Social Media Spam Detection Using Fuzzy String Matching

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

in

Information Security

Submitted By

Alok Kumar

(Roll No. : 801333001)

Under the supervision of:

Dr. Maninder Singh

Associate Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

June 2015

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*Social Media Spam Detection Using Fuzzy String Matching*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Information Security* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Dr. Maninder Singh and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Alok Kumar

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Dr. Maninder Singh
(Associate Professor, CSED)

Countersigned by

(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. S. Bhatia)
Dean (Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to Dr. Maninder Singh, my mentor and thesis supervisor for his constant support and motivation. He had been instrumental in guiding me throughout the thesis with his valuable insights, constructive criticisms and interminable encouragement.

I would like to thank all the faculty members and staff of Computer Science and Engineering Department who were always there at the need of hour and provided all the help and facilities, which I required, for the completion of this work.

I offer my deepest gratitude to my family for their support and affection and believing in me always. I also want to thank my colleagues, who have given me moral support and their relentless advice throughout the completion of this work.

Alok Kumar

ME (Information Security)

801333001

ABSTRACT

One of the most popular internet activities around the world is visiting online social networks. The number of users and time spend by users on these networks is increasing at a high rate. Moreover users tend to rely on the trustworthiness of the data present on the networks. But in wrong hands this trustworthiness can easily be exploited and be used for spreading spam. Users are harassed by spam messages which waste time and make users to click on malicious links. Spams effect many type of electronic communication including instant messaging, email and social networks but due to open nature, reliance on users for data and because of huge user base social networks are worst hit because of spams. To detect spams from the social networks it is desirable to find new unsupervised techniques which can save the training cost of supervised techniques.

In this thesis we present an unsupervised, distributed and decentralized technique to detect and remove spams from the social networks. We present a new technique which uses fuzzy based method to detect spams which is different from existing techniques and which can catch spam even from single message stream. We have parallelized our work with multi-core and multi-thread to give improved results. To handle huge data of networks we have implemented our technique on MapReduce platform which gave very promising results.

Table Of Contents

List of Figures

Page

List of Tables

Page

Chapter 1

Introduction

1.1 Information Security

Information security is branch of security which defends the information in various aspects like unauthorized access, use, disruption, inspection, modification, disclosure, perusal, recording or destruction. Information security is securing the computing devices either it is a desktop, router, data center or server etc.. The data must be safe either in a natural disaster or in any security breach or in case of system failure. As almost every information is kept on computers in today's world, there is a need of an IT security specialist to deal with such issues.The definition of information security is "Preservation of confidentiality, integrity and availability of information in addition with authenticity, accountability, non-repudiation and reliability". The basic principles of information security are:

- Confidentiality
- Integrity
- Availability

Confidentiality It means maintaining the privacy. Confidentiality is to giving access only to the authorized persons. No unauthorized person should get access to information.

Integrity Integrity means maintaining and preserving the consistency and accuracy of data. Data should not be affected/altered/lost in the way or transmission.

Availability The information should be available when needed. No unauthorized person should be able to un-avail the information to other users. Attacker tries to do this by DoS and DDos attacks which stops the availability of information to the users.
The above mentioned are the main and basic principles of information security.

1.2 Social Spams

Social spam is any unwanted content which appear on social networks or on any website with user-generated content. It includes bulk messages, insults, malicious links, fake friends, hate speech etc. Social spamming is included as a part of phishing attack [30], commercial spam messages [16, 72], to disseminate malware [15] and to promote websites. In 2008 about 80% of the users of the social networks received unwanted friend requests, messages or posting on their accounts (Source: Harris Interactive, June 2008). Social spam contains contextual information which was not present in email based spam and thus social spams have much greater impact [16, 22, 30] and thus no effective technique is available for zero day social spam attacks. Moreover an experiment showed that around 41% of Facebook users acknowledged friend request form random person [15] and because of which the social spams have changed their attack grounds from email to social networks.

1.2.1 Spam breakdown

Chris et al. [27] in his work found that about 8% of the total URLs in the post are phishing pages, malware and various scams which are listed on popular blacklists. After finding all the spams manual classification of the spams is done to find the spam category. Roughly about 50% of the spam were not categorized as they were random but remaining 50% were categorized and is shown in Table 1.1, maximum spam are present in content of Free music, games, books and downloads which have about 30% of spam content and so on.

Category	Fraction of spam
Free music, games, books, downloads	29.82%
Jewelers, electronics, vehicles	22.22%
Contest, gambling, prizes	15.72%
Finance, loans, realty	13.07%
Increase Twitter following	11.18%
Diet	3.10%
Adult	2.83%
Charity, donation scams	1.65%
Pharmaceutical	0.27%
Antivirus	0.14%

Table 1.1: Percentage of Spam Content [27]

1.3 Features which attract spammers

Chris et al. [27] has also discussed many features of twitter which motivate spammers like

Call Outs- Mentions can be used by spammers to create own messages in order to increase the likelihood of victim following a spam link. Moreover mentions can be used to send messages to users which do not follow spammer

Example: win an iphone @victim! http://spammer.com

Retweet- It can be explained as reposition of someone else tweet and is used to quickly share the tweet with all the followers.

Example: RT@spammer:win an iphone

Tweet Hijacking- It include spammers hijacking tweets posted by other users and retweet them, moreover no restriction is available on twitter on who can retweet a message which allows spammers to alter the tweets of the users and embed spam URLs and retweet them.

Example: http://spammer.com RT@spammer We will spam the network

Trend setting- Hashtags are used to constrain the search results of the content and if enough users tweet common hashtag, it become a trend topic.

Example: win iphone! http://spammer.com #ip6

Trend hijacking- In this case the spammers can append the current trend topic in their own spam and when anyone searches for the topic will encounter the spam message.

Example: Win iphone #ip6 ref:http://spammer.com

The above features can easily be used by spammers to spread the spam deep into the network and attack is more effective and distributed as compared to email spam which makes detection and removal of the spam content from the social network a big task.

