Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 7, Issue. 4, April 2018, pg.53 – 58

SPAM AND EMAIL DETECTION IN BIG DATA PLATFORM USING NAIVES BAYESIAN CLASSIFIER

G.Vijayasekaran[1], S.Rosi[2]
[1]Asst.Professor/Department of CSE
[2]PG Scholar/Department of CSE, rosisenthilvelo1@gmail.com
Sir Issac Newton College of Engineering and Technology, Nagapattinam, TamilNadu, India

Abstract— ***Email spam is operations which are sending the undesirable messages to different email client. E-mail spam is the very recent problem for every individual. The e-mail spam is nothing it's an advertisement of any company/product or any kind of virus which is receiving by the email client mailbox without any notification. To solve this problem the different spam filtering technique is used. The spam filtering techniques are used to protect our mailbox for spam mails. In this project, we are using the Naives Bayesian Classifier with three layer framework that includes obfuscator, classifier and anomaly detector for spam classification for bulk emails. The Naïve Bayesian Classifier is very simple and efficient method for spam classification. Here we are using the real time dataset for classification of spam and non-spam mails. The feature extraction technique is used to extract the feature in terms of digest based on bucket classification. The result is to increase the accuracy of the system. And implement Self Acknowledgeable Intranet Mail System has been designed and implemented to benefit the sender about the status of his mail. Once a mail is sent, the sender can know the receiver activity in the mail system until the mail is viewed. Finally provide the pop up window to identify the mail content at the time of open the spam mails.***

Index Terms — Email Spam, Data mining, Classifier, Anomaly detector, Acknowledgement system

I. INTRODUCTION

In recent years, internet has become an integral part of our life. With increased use of internet, numbers of email users are increasing day by day. It is estimated that 294 billion emails are sent every day. This increasing use of email has created problems caused by unsolicited bulk email messages commonly referred to as Spam. It is assumed that around 90% of emails sent everyday are spam or viruses. Email has now become one of the best ways for advertisements due to which spam emails are generated. Spam emails are the emails that the receiver does not wish to receive. A large number of identical message are sent to several recipients of email. Increasing volume of such spam emails is causing serious problems for internet users, Internet Service Providers, and the whole Internet backbone network. One of the examples of this may be denial of service where spammers send a huge traffic to an email server thus delaying legitimate message to reach intended recipients. Spam emails not only waste resources such as bandwidth, storage and computation power, but may contain fraudulent schemes, bogus offers and scheme. Apart from this, the time and energy of email receivers is wasted who must search for legitimate emails among the spam and take action to dispose the spam. Dealing with spam and classifying it is a very difficult task. Moreover a single model cannot tackle the problem since new spams are constantly evolving and these spams are often actively tailored so that they are not detected adding further impediment to accurate detection.

Spam is abuse of electronic messaging system to send unsolicited bulk messages. Emails are used by number of user to communicate around the world. Along with growth of internet and email, there has been dramatic growth in spam in recent year. Spam can originate from any location across globe, where internet access is available. Spam was created by Hornel in 1937 as the world's first canned meat that didn't need to be refrigated. It was originally named "Hornel Spiced

Ham", but was eventually changed to the catchier name, "SPAM". Usually they come in the form of advertisement, sometimes even containing explicit content or malicious code. Spam has been recognized as problem since1975. According to the statistics from ITU (International Telecommunication Union), 70% to 80% of emails in the internet are spams which have become worldly problem to the information infrastructure. In order to address growing problem there so many anti-spam methods. A spam filter is a program that is used to detect unsolicited and unwanted email and prevent those messages from getting to a user's inbox. Like other types of filtering programs, a spam filter looks for certain criteria on which it bases judgments. For example, the simplest and earliest versions (such as the one available with Microsoft's Hotmail) can be set to watch for particular words in the subject line of messages and to exclude these from the user's inbox. This method is not especially effective; it may omit legitimate messages (called false positives) and passing actual spam messages. More sophisticated programs such as Bayesian filters or other heuristic filters, attempt to identify spam through suspicious word patterns or word frequency. Filter classification strategies can separated into two categories: those based on machine learning (ML) principles and those not based on ML. ML approaches are capable of extracting knowledge from a set of messages supplied, and using the obtained information in the classification of newly received messages. Non-machine learning techniques, such as heuristics, blacklisting and signatures, have been complemented in recent years with new, ML-based technologies. In the last few years, substantial academic research has taken place to evaluate new ML-based approaches to filtering spam. ML filtering techniques can be further categorized into complete and complementary solutions. Complementary solutions are designed to work as a component of a larger filtering system, offering support to the primary filter (whether it be ML or non-ML based). Complete solutions aim to construct a comprehensive knowledge base that allows them to classify all incoming messages independently. The basic machine learning is shown fig 1.

