository, and provenance Repository. The web server runs on the JVM to host NiFis HTTP based command and control API. The Flow Controller is the brains of the operation; it provides the processor to run on and manages the resources. We utilized Apache Kafka to exchange data and Spark streaming for processing of in-coming data. And data collected in JSON format and processed in real-time using spark streaming. Hadoop is used to store and process the large datasets as Map Reduce job. Hadoop ecosystem contains hundreds of tools such as Spark, Kafka, Storm, Pig, Hive, Hbase, Oozie, Ambari, Mahout, Phoenix etc., can be used in different scenario. Also it supports different types of data like Structured, Semi-structured, and Unstructured. The information that's collected from social media will be stored in the distributed file system HBase. Then Apache Phoenix is the Quarrying tool that uses SQL queries to manage the big database. The data retrieved by the Phoenix will be used for Report generation. The figure 1.0 showes the full architecture and data flow from one to another.

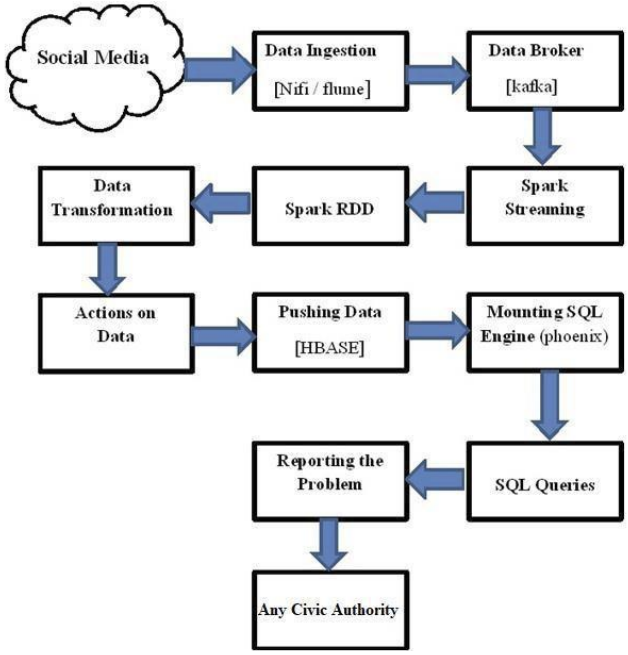


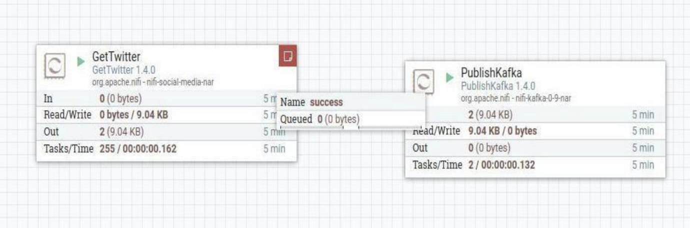
Figure 1: Complete Architecture of the Automation system

All these components are used to Stream the data in the real time and to analyse the real-time and historical data. The design and implementation of each component is discussed below.

Data Ingestion: The process of data ingestion is done by using the Apache Nifi. Apache Nifi provides an easy to use, powerful and reliable system to process and distribute the data over several resources. Apache Nifi [7] is used for routing and processing data from any source to any destination. The process can also do some data transformation. It is a UI based platform where we need to define our source from where we want to collect data, processors for the conversion of the data, a destination where we want to store the data. Each processor in Nifi has some relationships like success, retry, failed, invalid data, etc. which we can use while connecting one processor to another. These links help in transferring the data to any storage or processor even after the failure by the processor.

Here in our project we create two processors "GetTwitter" and "PublishKafka". GetTwitter extracts the data from the twitter whereas PublishKafka gives the data to the Kafka broker. Also there will be repositories like flow file, content and provenance , flow file repositories store status of active flow file , content repositories store the actual content and provenance repositories contains the events that are happening.

Nifi's "GetTwitter" processor uses Twitter Streaming API to fetch tweets. NiFi also act as a Filter in data ingestion. Below diagram shows the NiFi processor implementation.



GetTwitter
GetTwitter 1.4.0
org.apache.nifi - nifi-social-media-nar
In 0 (0 bytes) 5 m
Read/Write 0 bytes / 9.04 KB 5 m
Out 2 (9.04 KB) 5 min
Tasks/Time 255 / 00:00:00.162 5 min

Name success
Queued 0 (0 bytes)

PublishKafka
PublishKafka 1.4.0
org.apache.nifi - nifi-kafka-0-9-nar
2 (9.04 KB) 5 min
Read/Write 9.04 KB / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 2 / 00:00:00.132 5 min

Figure 2: NiFi Processor Design Architecture

Data Broker: Apache Kafka [8] will be the data broker between NiFi and Spark Streaming. The architecture of the kafka is as below

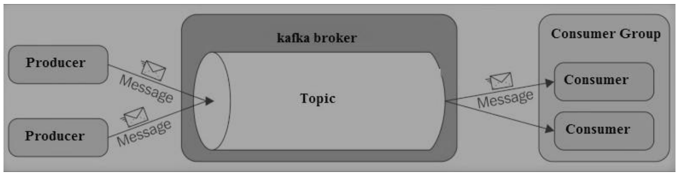


Figure 3: Kafka Architecture

The components of Kafka architecture are

- □ Kafka topic - A topic is a classification or encourages name to which records are distributed. Kafka topics are dependably multi-supporter; that is, a topic can be a zero, one, or more consumers that subscribe to the data kept in touch with it.
- □ Producers - It is an API we can write a program according to our requirements client to publish messages to a specified Kafka topic.
- □ Message Broker - It is one of the Kafka component responsible for the coordinating and communicating all the connected data sources and destinations.
- □ Consumers - It is client side program to consume messages from a Kafka topic.