1.4 Categories of spam bots

Gianluca et al. [56] in his work has categorized the spam bots based on the different level of activities and strategies for delivery of spam, they are-

Displayer: These are the bot which show spam content on their profile pages but they do not post any spam messages, so to view the spam content the victim has to manually go the spam page of the bot and thus are least effective in term of people reached.

Bragger: These post messages on their own feeds in case of Facebook it would be status update and in twitter it is tweets and thus the spam message is distributed to all victims but these messages will not be seen by the victim's friend and thus spam campaign is limited to the victims which are directly connected to the spam bot.

Poster: In this spammer send messages directly to the each victim in case of the Facebook the message is displayed on the wall of the victim and thus victim's friends can also see the post which increase the effectiveness of the spam and it reaches to greater number of users.

Whisperer: In this case spammers send private messages to the victim which can be seen by him only. These types of the bots are very common in the Twitter.

1.5 How users are exposed to the spam

Kurt et al. [59] in his work has discussed about ways by which normal users are exposed to the spam content which are

User timeline: Victim can receive the spam content on their timeline which have all the broadcasts from each of the victim's friend, so spammer can inject content into this stream.

Trending topics: Except of creating relationships with the user or targeting each user, spammer can send messages which contain popular keywords form trending topics and when users explore these topics, they will encounter the spam with legitimate content.

Search: Twitter give a facility to users to search public tweets and spammers can exploit this facility by embedding popular search terms into tweets.

Direct Messages: This include private tweet which is send by spammers to users which they follow, as user do not need any reciprocate relationship to receive spam content.

1.6 Detecting social spammers

Social spam detection techniques are evolving but still are facing open issues. Till date these issues do not have optimal answers. The need of the hour is deep analysis and further research to address problem of Social spamming confidently.

1.6.1 Spam profile detection features

Gianluca et al. [56] in his work also found various features which can be used to differentiate spammers and legitimate users these are-

FF ratio: It is the ratio of the friend request sent with the number of friends it has and author expects the ratio to be high in case of spammers and low in case of regular users.

URL ratio: It is presence of URL in the messages, as spammers are likely to include URL to send victim to spam pages.

Messages Similarity: It checks for the similarity between the messages by a user, as bot tend to send very similar messages including both content and message size.

Message sent: The number of message sent can also be a important feature to find spammers. Kyumin et al.[37] in his work has shown that spammers post fewer than 120 messages.

Friend Number: This feature is based on the fact that legitimate users have large number of friends and spammers generally have few friends. Kyumin et al. [37]

in his work has shown that 86% of spammers have fewer than 50 followings and fewer than 10 followers.

1.6.2 Various evasion techniques used by spammers

Evasion techniques are used by spammers to evade existing detection methods used and these are

Link based evasion techniques- In Link based detection the relationship between the users is checked and spammers are differentiated from the honest users. Link based evasion techniques include change in link structure of the network or manipulation of the link structure with the intent of improving the rating of one or more than one users and thus creating relations more like normal users.

Profile based evasion technique- For discovering spam accounts most common intuition is basic profile information such as number of followers and tweets as these reflect the reputation of a user. These features can be easily manipulated by the spammer to look more legitimate. To get more followers spammers can either buy the followers online or can exchange followers with other users. To look more legitimate spammers can send large number of tweets and occasionally send spam messages.

Content based evasion technique- Content of the messages is also a common indicator for detecting spam accounts as majority of spammer includes URLs which can redirect victims to spam pages and thus many techniques [38, 57, 64] are designed which check the percentage of URL posts. Moreover the spammers can be caught based on the tweet similarity [38, 57] and duplicate tweet count [64] for detection of the spams. To evade these features spammers use tactics like mixing normal tweets in spam tweets to dilute the spam tweets and posting heterogeneous tweets which have different terms with same semantic meaning.

URL based evasion techniques- Various URL shorting techniques are also used to evade detection where spammers can use freely available URL shortener like TinyURL .com

1.6.3 Social spam Detection techniques

To detect social spams various detection techniques are available. These techniques can be broadly classified into link based and content based detection techniques. Link based detection technique detect spam by using the nodes which represent the users and the graph which represent the network and the relation between them is shown by edges. This technique mainly uses Human established trust relationships and tries to find that whether the relationship between two user is honest or is made maliciously and thus differentiate spammers from honest users. On the other hand content based detection technique uses the content present on network for detection, it include messages, profile data, URLs which are used to differentiate normal and spam content.

Chapter 2

Literature Survey

In past few years there has been rapid rise in Web-based system which incorporates social features including Web based social network like Facebook, MySpace and online social media sites like YouTube, Flickr and large-scale information sharing communities like Yahoo! Answer. All of these attract a huge amount of research and media interest [11, 20, 43]. Main feature of these systems is reliance on users for content which have several positive effects like large scale growth in size and content of the community and new social based information retrieval algorithms. Twitter was the fastest growing website in February 2009 with growth rate of 1382% [44]. Another survey also revealed that users of twitter send about 500 million tweets in each day and about 460000 new accounts were opened every day [4]. Facebook alone has more than 1.35 billion monthly active users as of September 30, 2014 [10]. LinkedIn have profiles of more than 184 million users [9]. But because of wide spread interest and huge growth of these social systems and this openness and reliance on users have made these systems prime targets of social spammers. Social spam is low-quality information which users do not ask and thus misleads users from the information they are looking for. Spammers have many goals which include phishing, advertising and malware distribution. Goals are very much similar to traditional spam in email but methods are far more sophisticated and evolved making the effect of spamming more severe in social networks. It was found that about 1.5 million fake or compromised Facebook accounts were on sale in February 2010 [1]. These compromised accounts are used for various purposes [1, 2, 3] including sending abusive messages and posts [25, 60] and acquiring users

private profile data and contact lists [14, 24].

