A Novel Approach for Collaborative Spam Filtering to Protect Mail box

Haseeba Yaseen[1], Adithya D A[2], Ashwini M[3], Chandan S[4], Dhanesh P Kudalkar[5]

[1,2,3,4,5]Department of Computer Science and Engineering
Global Academy of Technology
Bangalore, 560098,India

Abstract

Spam has turned into the stage of decision utilized by digital law breakers to spread malignant payloads, for example, infections and Trojans. In view of this, the issue of early recognition of spam causes has been addressed. Communitarian spam location procedures can manage huge scale email information contributed by diverse sources, they have the prominent topic of requiring exposure of email content. Separation upkeep hashes are one of the normal arrangements applied for saving the security of email content while allowing message grouping for spam recognition. Anyhow, separate safeguarding hashes are not adaptable, along these lines making massive scale synergistic arrangements hard to represent. Spamdoop, a Big Data security safeguarding community oriented spam discovery stage based over a standard Map Reduce office has been projected. Spamdoop employs a very parallel encoding system that empowers the recognition of spam battles in aggressive circumstances. The outline has been judged by the execution, by utilizing a massive manufactured spam base and prove that the system performs positively against the creation and conveyance overhead of current spam age devices.

Keywords: Spamdoop, Hashing, Hadoop, HDFS.

1. Introduction

The massive unsolicited emails sent is normally said to be spam. It is quite not possible to describe the word spam more accurately. In all of its forms, a spam is considered as one of the difficult challenges of the linked generation. This in return drove a massive amount of research towards contradicting it. The mentioned effort led to the gradual decline of spamming activities which made many to end up believing spam is no longer a threat. However, recent thorough studies and statistics have concluded otherwise. In fact, about 66.34% of all e-mails sent worldwide are considered spam e-mails according to Kaspersky's Spam and Phishing Statistics for the first quarter of 2014 which leads us to conclude that it is still an evolving phenomenon and is still an active cyber threat E-mail these days has become a popular and most favored means of communication over the Internet. The extent of email received and the amount of spam is constantly growing. Spam mails are defined as electronic messages posted to thousands of recipients usually for the purpose of advertisement or profit. Some of the spam emails modify as a phishing emails seeking users' confidential data and accessing their bank accounts for financial fraud. Phishing is a fraudulent attempt aimed at capturing sensitive information such as usernames, passwords and credit card details by impersonating a trustworthy entity in an electronic communication. Apart from examining the attachments, the content alone may also be prone to word de-obfuscation technique to fool the spam filters [9].

2. Influence of spam/phishing attacks

Spear Phishing attacks are the new trends in spam landscape which targets the executives and higher officials whom have access to proprietary company data and corporate banking credentials. These confidential data are obtained from them by forwarding spoofed phishing mails and take control over the entire organization and may also use their identities to obtain information from the other users. The reports from Trend Micro in July 2011 states that out of the total volume, the average share of spam with malicious attachments 23 per day was 2.14 percent. Spam without attachments, including messages that have been embedded with malicious links, can be further classified into job, medical and commercial, and scam spam. For the purpose of validating the performance of Spamdoop, we replicated a spammer's behavior by constructing a spam campaign generator that imitates a commercially available spamming tool.

3. Related Work

Distance-Preserving Hashing: Distance-preserving Hashing has been utilized for preserving the confidentiality of the contents of the email. They are effective in masking the effects of hash busting and hence they offer various advantages over traditional hashing techniques [1] such as maintaining the Integrity of the Specifications.

Downsides:

•This requires the removal of calculation by guaranteeing that the yield isn't effectively influenced by minor adjustments.

The above system suffering from drawbacks like,

•No Explicit control of parallelization.

•These approaches don't bother about security of messages.

•The authors of these methodologies have announced that their calculation may take quite a while.