We are using "tweets" as the topic name which is divided into two partitions and stored in two nodes. And we can create the required number of Brokers that provides un-interrupted service to the consumer. The fault tolerance and back up mechanism is implemented using replication factor for a topic. The Apache NIFI act as a client to produce content in JSON format the producer program is configured in such a way that, it fetches the data from the NiFi to the topic "Tweets". The topic acts as mediator between producer and consumer to communicate each other. The consumer collects data from a topic for serializing. The topic utilized as database to hold data because we don"t like to lose data for particular duration. But, in our system we configured Kafka server to hold data for a specified period of time. To the consumer transfers the data to Spark Streaming.

Spark Streaming and Spark RDD: Apache spark is an open source project designed to perform in-memory cluster computation. Also, it provides an integrated framework to support for a different type of data processing requirements which include graph data, text data, and other sources such as batch and real-time data. In memory cluster computing feature of Spark increases the processing speed of Spark applications.

Apache Spark framework supports wide range of workloads such as Machine-learning algorithms, Batch and Streaming applications, Interactive SQL queries etc. It supports multiple programming languages such as Java, Python, R and Scala. Also, it contains many built-in API's in all languages. Apache Spark with a whole package of features that support Real-time Streaming of data, Graph processing, Map Reduce, SQL queries, and Machine Learning libraries help to do advanced analytics. It is 100X faster in in-memory processing and 10X faster on disk processing.

Apache Spark is used widely for large data processing. Spark can process the data in both i.e. Batch processing Mode and Streaming Mode.

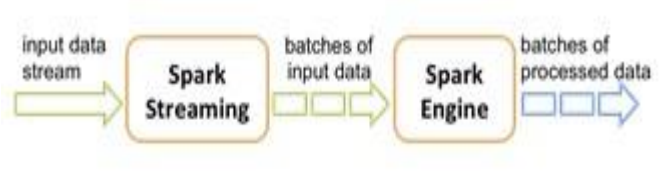
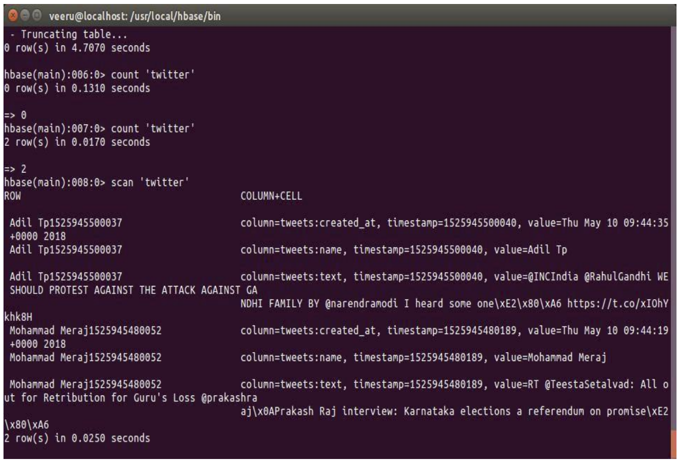


Figure 4: Illustrates the streaming process.

As the data generates on the topic, created pipeline is utilized to extract and stream the Tweets. Spark Streaming splits data as RDD (resilient distributed data) also defined as tiny batches of real-time clickstream data. The time batch in our system is set to 10 seconds. The Spark Streaming will starts work on each batch of data and process it using Spark engine and send them to Big data storage HBase.

Data Storage: In this paper we will be utilizing the HBase as the storage tool. HBase is a type of "NoSQL" database that means the database isn't an RDBMS which supports sql as its primary access language, Technically speaking, Hbase is really more a "Data Store" then "Data Base". HBase has many features which supports both linear and modular scaling. HBase clusters expand by adding RegionServers [10] that are hosted on commodity class server. In HBase the data is stored as tables, which consist of multiple Rows and columns. Columns consist of column family and column Qualifier, which are further divided into cell and timestamp. We have created a table as "twitter" and 3 cells for JSON twitter data attribute they are created_at(date and time the tweet has created), name(name of the tweeter), text(text of the tweet). The information generated by the spark is stored in the HBase in the form of salted data. Salted data means the byte data in a cryptographic format.