More over successful defending of these spammers is important to improve the experience of the users in the community, to decrease the system load in dealing with unwanted and dangerous content but problem is little knowledge about social spammers, their strategies and tactics and because of which efficient techniques are not available which can eliminate spammers and spam content from the network.

2.1 Link based detection techniques

Link based detection technique mainly focuses on social network and the relation between various nodes which represent users. These techniques try to find spammers using the relation between users and checks whether the relation is honest or dishonest.

2.1.1 Haifeng et al.(SybilGuard)

In his work [70] he proposes SybilGuard technique, that can limit the influence of the sybil attack. Sybil attack is an attack where a spammers subverts an honest node. The technique is based on the human established trust relationship which is represented as an edge between two identities and malicious users can have many identities but have only few trust relationship.

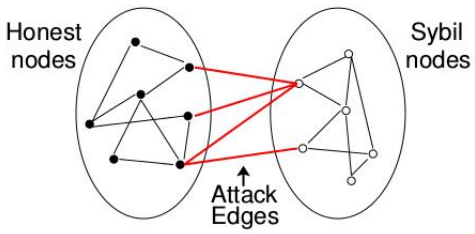

Figure 2.1: Social Networks [70]

Architecture= The technique is based on decentralized approach to defend against sybil attacks and uses the human established trust relationship for detection of spam, where all the users and sybil node present form a social network as shown in Figure 2.1. The edges between the honest and sybil regions are the attack edges. The SybilGuard makes sure that the attack edges are limited to

trust relation pair between honest user and malicious user(Which are assumed to be less) and is independent of number of sybil nodes. It give an option to honest user to decide whether to accept another node , "Accepting" means that a node is willing to receive and give service to other node. SybilGuard also observes that graph will have small quotient cut (a small number of edges whose removal disconnects the graph) if the sybil nodes are created on large scale.

2.1.2 Haifeng et al.(SybilLimit)

SybilLimit [69] is a technique which is an advanced version of SybilGuard which give better results then SybilGuard [70]. It is also based on human trust relationship and is able to limit the number of nodes accepted and give a near optimal result.

Architecture= SybilLimit have same system model like of SybilGuard, but SybilLimit tries to decrease the number of sybil nodes accepted to O(log n). In this a secure random route protocol is run to find the honest groups and the sybil groups, using the previous result with the verification protocol mechanism a honest node can know whether other node is trust worthy or not and on that bases it can either accept or reject association with other node.

2.1.3 Danezis et al.(SybilInfer)

In his work [21] he gave an approach SybilInfer that can label nodes in social network as honest users and sybil nodes. The model is based on probabilistic model which can be used to decide degree of certainty about a node to be either honest or sybil.

Architecture= The technique takes the social network graph and a known honest node as input. It is based on simple assumption that social networks are fast mixing [34]. A set of traces is formed by performing random walks in social graph G, then a probabilistic model defines the likelihood of a trace which take assumption that social networks are fast mixing and transitions in sybil nodes is slow, then

Bayes theorem is used to calculate whether a set of nodes are honest or not.

2.1.4 Jonghyuk et al.

In this work [54] author has proposed a spam filtering system that uses relation features between the users like connectivity and distance between them for spam detection. As these features are difficult to manipulate and these can be used to discriminate spammers from honest users.

Architecture= In this approach author constructs a directed graph which is based on following and followed relations in network. Two features (distance and connectivity) are measured, where distance is shortest path from sender to receiver and connectivity is measured using random walk and min-cut. The connectivity represent the strength of the relation and can be measured by presence of attack edges [27,32] . In the technique almost all the messages which have a distance of more than four is treated as spam because distance more than four have very less or no relationship with receiver and this is mainly in case of spammers. Moreover connectivity is used to find the strength of the relationship, and spammers have a weaker connectivity.

2.1.5 Qiang et al.(SybilRank)

SybilRank [17] is a protocol that is based on social graph properties [21, 61, 62, 63] and it ranks the users according to their likelihood of being fake and can be used on graph of large sizes having millions of nodes.

Architecture= The technique is based on the theory that the random walk from a honest node have a higher degree normalized landing probability to land over a honest node than over a sybil node and the reason for this can be limited attack edges which create a narrow passage from honest region to sybil region, So a shortened random walk from a honest node tends to end in honest region. This method is used to rank nodes and nodes with lesser rank are fake users Figure 2.2 tell more about technique which have 3 stages, in stage 1 trust flows from honest nodes over the entire network, in stage 2 ranking of nodes is done based

on degree-normalization trust, in stage 3 fake nodes are defined and thus enables OSNs to manually inspect those nodes.

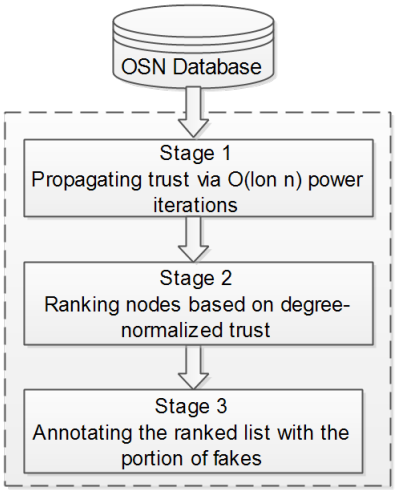

Figure 2.2: SybilRank(Architecture) [17]

2.1.6 Yinglian et al.

To detect legitimate users early in OSN author has designed a framework named souche [68], increasing both usability and security. For detection it uses the social connection between users which are established over time where a legitimate user help to identify other legitimate users through a voucing process which closely monitors vouching on social community structure and thus a large subset of legitimate users can be recognized as early as they start malicious attack and thus limit their growth at that time only.