Collaborative Privacy Aware Spam Detection. Distributed Checksum Clear-Housing has been deployed in the sphere of Collaborative privacy Aware Spam Detection. It involves sharing of hashes by the participants. The number of appearances of analogous email are counted and the suspicious ones get tagged as a spam. “Shingles”, i.e., distance-preserving fingerprints have been also used to identify similar emails. The downside of the above methods is that they are dependent on computing distance between shingles for identifying spams, also they haven’t been thoroughly tested [1].

Drawbacks: The security properties are not completely tried and thus are not suggested for private information sharing.

4. Methodology Used

In this section the fundamental methodologies are highlighted

4.1 The Spamdoop Platform

Spamdoop[9] is a stage enabling different elements to team up in early recognition of mass spam battles. Our stage additionally fulfils the security necessities of members. A review of Spamdoop engineering is depicted in figure 1 in the following segment which features the following key segments of the framework:

•The Obfuscator: The Obfuscator is mainly employed for the purpose of encoding the contents of the e-mails which will allow the parallel processing of the spam without the cost of revealing the contents of emails.

•The Parallel Classifier: The Parallel Classifier is used for the classification of the emails by utilizing the properties of encoding. This ensures routing messages similar to each other in the same buckets.

•The Anomaly Detector: The Anomaly Detector detects whether a certain email corresponds to a spam or not. The detection process depends on analyzing the size of the buckets along-with their rate of growth.

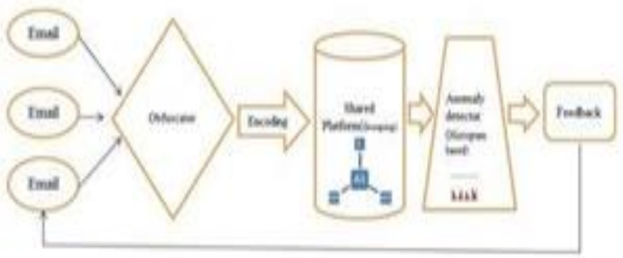


Fig.1: Spam Detection Process

4.2 The MapReduce Technique

Applications frequently require more resources than what are available on a conventional inexpensive machine. Many organizations find themselves with business processes that no longer fit on a single cost effective computer. A simple but expensive solution is to buy specialty machines having a lot of memory and high CPU Power. This solution scales as far as what is supported by the fastest machines available, but usually the only limiting factor is the budget. An alternate solution is building a high availability cluster. Such a cluster typically endeavors to look like a single machine, and usually requires very specialized installation and administration services. Most of the high- availability clusters are proprietary and high priced. Thus an economical solution for acquiring the necessary computational resources is cloud computing. A common pattern is to have bulk data that is to be transformed by processing each data item which are essentially independent of one another; that is, using a single-instruction multiple-data (SIMD) algorithm. Hadoop provides an open source framework for cloud computing, as well as a distributed file system. Hadoop supports the Map Reduce model, introduced by Google as a method of solving a class of pet scale problems with large clusters of inexpensive machines.

The model is based on two distinct steps for an application:

•Map: An initial ingestion and transformation step, in which individual input records can be processed in parallel.

•Reduce: an aggregation or summarization step, in which all associated records, must be processed together by a single entity.

The core concept of Map Reduce in Hadoop is that the input may be split into logical chunks, and each chunk is initially processed independently, by a map task. The results of these individual processing chunks can be physically divided into distinct sets and then sorted. Each sorted chunk is passed to a reduce task.

A map task may run on any compute node in the cluster, and multiple map tasks may be running in parallel across the cluster. The map task is accountable for the transforming of the input records into key/value pairs. The output of all the maps is partitioned, and each partition is sorted. There'll be a partition for each of the reduced tasks. Each partition's sorted keys and the values associated with the keys is then processed by the reduce task. There may be multiple reduce tasks running in parallel on the cluster. The application developer needs to provide the following four items to the Hadoop framework: the class that will read the input records and transform them into one key/value pair per record, a map method, a reduce method, and a class that will transform the key/value pairs that the reduce method outputs into output records. The Hadoop Map Reduce framework needs a shared file system. This shared file system isn't required to be a system-level file system, as long as there is a distributed file system plug-in available to the framework. When HDFS is used as the shared file system, Hadoop is able to take advantage of knowledge about which node hosts a physical copy of input data, and will attempt to schedule the task that is to read that data, to run on that machine.