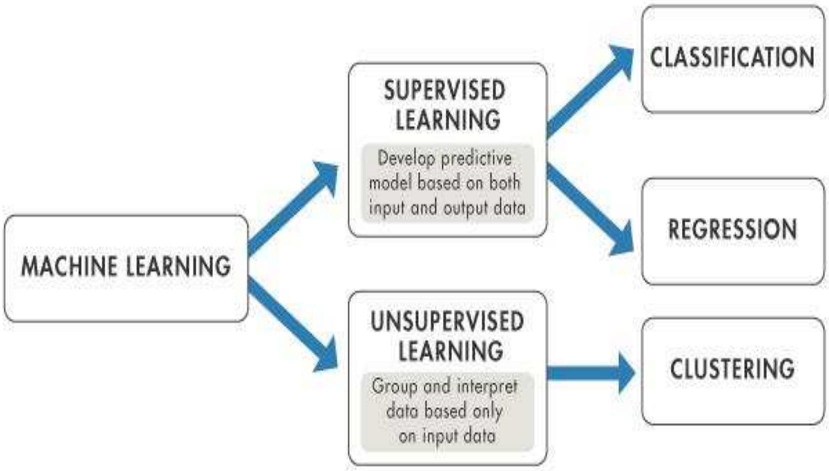
MACHINE LEARNING
SUPERVISED LEARNING
Develop predictive model based on both input and output data
UNSUPERVISED LEARNING
Group and interpret data based only on input data
CLASSIFICATION
REGRESSION
CLUSTERING

Fig 1: Machine learning approach

II. RELATED WORK

M. Crawford T, et.al,… [1] inclined to review a product or service if they had an exceptionally good or bad experience with it. While online reviews can be helpful, blind trust of these reviews is dangerous for both the seller and buyer. Many look at online reviews before placing any online order; however, the reviews may be poisoned or faked for profit or gain, thus any decision based on online reviews must be made cautiously. Furthermore, business owners might give incentives to whoever writes good reviews about their merchandise, or might pay someone to write bad reviews about their competitor's products or services. These fake reviews are considered review spam and can have a great impact in the online marketplace due to the importance of reviews. Review spam can also negatively impact businesses due to loss in consumer trust. The issue is severe enough to have attracted the attention of mainstream media and governments. It is important to mention that while most existing machine learning techniques are not sufficiently effective for review spam detection, they have been found to be more reliable than manual detection. The primary issue is the lack of any distinguishing words (features) that can give a definitive clue for classification of reviews as real or fake. A common approach in text mining is to use a bag of words approach where the presence of individual words, or small groups of words are used as features; however, several studies have found that this approach is not sufficient to train a classifier with adequate performance in review spam detection. Therefore, additional methods of feature engineering (extraction) must be explored in an effort to extract a more informative feature set that will improve review spam detection. In the literature, there are many studies that consider different sets of features for the study of review spam detection utilizing a variety of machine learning techniques.

M. Sheikhalishahi, et.al,…[2] present a framework to effectively and efficiently analyze and cluster large amounts of raw spam emails into spam campaigns, based on a Categorical Clustering Tree (CCTree) algorithm. We introduce a set of 21 categorical features representative of email structure, briefly discussing the discretization procedure for numerical features. The performance of CCTree has been thoroughly evaluated through internal evaluation, to estimate the ability in obtaining homogeneous clusters and external evaluation, for the ability to effectively classify similar elements (emails), when classes are known beforehand. Internal and external evaluation have been performed respectively on a dataset of 10k unclassified spam emails and 276 emails manually divided in classes. Spam emails yearly impose extremely heavy costs in terms of time, storage space and money to both private users and companies. Finding and persecuting spammers and eventual spam emails stakeholders should allow to directly tackle the root of the problem. To facilitate such a difficult analysis, which should be performed on large amounts of unclassified raw emails, in this paper we propose a framework to fast and effectively divide large amount of spam emails into homogeneous campaigns through structural similarity. The framework exploits a set of 21 features representative of the email structure and a novel categorical clustering algorithm named Categorical Clustering Tree (CCTree). The methodology is evaluated and validated through standard tests performed on three dataset accounting to more than 200k real recent spam emails. The CCTree is constructed iteratively through a decision tree like structure, where the leaves of the tree are the desired clusters. The root of the CCTree contains all the elements to be clustered. Each element is described through a set of categorical attributes, such as the Language of a message. Being categorical each attribute may assume a finite set of discrete values, constituting its domain. Shannon Entropy is used both to define a homogeneity measure called node purity, and to select the attribute used to split a node. In particular non-leaf nodes are divided on the base of the attribute yielding the maximum value for Shannon entropy. The separation is represented through a branch for each possible outcome of the specific attribute. Each branch or edge extracted from parent node is labeled with the selected feature which directs data to the child node.