Architecture= The system include two components, first bootstraps the system with help of constructing social graph and use graph properties to identify a group of honest users. Second component decides when and where vouching is allowed, for this community structures are build in form of vouching trees by which the vouching population can be controlled and monitored. In the bootstrapping process firstly some vouters are identified either manually or with automatic user reputation generation system. Then social graph is created where graph represent the connection between the users on the bases of communication and activities which a user do in social network. To limit and stop malicious accounts from connecting to large number of legitimate users, strong mutual users connection is

used. The vouching algorithm is applied over initial vouchers and the graph is dynamically monitored and growth of the vouched populations is controlled over time which helps in limiting any presence of malicious or compromised vochers.

2.1.7 Junxian et al.(SocialWatch)

Author has given a system SocialWatch [29] which uses a set of social graph properties which can show the overall social activity and connection pattern of the users online including PageRank, degree and social affinity features to detect attacker controlled accounts in social graph same as techniques [53, 46, 19]. Author has used these features as these are hard to mimic and are robust to attackers counter strategies.

Architecture= The architecture of the system is divided into two types depending upon the detection they do, one is for detection of Attacker created accounts and other is for Hijacked accounts detection. For detection of the attacker created accounts two graph properties are used node degree and PageRank, where node degree is the aggressiveness of the account and PageRank calculates the weights of nodes in overall graph. Defense using node degree includes the response rate of an account which is ratio of number of replies form all of the messages. In case of PageRank each page is given a initial reputation score and each node propagate its reputation to neighbors. For detection of hijacked accounts communication patterns are used and for this two social affinity features are used Recipient connectivity which checks the connection between the sender and receiver and Social distance which is distance between the sender and receiver for detections.

2.1.8 Enhua et al.(UNIK)

Author in the paper [58] has firstly discussed limitation in existing unsupervised detection scheme and main reason is reliance on spamming patterns which changes constantly for avoidance of detection. Author have designed a Sybil defense based spam detection technique SD2, to further improve the results, author has also defined UNIK framework which overcame the limitation of [25, 67] which can re-

move the spammers from the network. UNIK work on the principle that spammers have to change the pattern to evade detection but on the other hand non-spammers don't change their pattern and generally have a non-volatile pattern.

Architecture=Firstly SD2, in this a user graph is created by social graph which represent relationship between active non-spammers and user-link graph which represent the spam link sharing activity of spammers. Using the thought that spammers and non-spammers form different communities SD2 applies community detection based Sybil defense algorithm on the user graph and gives better performance than existing schemes, but the drawback of Sybil defense to give poor results in high spam density also persists in SD2.

To solve the problem and improve the detection performance in increased spam attack author has given a new framework UNIK (UN-supervised socIal networK) Figure 2.3 which capture the properties of non-spammers instead of spam users directly and then it forms clusters of suspicious spammers based on the URLs. UNIK first create a user-link graph which include all users who share a URL link and this graph will contain spammers and non-spammers. Then a social graph is created using the mutual social connection which persists between the users and using this graph non spammers are identified and the URL posted by these identified non spammers are added in the white list. Edges of the non-spam URL edges are trimmed in user link graph using this white list and thus a large portion of non-spammers are isolated in user-link graph and on bases of the node degree UNIK differentiate spammers from non-spammers which are trimmed from the user-link graph detecting majority of the spammers.

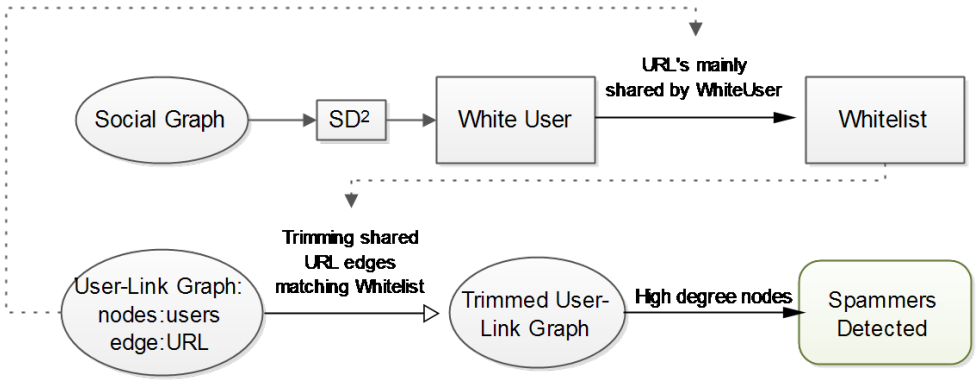

Figure 2.3: UNIK(Architecture) [58]

2.1.9 Meng et al.(CatchSync)

In this author has given a framework CatchSync [31] which works on two principles first is synchronized behavior which tells that almost all the suspicious node have same pattern because some tasks are performed by them together and next is rare behavior which tells that the suspicious nodes have very different connectivity patterns from the majority. CatchSync is designed to work on large datasets and it is parameter free as it work on synchronicity-normality plots results and it is side-information oblivious which means it can operate only on topology and don't need any labeled data etc. It works orthogonal to techniques [18, 12, 49].
Architecture= The approach firstly give feature space for target nodes, then measure node behavior using synchronicity and normality, then general theorem is provided for synchronicity vs normality plot and find the outliers on the plot which are suspicious nodes which have a synchronized behavior on the graph. Author have chosen degree value and HIT score as features because they are fast to compute and can easily plotted. Author has proved that these features work well to detect suspicious nodes. To detect the behavior pattern of a node author has proposed Synchronicity which is average closeness between each pair of a user's target and Normality which is average closeness between each pair of usersٔ target and other nodes. Both have values between 0 and 1. It uses the principle that for a suspicious node synchronicity is high and normality is low.

