International Journal of Distributed and Cloud Computing
7 (2) December 2019, 01-06
http://www.publishingindia.com/ijdcc

Security Mechanism for Twitter Data Using Cassandra in Cloud

Sangeeta Gupta[1] and Nikitha Kumari[2*]
[1]Associate Professor, CSE, Chaitanya Bharathi Institute of Technology, Hyderabad, India.
Email: ss4gupta13@gmail.com
[2]M.Tech 2nd Year, CSE, Vardhaman College of Engineering, Hyderabad, India.
Email: nikithakumari408@gmail.com
*Corresponding Author

Abstract: Most popular social networking and the broadly utilized site is Twitter, where clients share their opinions on posts through tweets. Every tweet has a limit of 140 characters. Tweets are in the form of structured and unstructured data which consist of characters, alphanumeric, special symbols, etc. Every single tweet may also contain some unwanted characters. In this work, twitter data are chosen to perform an analysis. Pre-handling methods are utilized to expel undesirable characters from tweets. Also, to enhance security, the encryption module is created, and the information is then put away in Cassandra. Just the individuals who hold the key can unscramble the information. The whole information is then put away in the cloud for upgrading the security.

Keywords: Cassandra, Encryption, Share market, Text pre-processing, Twitter.

I. Introduction

Nowadays, social media has become the most important media for the source of communication and gaining information about the latest news and events that are being posted on the social media. In recent years, microfilming, which has turned out to be basic [1], and a mainstream webpage for all the online users and billions of users share their opinion on different aspects on trending websites like Twitter, LinkedIn, Facebook, etc. [2]. Among all the social sites, most and broadly utilized site is Twitter where the clients offer and post their supposition as per their advantage, for example, sports, instruction, transport examine territory, therapeutic and wellbeing, legislative issues, etc.

In this research, we have focused on all the tweets related to share market information where these are sources for the companies to raise the funds and for the shareholders to buy the shares and earn partnership and thus grow their business. In other words, share market is the place where the shares are bought and sold. There are two types of share markets: primary and the secondary. In primary market, securities are offered to the general public for subscription for raising capital or fund associated. The secondary market is an equity commerce venue where pre-existing pre-issued securities are traded among investors.

In real-world scenario encryption is an important aspect. Encryption represents a procedure used to make touchy information progressively secure. There are a few present-day kinds of encryptions used to secure touchy electronic information, for example, email, documents, envelopes, and whole drives.

This paper is organized as follows: Section II consists of the background work. Section III is about the comparative study. Section IV describes the proposed work. Section V consists of and result & discussion. Section VI concludes the work by throwing a light onto the future direction.

II. Related Work

This part of the section is related to several related works on the pre-processing techniques. Here, the twitter-data-based works are discussed. The authors in [3] have examined results of pre-processing techniques on twitter knowledge for the defence wall of assumption grouping. They have collected the tweets that carry with them symbols, abbreviation, social-tagging, and uncorrected words. And also removing the URLs, user tags, hashtags, accentuation and significance of slang words, spelling adjustment were distinguished. Support Vector Machine (SVM) was used to carry out the experiment.

The authors in [4] focused on the effects of pre-processing techniques by using six pre-processing techniques with five twitter datasets and two models of features and four classifiers, increased by the strategies of replacing negations, expanding acronym, removing URLs, stop words and numbers.

The authors in [5] collected data from many Indonesian official twitter accounts which are always updated regarding football news. Latent Dirichlet Allocation (LDA) is used for topic modelling and obtaining many topics such as pre-match analysis, football club achievements, live match's update, etc.

The authors in [6] explored and differentiated between the various forms of existing NoSQL databases which are classified into four classifications like MongoDB for document stores, Neo4j for graph databases, Cassandra for wide segment stores, and Redis for putting away key-esteem. These databases look at a data model using many variables. By examining non-utilitarian choice, it's been discovered that a record store might be utilized an adaptable way using JSON positioning. Also, the columns are utilized in terms of versatility of semi-organized information.

The authors in [7] developed the ongoing Twitter information warehousing using no SQL database by comparing its querying performance using online dashboard technique. It is observed from the results that Cassandra performs well for a vast set of data, but it is significantly slower for less amount of information.

The authors in [8] have overcome the problems identified with the use of TF-IDF mechanism that fails to handle large datasets efficiently. They also present an adaptable methodology which permits an ongoing proposal of the client on their tweets where the users share similar interests with similar terms that showcases quick correlation and positioning. Compact binary user profile using twitter data was built by the authors using co-occurrence matrix, which was used as an input for Sim hash. The results obtained using the Hash graph are found to be similar to the one obtained with a user profile.