The Hadoop Distributed File System (HDFS) Map Reduce environment provides the users with a sophisticated framework to manage the execution of map and reduce tasks across a cluster of machines.

The user is required to tell the framework the following:

•location(s) in the distributed file system of the job input

•location(s) in the distributed file system for the job output

•input format

•output format

•class containing the map function

•Optionally: the class containing the reduce function

•JAR file(s) containing the map and reduce function and any other supports

5. Implementation

Implementations are highlighted below

5.1 Data Access Layer

Data access layer is the one which exposes all the possible operations on the data base to the outside world. It will contain the DAO classes, DAO interfaces, POJOs, and Utils as the internal components. All the other modules of this project will be communicating with the DAO layer in order to access their data.

Account Operations: The following functionalities are provided by the account access operations module to the end users of our project.

- Register a new account
- Login to an existing account
- Logout from the session
- Edit the existing Profile
- Change Password for security issues
- Forgot Password and receive the current password over an email
- Delete an existing Account

Account operations module will be re-using the DAO layer to provide the above mentioned functionalities.

Users Registration: The end users could perform various operations on their profiles. The users can register a new account and thus gaining an access to the portal. And then the users can login to their accounts using the email ID and password registered to access various other partitions in the portal. The users can then choose to update their profile details by providing the new values to the fields provided during the registration phase, or the user can change their password by providing their old password and new one. The users also have the privilege to delete their accounts in case they wish to no longer access our portal. The users' logout from the portal to make sure the session created for them during login is terminated.

Files Access operations on HDFS; the end users can perform various operations on the Hadoop Distributed File System (HDFS) through our platform. The end users can perform various tasks like upload the emails into HDFS, view the uploaded emails or delete unwanted emails. The users will also be given a hack to perform bulk email upload to ensure the users are going to save lot of time in case they have numerous emails to be uploaded to our portal-in-one-go [9].

5.2 Design Consideration:

There are some of the design consideration issues that needs to be addressed or handled before beginning to design a complete solution for the system.

Assumptions and dependencies

The main assumptions and dependencies recognized are as follows:

- JDK has to be installed in the machine where all the three sub component will be executing.
- The application servers like either the JBOSS or the Apache Tomcat will have to be supported by the host machines.
- There shall not be any firewall or other engines that prevents the remote requests from the portal.

5.3 Mode of operation

In User Profile Operation, the end users can perform various operations on their profiles. Firstly, the users can register a new account and thus getting an access to the portal. And then the users can login to their accounts using the registered email ID and password to access various other divisions in the portal. The users can then choose to update their profile by providing the new values to the fields they have provided during the registration phase, or the user can wish to change their password by providing their old password and new password. Couldn't be any permission related issues on any cluster. The host operating system should take of permitting all the requests to the cluster from the interface layer.

In file access operation on HDFS, the end users can perform various operations on the Hadoop Distributed File System (HDFS). The end users can upload the emails into HDFS, view the emails which were already uploaded or delete the emails which they don't wanted to test for the spam contents. The users will also be given a hack to perform bulk email upload to ensure the users are going to save lot of time in case they have numerous emails to be uploaded to our portal in one go. In account operation, User must be creating a new account in our portal to get access to the rest of the modules. The user can also perform various other account operations like retrieving the forgotten password, login, logout, delete profile, change password, and edit profile. In file access operations, the user will be performing various operation on their emails with respect to HDFS.

In obfuscation, the user will be able to perform the obfuscation on their uploaded emails. Then the user will be initiating spam detection algorithm.

In results and report phase, the use can visualize the result of spam detection algorithm.