2.2 Content Based Detection Techniques

These techniques use the content of social network to detect the spammers. Content can include profile content which include all the Profile data like name,about me,gender,age,hobbies. Message data which include all the messages which are send over the network. URL content which include all the URL that are send with the messages.
As discussed above the content based detection is sub divided in three types dis-

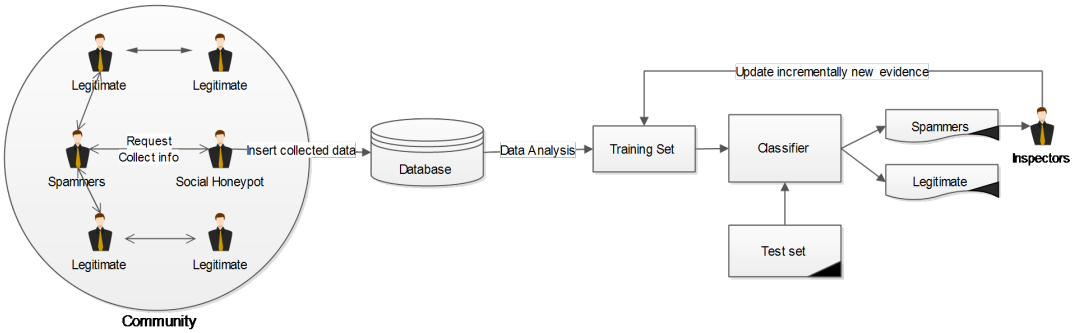

Figure 2.4: Social Honeypots(Architecture) [38]

cussed below:

2.2.1 Profile Based Detection

Kyumin et al.(Social Honeypots)

Author has a proposed a honeypot based approach [38] for uncovering social spammers in online social networks. Use of Honepot for detection was inspired from techniques generating intrusion detection signatures [34], for characterizing malicious hacker activity [50], observing email address harvesters [55] .Two main components of the approach are Deployment of social honeypots for catching spam profiles in social networks and Statistical analysis of the profile for creation of spam classifiers to actively catch new and existing spammers.

Architecture= The architecture of the approach include introduction of social honeypots in network for targeting community based online activities. Author deploys a social honeypot consisting of a legitimate profile and a bot is associated with it to detect spamming activity. If bot detect any spam activity it collects all the evidence of spam candidate by crawling the profile data. These evidences are later used to extract features from the profiles and data set is created of spam and legitimate profile to be fed for spam classifiers which can be used to detect spam on large scale. Figure 2.4 explain the framework of the approach.

Zifei et al.(CloneSpotter)

In the paper [52] author has described two improvements in Snowball sampling [26] and Iteration attacks which can be used to do cloning attack over OSNs and after that author has designed a new framework to detect the cloning attack named Clonespotter which can be deployed in OSN servers which works on IP records and find evidence of locations to judge whether accounts are manipulated by real users or malicious users. Cloning attack is attack in which attacker create many fake accounts of real accounts by copying their profiles and sending friend requests to friends of victim. Snowball sampling technique makes clone to get more accepted friend request and to be more embedded in the network and iteration attack helps attacker by injecting multiple fake identities of a single user in the community which in result increases the overall influence of the attacker.
Architecture= Author has designed Clonespotter and divided the overall architecture into two major categories Content free Detection(make judgments according to physical information neglecting user generated content) and Content related detection approach(uses the user content to make judgments). In the content free detection the there are various steps- firstly the server records the login IP addresses of a user and each time user login form different IP the sequence is updated. Next is looking on the friend requests send from one user (A) to another (B), the friend list of B is checked for finding any friend with same name as of A if no such name than stop, else for each same name found compare its profile with A's profile and calculate the similarity based on the profile content, if the similarity exceed some threshold, it means two profiles are of same person. But the same accounts can be either case of cloned profiles of can be case of where normal users have two accounts. To know whether A is clone or not compare IP sequence of both accounts, if both have same IP sequence it is not a clone else it is a clone.

Marcel et al.(Web Search)

Author has proposed a system [23] which uses web search to find whether an account is fake or real. The system uses the web search to find the online presence

of the user and accounts with less web presence are likely to be fake. The system has ability to operate even before a single message is being send by it and thus can be used as a first line of defense for alerting fraudulent accounts.

Architecture= The main thinking behind the system is honest users tend to link their various accounts to each other like twitter to other blogs etc. but it is very costly affair if same done by spammers as each service has its own spam detection algorithm. The Figure 2.5 is architecture of system, here firstly input data is fed from the twitter through search module which performs a web search of the data for the username and display names. The results are feed to the analysis module where a number of noise reduction techniques are applied to throw out data which is useless and finally data is analyzed for the accounts, if any match is found user is legitimate else it is spammer.

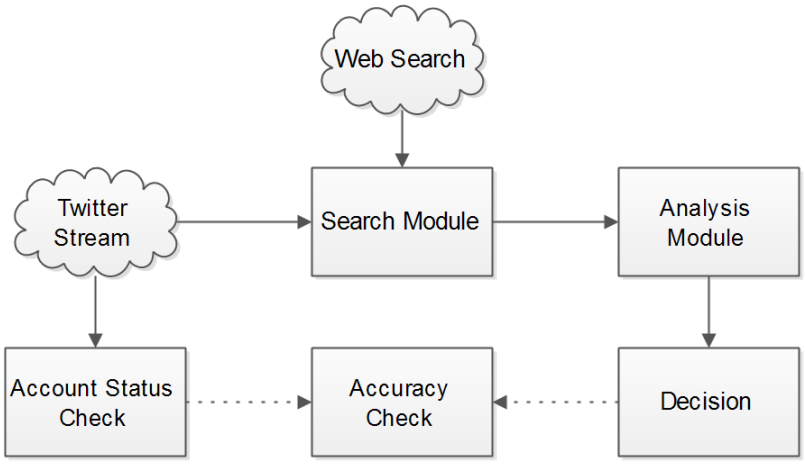

Figure 2.5: Web Search(Architecture) [23]

2.2.2 Message Based Detection

Lin et al.

In this paper [40] author is proposing a framework which can detect the spam from even a single message, as more and more spam messages are send by normal users or popular users. Author tries to extract the content features from the messages which can be used to detect the spams. The technique is based on the machine learning algorithm.