J. Francois, et.al,…[3] controlled by an attacker also called the bot-master. The bot-master sends commands via a C&C (Command and Control) channel. Although first botnets relied on a central architecture with a core network of few interconnected IRC (Internet Relay Chat) servers, current botnets are based on P2P technologies. In a P2P architecture, each bot acts as a client and a server. Hence, for sending a command to certain bots, other bots are involved. Therefore, P2P bots are well interconnected and that is why we argue in this paper that analyzing interactions between hosts is valuable for detecting botnets. Hence, distributed computing might be the only viable solution. The main idea of cloud computing is to provide a simple interface to clients who do not want to manage hardware related details such as the resource allocations. The cloud computing service aims to be very scalable and on demand without long delays: computing power should be available instantaneously. In brief, it can be seen as an abstraction layer taking benefit of recent virtualization outcomes in order to provide a simple way for final users to run tasks requiring intensive computing and storage. MapReduce is a high-level abstraction of parallel computing introduced by Google. Although traditional approaches need to define exactly the way to carry out the data to process, MapReduce programming model focuses on the processing code. The key idea of the method is to shift the network transfer from the data to the code. In brief, the data is distributed a priori using a distributed file system. For achieving a task, the code is then distributed where the needed data is. Therefore, the cloud computing paradigm is well suited for problem dealing with huge volumes of data. MapReduce comes from functional programming concepts with functions (map and reduce) that take other functions as inputs. The map aims to divide input data into multiple inputs for applying a function on each of them (mapper). The reduce function applied by reducers aggregates the individual results from the mappers. These tasks are attributed by a master machine to slave ones. The master is also responsible to detect node or network failures by a ping mechanism in order to reassign tasks to others nodes.

J. Kornblum, et.al,…[4] overwhelmed with data. Modern hard drives contain more information that cannot be manually examined in a reasonable time period creating a need for data reduction techniques. Data reduction techniques aim to draw the examiner's attention to relevant data and minimize extraneous data. For example, a common word processing application is not worth examining, but a known malicious program should be highlighted. This paper describes a method for using a context triggered rolling hash in combination with a traditional hashing algorithm to identify known files that have had data inserted, modified, or deleted. First, we examine how cryptographic hashes are currently used by forensic examiners to identify known files and what weaknesses exist with such hashes. Next, the concept of piecewise hashing is introduced. Finally a rolling hash algorithm that produces a pseudo-random output based only on the current context of an input is described. By using the rolling hash to set the boundaries for the traditional piecewise hashes, we create a Context Triggered Piecewise Hash (CTPH). Such hashes can be used to identify ordered homologous sequences between unknown inputs and known files even if the unknown file is a modified version of the known file. We demonstrate the spamsum algorithm, a CTPH implementation, and briefly analyze its performance using a proof of concept program called ssdeep. To date, forensic examiners have used cryptographic hashing algorithms such as MD5 and SHA-1 for data reduction. These algorithms take an input of arbitrary size and produce a fixed-length value corresponding to that input. Cryptographic hashes have many properties, but forensic examiners take advantage of two of them in particular. First, if even a single bit of the input is changed, the output will be radically different. Second, given an input and its hash, it is computationally infeasible to find another input that produces the same hash. These two properties can be used to identify known files in sets of unknown files. An examiner gathers a set of known files, computes their cryptographic hash values, and stores those values. During future investigations, the examiner can compute the hash values for every file in the investigation and compare those hash values to the known values computed previously. If any of the new hash values match the known values, the investigator has almost certainly found the known files