Data flow diagrams illustrate the way in which the data is sent as an input to the system, how it is processed, where it is stored and where it goes as the output. The below Figure 2 represents the over overall data flow of the whole system

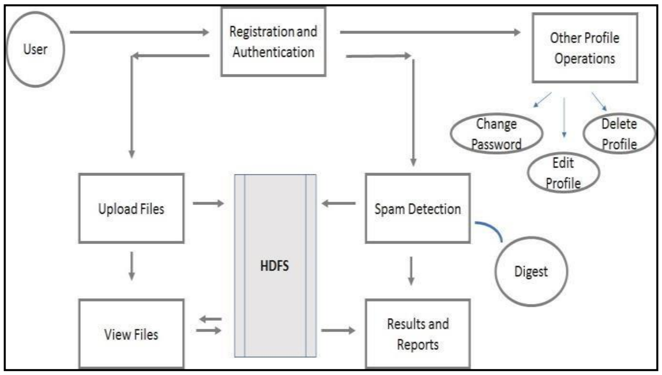


Fig. 2: Data Flow Diagram

User must be creating a new account in our portal to get access to the rest of the modules. The user can also perform various other account operations like retrieving the forgotten password, login, logout, delete profile, change password, and edit profile. Then the user will be performing various operation on their emails with respect to HDFS

5.4 Modules

The different modules use in the system are:

•Account Access Layer

Account access module uses the DAO layer in order to furnish the above mentioned functionalities. The DAO layer is the service layer which provides database CRUD (create, update, read, and delete) services to the other layers.

•Implementation of Spam Detection System

In this phase Spam Detection Algorithm will be implemented to classify the uploaded emails into spam mails and legitimate emails as shown in figure 3,4 and figure 5.

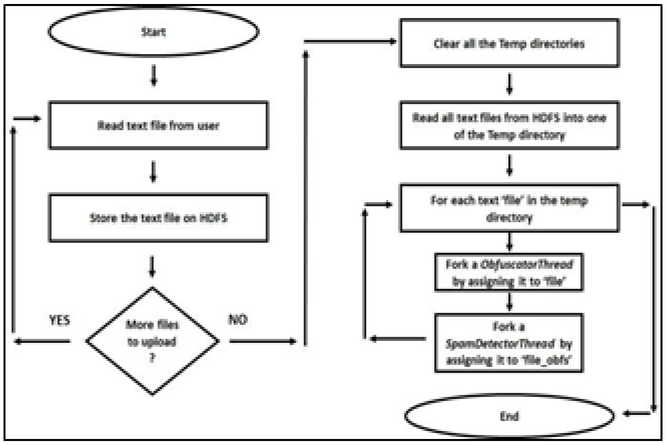

Start
Read text file from user
Store the text file on HDFS
More files to upload ?
YES
NO
Clear all the Temp directories
Read all text files from HDFS into one of the Temp directory
For each text 'file' in the temp directory
Fork a ObfuscatorThread by assigning it to 'file'
Fork a SpamDetectorThread by assigning it to 'file_obfs'
End

Fig 3: Spam Detection Implementation

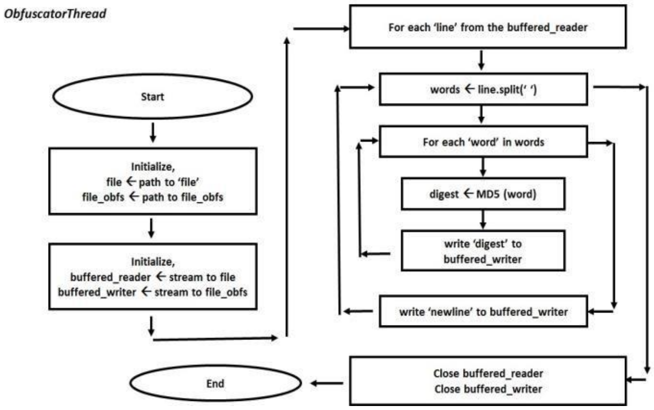

ObfuscatorThread
Start
Initialize,
file ← path to 'file'
file_obfs ← path to file_obfs
Initialize,
buffered_reader ← stream to file
buffered_writer ← stream to file_obfs
End
For each 'line' from the buffered_reader
words ← line.split(' ')
For each 'word' in words
digest ← MD5 (word)
write 'digest' to buffered_writer
write 'newline' to buffered_writer
Close buffered_reader
Close buffered_writer