Architecture= The detection problem is treated as a binary classification problem

where positive are spam and negative are honest users. Various features are extracted from the messages like lexical features, status features, user features. In lexical features the text is divided in unigrams, bigrams, and trigrams tokens and using these binary features can be created. Status features include external URL in the posts. User's features include number of followers and followers which can be used to classify the content present in the post. For classification various classifiers are used like Naive Bayes, Support Vector Machine and Logistic Regression for machine learning.

Charles et al.(REPLOT)

In the paper [48] author has given a framework which combine the authorship attribution technique and behavioral analysis for detection and characterization of social campaigns. The process is done in 3 steps- firstly identification of the suspicious profile is done with help of behavioral analysis, then the connection between the suspicious profiles are identified using authorship attribution and temporal similarity, in last a clustering algorithm is run to identify suspicious campaigns.
Architecture= The framework is based on combination of content based (NAUNCE methodology [36])) and behavioral analysis(SPOT tool [36]).Figure 2.6 give the architecture of the framework. The first step involves the identification of the suspicious users for which a harvester module is connected to the OSN which collects and stores the activities by various users in database. A profile detection algorithm is run against selected profiles. Then a classification is performed which identify the suspicious and honest users. Two constraints for the step are rapidity of analysis and the number of profiles treated. Next step is to identify the relationship between the suspicious profiles by using content similarity features. Then a temporal similarity is found based on messages. Both of the features contribute to find measure of similarity between profiles. In last the clustering algorithm is run to highlight campaigns. In this step profiles are clustered which have significant similarities. The clusters are characterized based on behavior and content bases to clearly identify their intent. Clustering helps in understanding the strategies used by malicious users and to find other undetected spamming profiles by applying

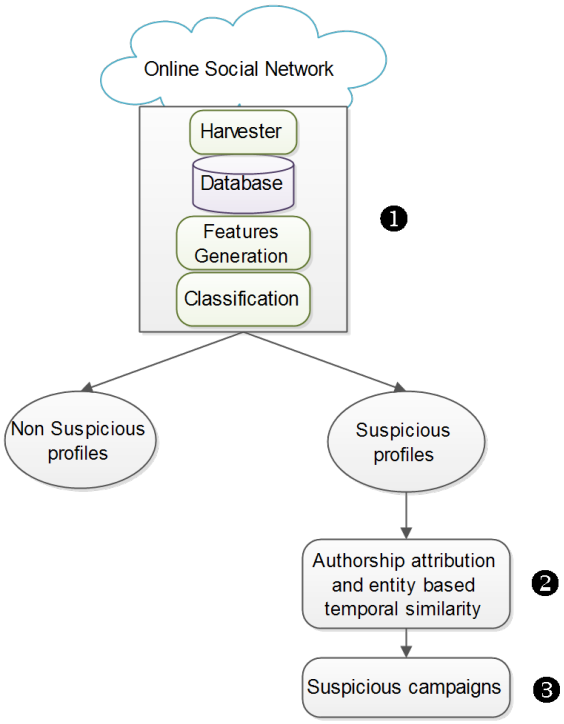

Figure 2.6: Replot(Architecture) [48]

features like affinity to cluster.

Quanyan et al.(SODEXO)

In the paper [71] Author has used a proactive approach for improving the security in OSNs by catching social botnets [13, 32] with honeypots and for this author has given a framework named SODEXO (Social Deception and Exploitation through hOneybots) which can be integrated with existing network to create deceptive honeypots to leverage botnets and getting information from them and author has also given a game theoretic model for the better understanding of various trade-off faced by honeybots in analyzing the behavior to exploit spammers and gain information.

Architecture= System architecture of the SODEXO include honeybot-based defense. Honeybot are fake accounts which imitate real user in OSN [38] which have bots associated with them which can impersonate an infected node. Figure 2.7 shows the architecture of the SODEXO, it have two main components Honeybot deployment (HD) and honeybot exploitation (HE) and the work of both of them is coordinated with the help of a Protection and Alert system(PAS) which gath-

ers the information in real time and raise alarms in social network. In honeybot deployment an account is created on a Social networking website and account profile data is fed to imitate a real user. After deployment this profile sends friends request to random users until the user have at least desired number of neighbors. After this honey bot monitors the message traffic of neighbors to find any presence of botnets and if found any trace that botnet. In Honeybot exploitation information is collected from the botnets in form of command and control messages. The protection and Alert system provides security policy for the honeybot detection. It uses the data mining and machine learning techniques to infer the structure of the botnets in network traffic [28] by which botmasters can be detected and removed.

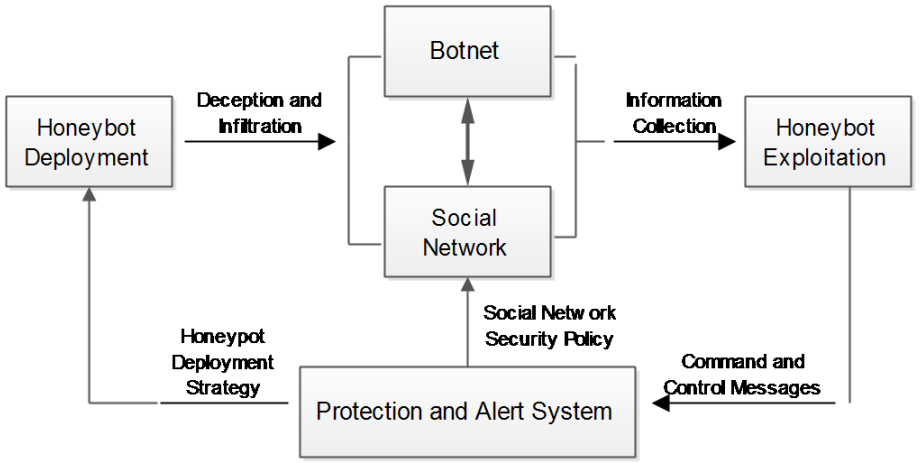

Figure 2.7: SODEXO(Architecture) [71]