M. Sirivianos, et.al,…[5] implemented a collaborative platform aiming at suppressing malicious traffic. In addition, it is an open system, meaning that any admin with a social network account and a device can join. As such, it is reasonable to assume that SocialFilter itself will be targeted in order to disrupt its operation. Malicious nodes may issue false reports aiming at reducing the system's ability to detect spam or at disrupting legitimate email traffic. In addition, an adversary may attempt to create multiple SocialFilter identities aiming at increasing its ability to subvert the system using false spammer reports and direct trust updates. Defending against Sybil attacks without a trusted central authority that issues verified identities is hard. Many decentralized systems try to cope with Sybil attacks by binding an identity to an IP address. However, malicious users can readily harvest IP addresses through BGP hijacking or by commanding a large botnet. However, when malicious users create numerous fake OSN accounts, SocialFilter's spammer belief measure can be subverted. Specifically, a malicious user with high reporter trust may create Sybils and assign high direct trust to them. As a result, all the Sybils of the attacker would gain high reporter trust. The Sybils can then submit reports that greatly affect the spammer belief values. Social-network-based Sybil detection takes advantage of the fact that most OSN users have a one-to-one correspondence between their social network identities and their real-world identities. Malicious users can create many identities or connect too many other malicious users, but they can establish only a limited number of trust relationships with real users. Thus, clusters of Sybil attackers are likely to connect to the rest of the social network with a disproportionately small number of edges, forming small quotient cuts. When a node queries the repository for the spammer belief of a host, the repository is interested on the reports for a single host. These reports are sent by multiple nodes, thus for efficiency it is reasonable to index (key) the reports based on the hash of the host's IP.

III. EXISTING METHODOLOGIES

Many spam filtering techniques have been put into business, which include Bayesian spam filtering and collaborative filtering. The concept of Bayesian spam filters is proved to be remarkably efficient. However it is difficult to detect all spam as spammers present many challenges to this content-based filtering technique, like changing vocabulary, introducing the most recognizable terms or adding a relatively high number of random words. The process of spam detection is similar to how memory is developed in our brain, as our spam detecting system can distinguish spam from non-spam emails based on a self-learning algorithm according to the principles of memory forming. The arrival of the new email can be treated as the excitatory input to each existing item, and the scale of the input is analogous to the similarity between the new email and each existing email item in the database. The strength of each item is then accumulated, i.e. the more the item resembles the new email, the stronger the stimulation is, and the faster the corresponding strength grows. When the strength value of an item exceeds the 'remembered threshold', it will be defined as 'spam' by the system. On the contrary, while there is no more newly entering similar emails, the strength value of the corresponding item will decrease. When it drops below an inhibitory threshold, called the 'forgotten threshold', the item is deleted from the database. A sequence of chunk hashes is created to represent the text. Each text file is firstly partitioned into a sequence of chunks according to the algorithm TTTD (Two Thresholds, Two Divisors). The chunks of suspicious emails are then encoded by hash function, which is able to provide privacy for email users.

Whitelist/Blacklist: **-** These approaches simply create a list. A whitelist is a list which includes the email addresses or entire domains which the user knows. An automatic white list management tool is also used by user that helps in automatically adding known addresses to the whitelist. A blacklist is the opposite of whitelist. In this list we add addresses that are harmful for users.

Mail Header Checking: **-** This approach is very known approach. In this we simply consist of set of rules that we match with mail headers. If a mail header matches, then it triggers the server and return mails that have empty "From" field, that have too many digits in address that have different addresses in "To" field from same source etc.

Signatures: **-** This approach is based on generating a signature having unique hash value for each spam message. The filters compare the value of previous stored values with incoming emails values. It is probably impossible for legitimate message having same value with spam message value stored earlier

IV. PROPOSED FRAMEWORK

Email is one of the crucial aspects of web data communication. The increasing use of email has led to a lucrative business opportunity called spamming. A spam is an unwanted data that a web user receives in the form of email or messages. This spamming is actually done by sending unsolicited bulk messages to indiscriminate set of recipients for advertising purpose. These spams messages not only increases the network communication and memory space but can also be used for some attack. This attack can be used to destroy user's information or reveal his identity or data. Spam emails are the emails that the receiver does not wish to receive. A large number of identical message are sent to several recipients of email. Increasing volume of such spam emails is causing serious problems for internet users, Internet Service Providers, and the whole Internet backbone network. This may be denial of service where spammers send a huge traffic to an email server thus delaying legitimate message to reach intended recipients. Spam emails not only waste resources such as bandwidth, storage and computation power, but may contain fraudulent schemes, bogus offers and scheme. Apart from this, the time and energy of email receivers is wasted who must search for legitimate emails among the spam and take action to dispose the spam. Dealing with spam and classifying it is a very difficult task. Moreover a single model cannot tackle the problem since new spams are constantly evolving and these spams are often actively tailored so that they are not detected adding further impediment to accurate detection. Spamdoop is a platform that allows multiple entities to collaborate in early detection of bulk spam campaigns. In this project implement the Obfuscator which is used to encode the email content. And implement parallel classifier and propose anomaly detection approach to analyze the spam and normal mails. Finally provide email acknowledgement system to identify the view status of recipients with pop-up windows for email content.