The authors in [9] planned a unique social mindful data stockpiling and replication that considers the OSN client spatial circulation, companionship framework, and intuitiveness between the client and his companions to diminish the client's normal postponement for data. The developed program is contrasted with Cassandra by using Anycast steering and numerical improvement calculations for data stockpiling. The capacity assignment downside in OSN is numerically displayed as a partner in nursing enhancement and number advancement is utilized to search out the issue. The best hypothetical answer is obtained by using MATLAB apparatus to mimic a social network with several data centres by comparing both online social network and Cassandra. Based on the comparisons carried out, Cassandra is found to be better to assess online social networking paradigms.

The authors in [10] compared SQL database and no SQL database Cassandra which handles a large amount of data in an efficient and cost-effective manner. Cassandra is an open source technology. The authors compared 'Oracle and Cassandra by fetching the students' information in terms of writing queries. The results show that Cassandra takes less time compared to Oracle for operations of over 40,000 records.

The authors in [11] compared the overall performance and measurability analysis of enormous collection of data stores, like NoSQL – Cassandra and MySQL. It was observed that Cassandra performed poorly in terms of recovery of records and circle utilization

III. Comparative Study

Table I describes the comparison of previous works where it shows that the authors in [12] evaluated their experimental setup by collecting the tweets related to Kiran Bedi and Arvind Kejriwal from the twitter API, and executed by using technique named Word Sense Disambiguation. In [13], by using the Kaggle dataset, the authors extracted tweets related to airlines services and conducted experiments using machine learning algorithms. Similarly, by using different machine learning algorithms, the authors in [14] carried out an experimentation using two datasets such as SST and Sem-eval dataset, and obtained the results. However, in the aforementioned works, none of the authors discussed the robustness of the proposed techniques, when applied to unstructured data. Also, there is no

Table I: Comparative Analysis Table

Title Name	Methods Used	Dataset Used	Collected Tweets Related to	Pre-processing Technique Used
Forecast of the result by improved feeling investigation on Twitter information utilizing word sense disambiguation [12]	Word sense disambiguation.	Twitter API is been used. Sentiment classification of word sense.	Kiran Bedi and Arvind Kejriwal	Removing URLs and negation handling
A notion order arrangement of Twitter information for aircrafts administration examination [13]	Decision tree' SVM, Gaussian naïve Bayes, Ada boost, KNN, logistic regression.	Kaggle dataset is been used to extract tweets.	US Airlines services	Tokenization, Stop words and lemmatization
An examination of pre-preparing systems for Twitter assumption investigation [14]	Calculated relapse calculation, Bernoulli innocent Bayes, direct svc calculation.	Sentiment strength twitter dataset and Sem-Eval dataset.	Positive, negative, Neutral tweets	15 pre-processing techniques were used

focus on the size of the dataset and security aspects.

In this paper, we examined and compared the load time in Cassandra with a variety of tweets related to share market consortium by using twitter API.

IV. Proposed Work

This section is designed to generate a huge amount of data from the famous social media site, i.e., Twitter. Tweets collected to implement this work are related to share market consortium. Sometimes, tweets are made out of a boisterous, inadequately organized sentence, unpredictable with articulation, fragmented, non-lexicon terms, and many short structures. To expel all these uproarious terms from the tweets, pre-handling techniques are used.

Tweets with respect to stock market analysis are used by people who greatly utilize social media as a source for bidding based on the quotations of the stock prices, which if presented in plain text format (as tweets are publicly accessible) would pave the path for the intruder to intercept the conversation among bidders and mis-use the information that may yield great losses to the company. For this reason, an attempt is made to encrypt the tweet messages that possess stock quotations and the person who has the key (one among the bidding group) can only decrypt it.

Also, the work that is discussed in the literature survey section considers either application of pre-processing techniques or an encryption mechanism, but not both aspects as presented in the proposed work. This specifies the significance and necessity to proceed ahead with the proposed mechanism.

The method of converting the data to something which a computer can understand is defined as pre-processing which aims at filtering all the unnecessary and irrelevant data. In this paper, three pre-handling techniques are used to expel the useless data from tweets, i.e., replacing slang and abbreviations, removing numbers, removing punctuations and stop words. Fig. 1 describes the whole process flow in step by step procedure.