Fig 4: Obfuscator thread

•Implementation of upload file operations to HDFS

Upload file operations module allows end users to perform different operations on the Hadoop Distributed File System (HDFS). The end users can upload the emails into HDFS, view the emails which were already uploaded or delete the emails which they don't wanted to The users will also be given an hack to perform bulk email upload to ensure the users are going to save lot of time in case they have numerous emails to be uploaded to our portal in one go.

•Implementation of File Access components

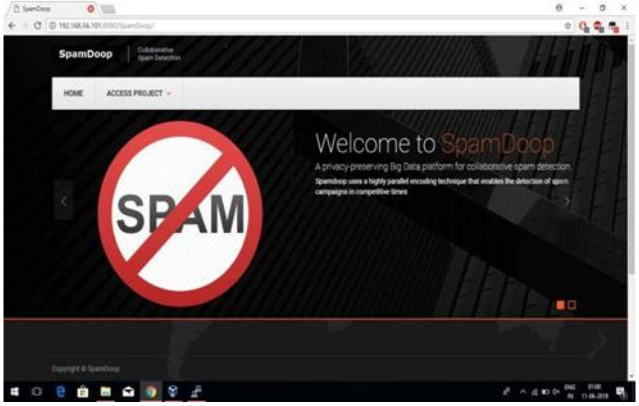
The end users can perform various operations on the Hadoop Distributed File System (HDFS). The end users can upload the emails into HDFS, view the emails which were already uploaded or delete the emails which they don’t wanted to test for the spam contents. The users will also be given a hack to perform bulk email upload to ensure the users are going to save lot of time in case they have numerous emails to be uploaded to our portal in one go.

•Implementation of Results and Reports module

This module renders the result of the spam detection process to the end users. The end users must ensure the algorithm has been executed prior executing this module. This module clearly classifies the uploaded emails into spam emails and not a spam email. All the spam emails will be marked red and all the legitimate emails will be marked green. The users will also be given with an option to download the emails (both spam and legitimate) in case they wanted to analyze the contents of the emails on the go. The users will also be shown with the list of all the words in the emails which caused the email to be marked as spam.

6. Results

Home Page: Here the users can register a new account and get an access to the portal. If the user already has an account, he can login directly by providing email- id and password as shown in figure 6.



Welcome to SpamDoop

A privacy-preserving Big Data platform for collaborative spam detection.

Spamdoop uses a highly parallel encoding technique that enables the detection of spam campaigns in competitive times