2.2.3 URL Based Detection Techniques

Md Sazzadur et al.(MyPageKeeper)

In the paper [51] author has designed an application MyPageKeeper to protect the users form the socware. Socware include all the criminal and parasitic activities in the OSN, including anything which annoys, hurts or is used to make money of the users, spreading of malware, false rewards etc. So basically socware appear on the users wall with some post which either contain URL which points to malicious content. The major contribution of the application is accurate and efficient detection of socware by continually checking the walls and the news feeds of the

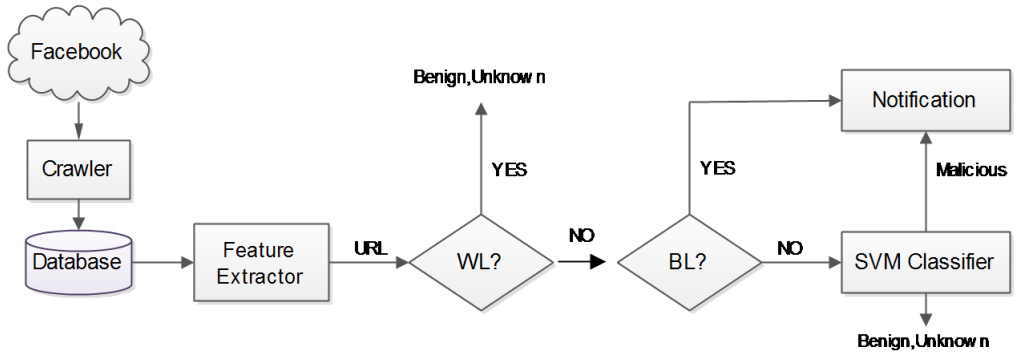

Figure 2.8: MyPageKeeper(Architecture) [51]

subscribed users.

Architecture= Talking about the components of the application MyPageKeeper (Figure 2.8) has 6 functional components . First is user authorization module in which application get the required permissions from the users to access the information about the user from the profile, Next module is Crawling module which periodically collects the data or posts from the user wall and news feed which contains the URL, Next is the features extraction module which classify the posts according to the embedded URL, Next module is classification module which uses the machine learning classifiers, it check the URL got from the previous step and classify each URL to be socware or not and if it is found to be socware all the posts which have the same URL are labeled socware, Next is notification module notifies the users about the presence of socware and user can configure which type of notification s/he wants either email, comment etc., and the last is User feedback module which is used to improve the ability of the application by getting inputs from the users. Identification is mainly done in classification module where the classificaiton algorithm operate, firstly whitelists and blacklists(can be obtained from [7, 8, 6]) are compared with URL to improve efficiency and accuracy, then the URLs are evaluated by Support Vector Machines based classifiers in Machine learning which gives the binary output decision to indicate whether URL is malicious or not.

Sangho et al.(WARNINGBIRD)

Author has proposed Warning-Bird [39]which is a suspicious URL detection system and it uses correlated redirect chains of URLs in various tweets(Twitter [35]). As the resources of attacker are limited they reuse the portion of redirect chain and technique focus on the shared resources to detect suspicious URLs [42, 56].
Architecture= Warning-Bird have 4 major components, first is Data Collection which collect the tweets for URL and then crawl for redirection from URLs. For collecting tweet from timeline twitter streaming APIs is used and after getting URL crawling is done to find the IP addresses of the URLs and combining both forms a tweet queue. Next component is Feature Extraction which have subcomponents grouping identical domains, extracting features vectors and finding entry point URLs. This monitor tweet queue to check that enough tweet are collected or not and after collecting enough tweets all are popped from queue and shared IP are found between them and groups are formed with similar IP, then the entry point URL is found and various features are extracted from these URL redirect chain. Results are fed to training component which helps in retrieval of accounts status and in training the classifiers. Last component is Classification which executes classifier to classify malicious URLs.

2.2.4 Hybrid Techniques

F. Javier et al.

In this paper [45] author has given a framework for detection of spam based on a random walk algorithm which obtain Ranking of a page according to their spam likelihood and relevance [41]. In order to detect spam the author has used the content of the webpage which is used to characterize the web graph and a priori estimation of the spam likelihood is done. Each node is given two values in the graph which represent how good or bad a web page is based on its relation graph and its textual content. The approach uses some content based heuristics for characterization of link based algorithm, in such a way that information got from heuristics improves the rank of page which are obtained by graph based algorithm. The demotion of bad pages and promotion of good pages is finally done on the

bases of their content and their relation in the web graph.

Architecture= A spam based random walk algorithm is run which give more relevance to specific set of seeds [33, 47] and from this ranks can be given to web pages by reinforcing each page with a positive and a negative score. After computed two scores for each page, there is need two individual sets of seeds where first set of seed contains pages which are non-spam and second set contain only spam pages. This process of creating two sets can be made automatic by use of content based heuristics. Various metrics of the content based algorithm are compressibility which give size of the web page before and after being compressed and more compressed page is thought to be spam, next metric is fraction of use global popular words where a web page with high fraction of popular word is more likely to be spam, last metric is average length of the words where normal webpages have a bell shaped distribution on the other hand malicious have much higher values.