Replacing Slang and Abbreviations: Tweets written by the users contain many short forms and slangs. A slang may be a kind of language that consists of phrases and words. Shortening of the words is known as abbreviation. Some of the examples are ‘omg’, ‘TBT’, and ‘ty’, which are phrases and are commonly used while tweeting which literally means ‘oh my god’, ‘Throwback Thursday’, and ‘thank you’, respectively.

Example:

Consider the following Tweet:

“OMG. This stand up is hilarious. I’m dying.”

After applying pre-processing, the processed tweet is obtained as:

“Oh my god. This stand up is hilarious. I’m dying.”

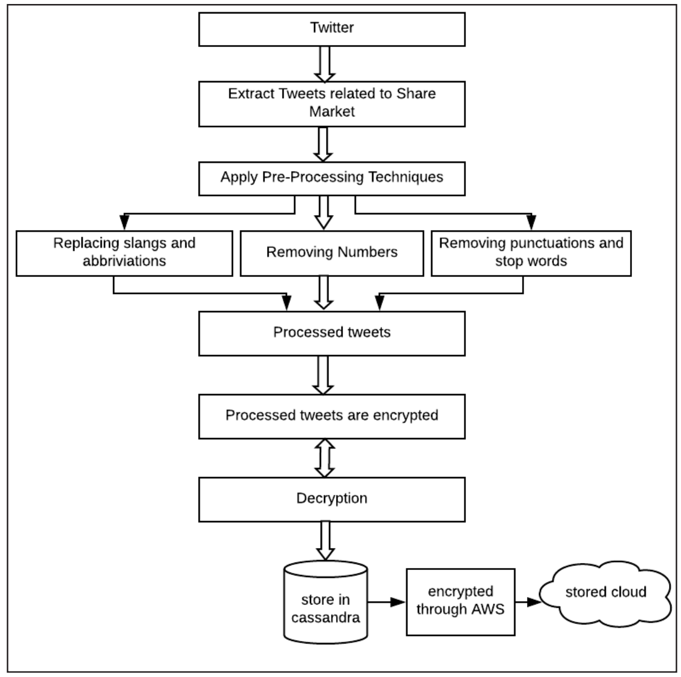

Fig. 1: Architecture System for Proposed Work

Removing Numbers: Tweets consists of words such as: ‘L8, B4, G2B, 5N’ which contain a number in it, which literally means ‘Late, Before, Go to Bed, Fine’, respectively. This process is done after replacing the slang because some slang words like ‘gr8’ (great) that also contain the numbers.

Example:

Consider the following tweet:

excited 2 announce in-steam payments on twitter! Check it out R8 here in this tweet! Rbn.co\cba7a2.

These are being transformed after applying pre-processing step to:

excited to announce in-steam payments on twitter! Check it out right here in this tweet! Rbn.co\cba7a2.

Removing Punctuations or Stop Words: Most commonly used words in the languages such as ‘a’, ‘but’, ‘an’, ‘in’, ‘and’, ‘at’ are stop words. The sentences which consists of punctuation marks such as ‘!’, ‘.’, ‘comma (,)’, ‘?’, denote the existence of few sentiments which suggest associate, intensive, positive or the negative sentiment. Therefore, when we have a tender to take away punctuation marks, it will decrease the efficiency.

Example: Consider the following Tweet:

The victim is on record, on camera. How many major media outlets except @MyNation have covered this?

After applying the pre-processing step, the text is as follows:

The victim is on record on camera How many major media outlets except @MyNation have covered this

Encryption: It is the method of changing the data or the information into the code to stop unauthorized access [12]. In this paper, encryption is done when all the tweets are collected. Pre-processing step is applied to the tweets after which the processed tweet is encrypted. This progression is done to fortify the security of tweets where the clients may speak about the quotes of shares they have purchased, which may give an entry point to the opposition to hack and extract the confidential data.

Decryption: Encrypted data are converted back to the text data which can be read easily. Data can be decrypted only the person who has the key to access.

AWS Service: The encrypted data are further encrypted and results are stored in cloud through AWS services. The final set of text obtained is loaded through AWS (Amazon Web Service) account in a secure way to cloud.

AWS is secured by generating a key-pair for the server or any other virtual instance that a user is utilizing for their personal or commercial purpose. However, if the user forgets to logout from the AWS domain, it paves path for the intruders to damage confidential information by gaining access to the server. For this reason, there is a need to strengthen the security by further encrypting the data before being loaded into the AWS via Cassandra database thereby generating a new key pair for every tuple to be stored.

In AWS, a key pair is generated while the creation of virtual machine (VM). This VM can be used to load data and operate on huge data sets. But, securing data with key pair alone does not strengthen the security. Hence, in this work, the key pair generated is appended with the pre-processed tweets at random positions, making it tough for the intruder to access the secret data.