Fig 6: Home Screen

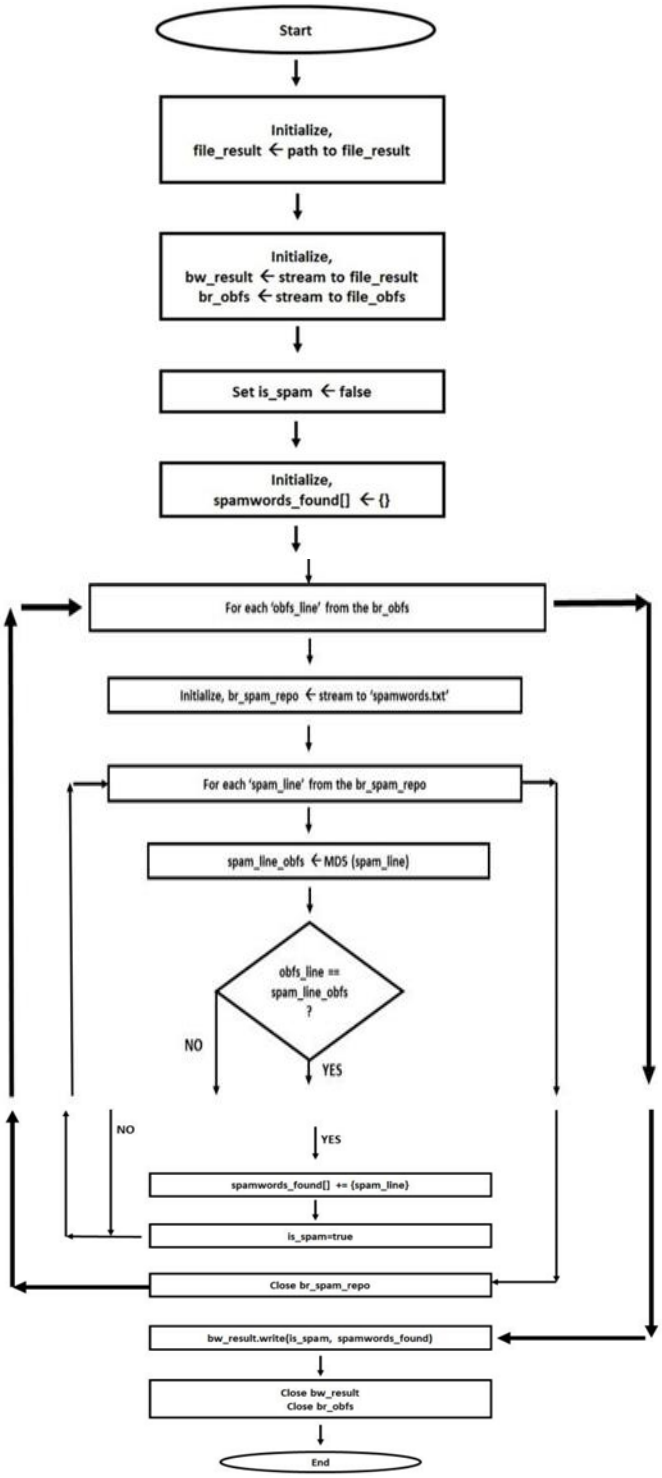
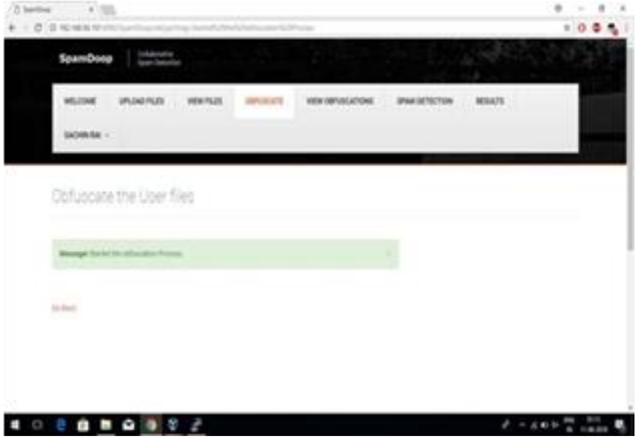

Start
Initialize,
file_result ← path to file_result
Initialize,
bw_result ← stream to file_result
br_obfs ← stream to file_obfs
Set is_spam ← false
Initialize,
spamwords_found[] ← {}
For each 'obfs_line' from the br_obfs
Initialize, br_spam_repo ← stream to 'spamwords.txt'
For each 'spam_line' from the br_spam_repo
spam_line_obfs ← MD5 (spam_line)
obfs_line == spam_line_obfs ?
NO
YES
NO
YES
spamwords_found[] += {spam_line}
is_spam=true
Close br_spam_repo
bw_result.write(is_spam, spamwords_found)
Close bw_result
Close br_obfs
End

Fig 5: Spam Detector thread

Spam Detection Page: User needs to select this tab to start the spam detection process. Once the process is finished, a message is displayed as shown in figure 9.