Secure Hash Algorithm:

In cryptography, SHA-1 (Secure Hash Algorithm 1) is a cryptographic hash function designed by the United States National Security Agency and is a U.S. Federal Information Processing Standard published by the United States NIST.[3] SHA-1 produces a 160-bit (20-byte) hash value known as a message digest. A SHA-1 hash value is typically rendered as a hexadecimal number, 40 digits long. Hashing function is one of the most commonly used encryption methods. A hash is a special mathematical function that performs one-way encryption. The procedure is used to send a non-secret but signed message from sender to receiver. In such a case following steps are followed: Sender feeds a plaintext message into SHA-l algorithm and obtains a 160-bit SHA-l hash. Sender then signs the hash with his RSA private key and sends both the plaintext message and the signed hash to the receiver. After receiving the message, the receiver computes the SHA-l hash himself and also applies the sender's public key to the signed hash to obtain the original hash H.

Naives bayes Classifier:

Naive Bayes classifiers are a popular statistical technique of e-mail filtering. They typically use bag of words features to identify spam e-mail, an approach commonly used in text classification. Naive Bayes classifiers work by correlating the use of tokens (typically words, or sometimes other things), with spam and non-spam e-mails and then using Bayes' theorem to calculate a probability that an email is or is not spam. Naive Bayes spam filtering is a baseline technique for dealing with spam that can tailor itself to the email needs of individual users and give low false positive spam detection rates that are generally acceptable to users. It is one of the oldest ways of doing spam filtering. In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers. In the statistics and computer science literature, Naive Bayes models are known under a variety of names, including simple Bayes and independence Bayes. All these names reference the use of Bayes' theorem in the classifier's decision rule, but naive Bayes is not (necessarily) a Bayesian method. Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods. Despite their naive design and apparently oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations. An analysis of the Bayesian classification problem showed that there are sound theoretical reasons for the apparently implausible efficacy of naive Bayes classifiers. After training, the word probabilities (also known as likelihood functions) are used to compute the probability that an email with a particular set of words in it belongs to either category. Each word in the email contributes to the email's spam probability, or only the most interesting words. This contribution is called the posterior probability and is computed using Bayes' theorem. Then, the email's spam probability is computed over all words in the email, and if the total exceeds a certain threshold, the filter will mark the email as a spam. As in any other spam filtering technique, email marked as spam can then be automatically moved to a "Junk" email folder, or even deleted outright. Some software implements quarantine mechanisms that define a time frame during which the user is allowed to review the software's decision. The initial training can usually be refined when wrong judgments from the software are identified (false positives or false negatives). That allows the software to dynamically adapt to the ever evolving nature of spam. The proposed framework is shown in fig 2.