Also, the data are encrypted before storing in Cassandra and also before storing in cloud ensuring a double-level security.

V. Results and Discussion

The work is developed using python 3.5, Cassandra 3.0, and the twitter development account is used to extract the tweets. Windows 10 operating system, with the RAM capacity of 4 GB, Internet bandwidth 10 mbps, and the Processor used is core 3. The following algorithm is developed as a part of the proposed work as mentioned below:

Algorithm

Step 1: Extracting the tweets from the account.

Step 2: Applying the pre-processing techniques to tweets.

Step 3: Applying techniques like removing numbers, replacing slang and abbreviations, removing punctuations and stop words from tweets.

Step 4: A processed tweet is been obtained.

Step 5: The processed tweet is encrypted and then stored in Cassandra.

Step 6: A key is used to decrypt the encrypted data.

Step 7: The final set of text obtained are all loaded through AWS (Amazon Web Services) in a secure way to cloud.

For every step, the time is been noted for loading the tweets in Cassandra as shown in the Table II, III, IV, V and VI.

A. Tweets Load Time in Cassandra for Plain Text

In Table II, plain tweets without applying any techniques are been inserted into Cassandra. The time is recorded to stack the tweets in Cassandra for inserting records in the range 1 through 500 as shown in Table II.

Table II: Load Time in Cassandra for Plain Tweets

No. of Tweets	Load Time in Cassandra (ms)
1	6
10	13
100	209
200	300
300	402
400	497
500	504

B. Load Time for Encrypting (Ciphertext) the Tweets

After collecting the plain text-based tweets they are encrypted and inserted into Cassandra in the range from 1 through 500. The time taken to load the tweets in Cassandra is noted in milliseconds as shown in Table III.

Table III: Load Time in Cassandra for Encrypting Ciphertext

No. of Tweets	Load Time in Cassandra (ms)
1	9
10	90
100	160
200	280
300	477
400	529
500	600

C. Calculating the Load Time for the Tweets After Applying Pre-processing Techniques With Encryption

The processed tweets are encrypted by converting them into ciphertext which can't be read. The entire data are stored in Cassandra with enhanced security. Records in the range 1 through 500 were inserted into Cassandra and load time for each insertion is been noted in milliseconds as shown in Table IV.

TABLE IV: LOAD TIME FOR TWEETS WHEN PROCESSED TWEETS ARE ENCRYPTED

No. of Tweets	Load Time in Cassandra (ms)
1	15
10	87
100	109
200	283
300	402
400	589
500	600

D. Calculating Time Taken for Retrieval Process

The retrieval time for the tweets within a range from 1 through 500 tweets is recorded as shown in Table V.

TABLE V: LOAD TIME IN CASSANDRA FOR TWEETS RETRIEVAL

No. of Tweets	Load Time in Cassandra (ms)
1	1
10	9
100	81
200	158
300	239
400	427
500	400

E. Calculating the Time for Tweets When Decrypted

Tweets are now in the form of ciphertext which is not understandable to the user as well as the machine. So to read the ciphertext, decrypt method is applied, which converts the ciphertext to the plain text. In Table VI, the load time for cipher tweets to convert into the plain tweets, which can be readable, is presented. The time taken to decrypt the key is been calculated for a records ranging from 1 through 500 as noted in Table VI.

TABLE VI: LOAD TIME IN CASSANDRA FOR DECRYPTION OF TWEETS

No. of Tweets	Load Time in Cassandra (ms)
1	69
10	103
100	225
200	328
300	450
400	550
500	564

F. Performance Comparison of the Load Time for Tweets in Cassandra Versus Encrypted Pre-processed Text

Fig. 2 shows the load time performance for the tweets when inserted in Cassandra for the plain text (ciphertext) when encrypted and the tweets encrypted after applying the pre-processing steps. Even though the plain tweet is encrypted for 500 tweets, the time taken to encrypt the tweets is same as with the application of pre-processing technique.