Obfuscator Page: This tab encodes user's uploaded e-mail content. It displays a message to the user as shown in figure 7. The user can view the encoded email content by clicking on view obfuscator tab where the output of the hashing function is displayed as shown in figure 8



Obfuscate the User files

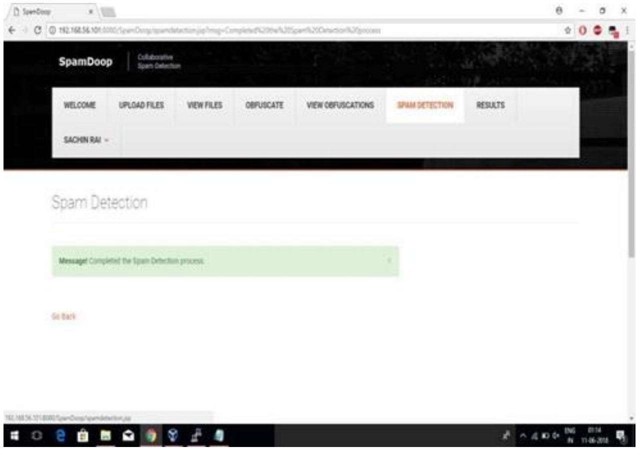
Message! Started the obfuscation Process

Fig 7: Obfuscator Screen

Obfuscated Files

Filename Download

Fig.8 View Obfuscation page

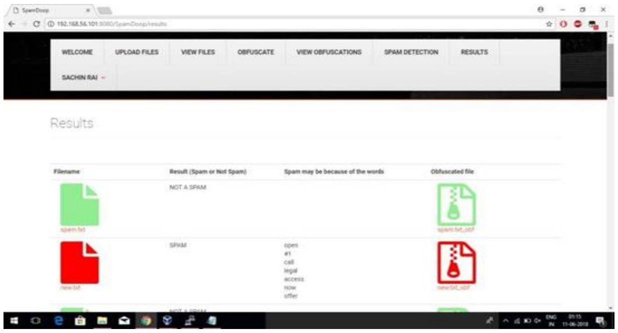


Spam Detection

Message! Completed the Spam Detection process

Fig. 9: Spam detection page

Results Page: In this page, the user can visualize the result of spam detection algorithm. The user must ensure the algorithm has been executed prior executing this module. All the spam emails will be marked red and all the legitimate emails will be marked green as shown in the figure 10. It also displays the list of words which caused a mail to be spam.



Results

Filename	Result (Spam or Not Spam)	Spam may be because of the words	Obfuscated file
spam.txt	NOT A SPAM		spam.txt_obf
new.txt	SPAM	open #1 call legal access now offer	new.txt_obf
	NOT A SPAM		

Fig 10: Results Page

Conclusions

The collaborative spam detection platform being referred to in this paper offers multiple benefits in terms of safeguarding the privacy of all the stakeholders involved and the amount of data being used. The encoding technique employed is effectively scalable on MapReduce Platforms outdoing various distance-preserving hashing techniques.

The techniques used for bucketing simplified the process by offering easy classification and grouping of objects along- with anomaly detection based on histogram efficiently distinguished spam from ham.

Recent tests conducted have shown that the grouping time of digests is reduced by 53% when the work is distributed across four nodes. The computation time is decreased by 57% after using CRC32 on a single node and by 46% on four nodes. Also the processing was of six batches across four nodes was 52% faster. The above mentioned experimental results clearly shows that the Spam detection technique using Big Data are the need of the hour and should be employed to deal with the multitudes of problem related to spams.

Acknowledgments

We consider ourselves proud, to be part of Global Academy of Technology family, the institution which stood by our way in endeavors. We express our deep and sincere thanks to our Principal Dr. N. Rana Pratap Reddy for his support. We are grateful to Dr. Kavitha C, Professor and HOD, department of CSE who is source of inspiration and of invaluable help in channelizing my efforts in right direction.