The below figure shows how the data will be presented in the HBase



```
 - Truncating table...
0 row(s) in 4.7070 seconds

hbase(main):006:0> count 'twitter'
0 row(s) in 0.1310 seconds

=> 0
hbase(main):007:0> count 'twitter'
2 row(s) in 0.0170 seconds

=> 2
hbase(main):008:0> scan 'twitter'
ROW                                       COLUMN+CELL

 Adil Tp1525945500037                     column=tweets:created_at, timestamp=1525945500040, value=Thu May 10 09:44:35
 +0000 2018
 Adil Tp1525945500037                     column=tweets:name, timestamp=1525945500040, value=Adil Tp

 Adil Tp1525945500037                     column=tweets:text, timestamp=1525945500040, value=@INCIndia @RahulGandhi WE
 SHOULD PROTEST AGAINST THE ATTACK AGAINST GA
                                          NDHI FAMILY BY @narendramodi I heard some one\xE2\x80\xA6 https://t.co/xIOhY
khk8H
 Mohammad Meraj1525945480052              column=tweets:created_at, timestamp=1525945480189, value=Thu May 10 09:44:19
 +0000 2018
 Mohammad Meraj1525945480052              column=tweets:name, timestamp=1525945480189, value=Mohammad Meraj

 Mohammad Meraj1525945480052              column=tweets:text, timestamp=1525945480189, value=RT @TeestaSetalvad: All o
ut for Retribution for Guru's Loss @prakashra
                                          aj\x0APrakash Raj interview: Karnataka elections a referendum on promise\xE2
\x80\xA6
2 row(s) in 0.0250 seconds
```

Figure 5: Data present in HBase.

CONCLUSION AND FEATURE ENHANCEMENT

Transformation from normal city to a Smart city is the main aspect of the city development. Responsiveness of the city for its citizen's problem is a feature of the smart city. So a Real-Time automation tool is needed to become responsive for the citizens. Our paper contributes the automation tool to become a reality. Since citizens are connected to social media, it becomes easy to get the opinions of the citizens about the civic problems. The system is real time in nature and hence, the problems of the people can be listened instantly and action can be taken by the authorities. This system will help in uplift of the society. The existing system is only concerned about the civic issues and we can enhance this project to be working on different domain like waste management, city development, government event feedback, etc. the ingestion tool what we have created is powerful and real time and different data analytics can be implemented on that to make multipurpose results based on the requirements.

REFERENCES

[1].http://bengaluru.citizenmatters.in/bangalore-roadspothole-problem-solutions-21743

[2].www.civiced.org/papers/articles_mb_june00.html

[3].http://www.smartcityplanning.co.jp/en/approach/04.html

[4].https://www.internetsociety.org/internet/history-internet/brief-history-internet/

[5].https://www.techopedia.com/definition/4941/facebook

[6].https://www.lifewire.com/what-exactly-is-twitter-2483331

[7].https://nifi.apache.org/docs.html

[8].https://kafka.apache.org/documentation/

[9].https://spark.apache.org/documentation.html

[10].https://hbase.apache.org/book.html

[11] https://phoenix.apache.org/

[12].Dan puiu CityPulse: "Large Scale Data Analytics Framework for Smart Cities" IEEE-2016

[13].Haewoon Kwak “What is Twitter, a Social Network or a News Media?”, IEEE-2010

[14].Rashid Kamal “Real-time Opinion Mining of Twitter Data using Spring XD and Hadoop”, IEEE -2017

[15].M.Trupthi “SENTIMENT ANALYSIS ON TWITTER USING STREAMING API”, IEEE-2017”Live Data Streaming System for Social Media to Address Civic Issues Using Big Data Technology

[16].Bijesh Dhyani “Big Data Analytics using Hadoop”, IEEE Dec 2014

[17].Farzana Shaikh “Social Media Analytics Based on Big Data”, IEEE-2017

[18].Wu Wen, Wang Wei,” Social Media as Research Instrument for Urban Planning and Design”, IEEE-2016

[19].Ramanna Hanamanthrao, Department of CSE, SIT Tumkur,” Real-Time Clickstream Data Analytics and Visualization”, IEEE-2016

[20].Babak Yadranjiaghdam “Developing a Real-time Data Analytics Framework For Twitter Streaming Data”, IEEE-2017

[21].Alessandro Ferreira Leite “Big Data Management and Processing in the Context of the System Wide Information Management”, IEEE-2017

[22].Thanga Jawahar Kalidoss Engineering and Industrial Services, Tata Consultancy Services Chennai, India”Disaster management system leveraging the emerging digital technologies”

[23].http://www.inurture.co.in/evolution-of-web-technologies-and-the-future/

[24].https://www.scss.tcd.ie/owen.conlan/CS7062/1_Web_Technologies_Handout.html

[25].https://www.digitalunite.com/guides/social-networking-blogs/google-plus/what-google-plus

[26].https://www.sharcnet.ca/help/index.php/Big_Data

[27].https://www.xenonstack.com/blog/data-engineering/data-ingestion-using-apache-nifi-for-building-data-lakes-twitter-data.

[28].https://spark.apache.org/docs/2.2.0/rdd-programming-guide.html#overview.

[29].https://hortonworks.com/apache/hbase/