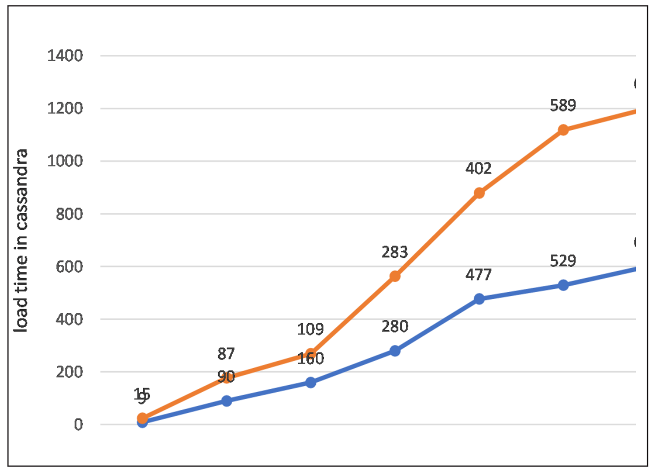

Fig. 2: Performance for the Processed Tweets Encryption and Plain Tweet Encrypted

VI. CONCLUSION AND FUTURE WORK

This paper depicts the load time performance calculated for five different processes when loaded into the Cassandra. When 500 tweets are inserted, their load time is approximately 504 milliseconds for the plain text. While the time taken to encrypt 500 tweets is approximately 600 milliseconds. Processed tweets, when encrypted, record a load time of approximately 600 milliseconds (for 500 tweets). In retrieval process, for 500 tweets it takes approximately 400 milliseconds in Cassandra. While in case of the tweets to be decrypted, the load time in Cassandra is approximately 564 milliseconds. Though the time taken to encrypt as well as decrypt the tweets is more than the time to deal with the plain text, the proposed method strengthens security of the tweets. It is observed that the time taken to load 500 tweets is nearly same for processed tweets encryption and plain tweets encryption as well, which can be further reduced as a part of further extension of the proposed work.

The future work can also be focused to reduce the time to encrypt and decrypt and also to increase the number of tweets dealing with the scalability aspect of Cassandra in the cloud.

REFERENCES

[1] L. Williams, C. A. Bannister, M. Arribas-Ayllon, A. Preece, and I. Spasic, "The role of idioms in sentiment analysis," *Expert Systems with Applications*, Elsevier, vol. 42, no. 21, pp. 7375-7385, 2015.

[2] A. Dalmia, M. Gupta, and V. Varma, “Twitter sentiment analysis the good, the bad and the neutral,” *9th International Workshop on Semantic Evaluation (SemEval)*, pp. 520-526, 2015.

[3] J. Zhao, and X. Gui, “Comparison research on text pre-processing methods on twitter sentiment analysis,” *IEEE Access*, vol. 5, pp. 2870-2879, 2017.

[4] T. Singh, and M. Kumari, “Role of text pre-processing in twitter sentiment analysis,” *Procedia Computer Science*, vol. 89, pp. 549-554, 2016.

[5] A. F. Hidayatullah, E. C. Pembrani, W. Kurniawan, G. Akbar, and R. Pranata, “Twitter topic modelling on football news,” *2018 3rd International Conference on Computer and Communication Systems*, 2018.

[6] A. Gupta, S. Tyagi, N. Panwar, S. Sachdeva, and U. Saxena, “NoSQL databases: Critical analysis and comparison,” *International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, 2017.

[7] M. R. Murazza, and A. Nurwidyantoro, “Cassandra and SQL database comparison for near real-time twitter data warehouse,” *“International Seminar on Intelligent Technology and its Application (ISITIA)*, 2016.

[8] J. Subercaze, C. Gravier, and F. Laforest, “Real-time, scalable, content-based twitter user recommendation,” *Journal: Web Intelligence*, vol. 14, no. 1 pp. 17-29, 2016.

[9] M. E. Elsaid, A. Shawish, and C. Meinel, “A social aware approach for online social networks data allocation and replication,” *European Conference on Electrical Engineering and Computer Science (EECE)*, 2017.

[10] S. Anand, P. Singh, and B. M. Sagar, “Working with cassandra database,” *Information and Decision Science*, pp. 531-538, 2018.

[11] S. Gupta and N. Narasimha, “Performance evaluation of NoSQL-cassandra over relational data store-MySQL for bigdata,” *International Journal of Technology*, vol. 6, no. 4, 2015.

[12] R. Jose, and V. S. Chooralil, “Prediction of election result by enhanced sentiment analysis on twitter data using word sense disambiguation,” *2015 International Conference on Control, Communication & Computing India (ICCC)*, 2015.

[13] A. Rane, and A. Kumar, “Sentiment classification system of twitter data for US airline service analysis,” *2018 42nd IEEE International Conference on Computer Software & Applications*, 2018.

[14] D. Effrosynidis, S. Symeonidis, and A. Arampatzis, “A comparison of pre-processing techniques for twitter sentiment analysis,” *21st International Conference on Theory and Practice of Digital Libraries (TPDL)* 2017.