We wish to thank our guide Mrs. Haseeba Yaseen, Assistant Professor, department of CSE for guiding and correcting various documents with attention and care. She has taken lot of pain to go through the document and make necessary corrections as and when needed.

References

[1] Abdelrahman AlMahmoud, Ernesto Damiani, Hadi Otrok and Yousof Al-Hammadi. "Spamdoop: A privacy-preserving Big Data platform for collaborative spam detection." IEEE Transactions on Big Data, 2017

[2] Cao, Ning, and Yingying Wang. "A Novel Approach to Improve Robustness of Data Mining Models Used in Cyber Security Applications." Computational Science and Engineering (CSE) and Embedded and Ubiquitous Computing (EUC), 2017 IEEE International Conference on. Vol. 2. IEEE, 2017.

[3] Guo, Bo, et al. "Detecting Spammers in E-Commerce Website via Spectrum Features of User Relation Graph." Advanced Cloud and Big Data (CBD), 2017 Fifth International Conference on. IEEE, 2017.

[4] Nagiwale, Amin Nazir, and Manish R. Umale. "Design of self- adjusting algorithm for data-intensive MapReduce applications." Energy Systems and Applications, 2015 International Conference on. IEEE, 2015.

[5] Chen, Long, and Guoyin Wang. "An efficient piecewise hashing method for computer forensics." Knowledge Discovery and Data Mining, 2008. WKDD 2008. First International Workshop on. IEEE, 2008.

[6] Wang, Min, et al. "A General Framework for Linear Distance Preserving Hashing." IEEE Transactions on Image Processing27.2 (2018): 907-922.

[7] Patil, Prajakta, Rashmi Rane, and Madhuri Bhalekar. "Detecting spam and phishing mails using SVM and obfuscation URL detection algorithm." Inventive Systems and Control (ICISC), 2017 International Conference on. IEEE, 2017.

[8] Ankali, Sanjay B., and D. V. Ashoka. "Detection architecture of application layer DDoS attack for internet." International Journal of Advanced Networking and Applications 3.1 (2011): 984.

[9] AlMahmoud, Abdelrahman, et al. "Spamdoop: A privacy-preserving Big Data platform for collaborative spam detection." IEEE Transactions on Big Data (2017).

[10] “What is Big Data?” Available from: https://www.techopedia.com/definition/27745/big- data

[11]Zhong, Zhenyu, Lakshmish Ramaswamy, and Kang Li. "ALPACAS: A large-scale privacy-aware collaborative anti-spam system." INFOCOM 2008. The 27th Conference on Computer Communications. IEEE, 2008.

Haseeba Yaseen working as Assistant Professor in the Department of Computer Science and Engineering, Global Academy of Technology, Bangalore with more than 9 years of teaching experience. She has completed her BE in 2008 and MTech in 2014 from VTU. She published more than 7 research paper in national and international conferences and journals. Her research interest includes machine learning, compiler optimization. She is the member of Computer Society of India.

Adithya DA studying in final year Bachelor of Engineering in Computer Science and Engineering at Global Academy of Technology, Bangalore. His are of interest includes Machine learning, Artificial Intelligence, Operating System and Storage Area Networks.

Ashwini M studying in final year Bachelor of Engineering in Computer Science and Engineering at Global Academy of Technology, Bangalore. His are of interest includes Machine learning, Computer Networks, and Storage Area Networks.

Chandan S. studying in final year Bachelor of Engineering in Computer Science and Engineering at Global Academy of Technology, Bangalore. His are of interest includes Artificial Intelligence, and Distributed systems.

Dhanesh P Kudalkar studying in final year Bachelor of Engineering in Computer Science and Engineering at Global Academy of Technology, Bangalore. His are of interest includes Machine learning, Artificial Intelligence, Operating System and Mobile Networks.