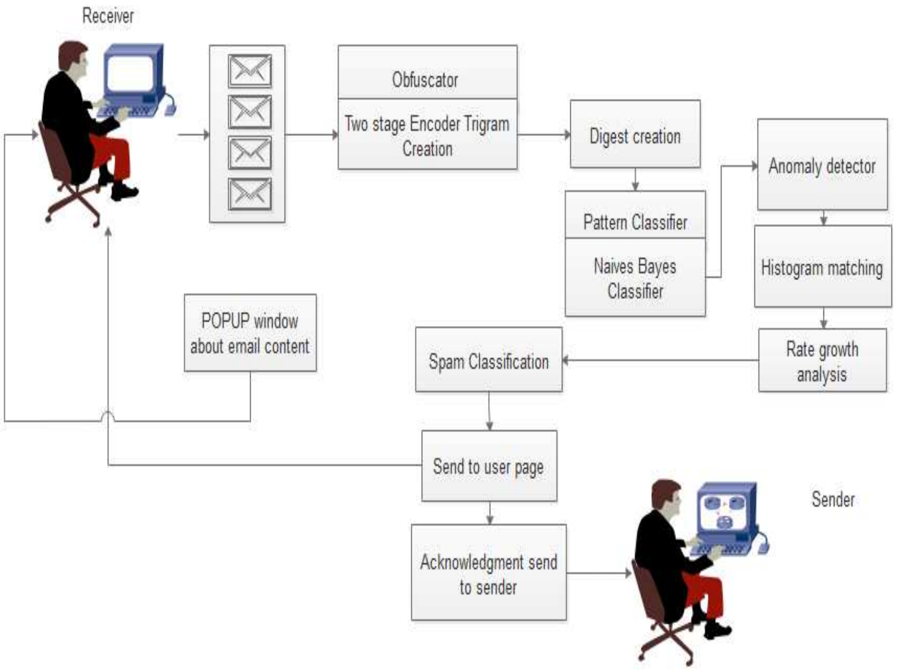

Fig 2: Proposed Work

Spam is abuse of electronic messaging system to send unsolicited bulk messages. Today large volumes of spam emails are causing serious problem for the users, and internet services. Such as, It degrades user search experience, It assists propagation of virus in network, It increase load on the network traffic, It wastes the resources such as bandwidth, storage, and computation power, It also wastes the user time and energy. Spam filter it minimize the amount of junk email. Email filtering is the processing of emails to organize it according to specified criteria. Common use of mail filters are Organize incoming mail, Removal of spam emails, Removal of computer virus. Implement Naives Bayesian filter learn from both good and spam emails, result in an adapting and efficient anti-spam approach which includes The Obfuscator, Pattern Classifier, and enhanced anomaly detector with acknowledgement system. The pop window based alert system to identify email content details

V. Conclusion

E-mail is an efficient, quick and low-cost communication approach. E-mail Spam is non-requested data sent to the E-mail boxes. Spam could be a huge drawback each for users and for ISPs. According to investigation nowadays user receives a lot of spam emails then non spam emails. To avoid spam/irrelevant mails we'd like effective spam filtering strategies. Spam mails area unit used for spreading virus or malicious code, for fraud in banking, for phishing, and for advertising. Spam messages are nuisance and huge problem to most users since they clutter their mailboxes and waste their time to delete all the junk mails before reading the legitimate ones. They also cost user money with dial up connections; waste network bandwidth and disk space. Bayesian classifier is one of the most important and widely used classifier and also it’s the simplest classification method due to its manipulating capabilities of tokens and associated probabilities according to the users’ classification decision and empirical performance. In this project, we implemented the system to analyze each and every mail. And also provide privacy based detection system to encode the emails using Digest based system. Enhance the anomaly detector, to predict emails with pop up window with email tracking system. In the future work we have a plan to implement other algorithm to our classification method to achieve better performance.

References

[1] M. Crawford T. Khoshgoftaar J. Prusa. “survey of review spam detection using machine learning techniques”. Journal Of Big Data, Vol. 2:pages 23, 2015.

[2] M. Sheikhalishahi, A. Saracino, M. Mejri, N. Tawbi, and F. Martinelli. Fast and effective clustering of spam emails based on structural similarity. In International Symposium on Foundations and Practice of Security, pages 195–211. Springer, 2015.

[3] J. Francois, S. Wang, W. Bronzi, R. State, and T. Engel. Botcloud: Detecting botnets using mapreduce. In IEEE International Workshop on Information Forensics and Security (WIFS), pages 1–6. IEEE, 2011.

[4] J. Kornblum. Identifying almost identical files using context triggered piecewise hashing. Digital investigation, vol. 3 Pages 91–97, 2006.

[5] M. Sirivianos, K. Kim, and X. Yang. Socialfilter: Introducing social trust to collaborative spam mitigation. In INFOCOM, pages 2300–2308. IEEE, 2011.

[6] G. Caruana, M. Li, and M. Qi. A mapreduce based parallel svm for large scale spam filtering. In International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), volume 4, pages 2659–2662. IEEE, 2011.

[7] C. Tseng, P. Sung, and M. Chen. Cosdes: A collaborative spam detection system with a novel e-mail abstraction scheme. IEEE Transactions on Knowledge and Data Engineering, vol. 23: pages 669–682, 2011.

[8] S. Dinh, T. Azeb, F. Fortin, D. Mouheb, and M. Debbabi. Spam campaign detection, analysis, and investigation. Digital Investigation, vol. 12: pages 12–21, 2015.

[9] R. Fontugne, J. Mazel, and K. Fukuda. Hashdoop: A mapreduce framework for network anomaly detection. In Computer Communications Workshops (INFOCOM WKSHPS), pages 494–499. IEEE, 2014.

[10] M. Goldstein and A. Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. KI-2012: Poster and Demo Track, pages 59–63, 2012.