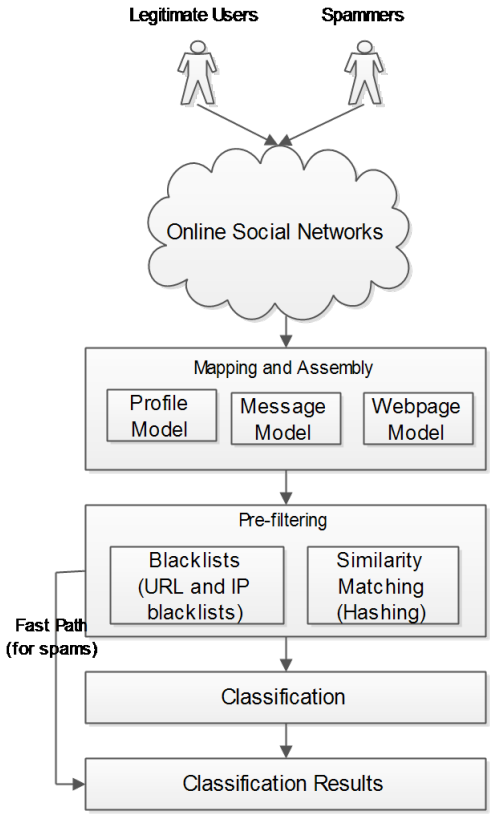

Figure 2.9: Random Walk(Architecture) [65]

Wang et al.

Author has proposed a framework [65] for detection of spammers which can be used across all the social networks. This technique avoids the need of separate spam detection mechanism for each social network. Author has also discussed some of the advantages of this technique which include more accuracy, blacklisting can be centralized etc. The framework has three major components - Mapping and assembly, Pre-filtering, Classification.

Architecture= Architecture of the framework (Figure 2.9) has three major components.

Mapping and Assembly- Mapping techniques is used to convert an object based on social network into framework defined standard model which is defined for object. The mapping is done automatically by feeding list of incoming attributes. Whereas Assembly is a process in which each model object is probed for associated objects and are assembled together.

Pre-filtering- various techniques are used to check incoming objects for spammers like blacklists, hashing, string matching. All though these techniques have shortcomings in detection but they significantly save a lot of time and can do coarse filtering.

Classification- The classification component defines three model namely Profile Model, Message model and Web Page model. The profile model is derived from Google Open Social Person API [5], the message model was derived from common attributes from the message, the web page model is derived from common HTTP session header information [66] Various supervised machine learning techniques are used to classify the incoming data, author has used Bayesian technique for detection of spam and non-spam. Author has built one classifier for each model and has used over 40 different classifiers for machine learning.

Chapter 3

Problem Statement

3.1 Gap in study

All supervised machine learning techniques have some inherent drawbacks when these are applied to large social networks. Labelling of training set is required for machine learning algorithms to work which need human intervention. Moreover this labeling is to be done again and again to maintain the effectiveness of the technique.

The linked based detections uses the human based trust relationship for the detection of spam and uses the network graph where nodes represent the users and the links represent the relation between two users. All these technique use the ground truth that social networks are fast mixing which is not true in each case. More over the detection is based on assumption where a node knows whole network.

The content based and behavioural based algorithms gave results based on similarity score which is used to find the cloned profiles in social networks. But these type of techniques cannot detect spams from compromised accounts and a study revealed that majority of spam messages are send by compromised accounts. All the exact string matching techniques which scan each message for spam, extract features from the messages which are fed to machine learning algorithm to find out the spams. But the spammer can easily bypass the technique by even small change in message content.

All the URL based detection techniques uses the URL content of the messages to detect the spams. Various features are extracted from messages which contain

URL, which are further fed to machine learning algorithm to detect spams. The technique fails if the spammer use URL shortners and technique can only catch spams which contain URL's.

3.2 Problem Definition

After studying the problem in the existing system, this conclusion has been derived that we need an unsupervised, distributed and decentralized technique to detect and remove spams from the social networks.
The technique should be an unsupervised one so that it has no overhead of labeling of training set for machine learning algorithm. The technique should be distributed and decentralized one and thus there is no need to collect all the messages at one place and because of which do not become the bottleneck for the network. The technique should use fuzzy based method to detect spams thus can catch normal as well as altered spam messages.To speed up the technique we can parallelize the technique and modify the code to exploit the multi-core and multi-thread features of the modern CPU's. The technique can also be implemented over Map Reduce platform thus can handle large networks easily without becoming bottleneck.

3.3 Objectives

- To design a fuzzy string based technique which can detect spams from social networks.
- To speed up the technique by parallelizing its processing.
- To implement and validate technique with huge dataset.

Chapter 4

Implementation

In this section we will discuss our approach about how we identify the spams from all the messages.

4.1 Work Flow

The workflow of the technique is given in Figure 4.1. In the Figure we can see we have two modules, first is feature extraction and second is fuzzy string matching algorithm. We will discuss each of them seperately.

The spam detection is treated as a binary classification problem where either a messages is spam or is non-spam. Each spam message is assigned a positive value '1' and a non-spam message is assigned a value '0'.

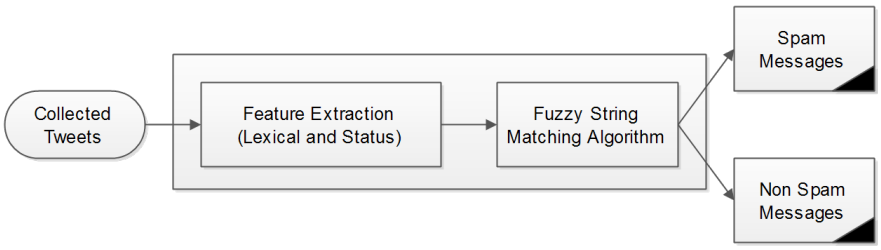

Figure 4.1: Work flow(Fuzzy String Matching algorithm)

4.1.1 Features Extraction

To detect the spam from the message stream, content features are extracted from the messages. Content features include the status and lexical features. Status features include the Length of message, presence of URL and presence of hot topics. The spam message are generally large in length and contains a URL to mislead the victim and include hot topic to increase the presence over the network so scanning for these features help in first filtering for the spam messages. Lexical features include the breaking of the messages into unigram, bigram, trigram tokens which excludes all the punctuations and any meaningless words.

4.1.2 Fuzzy String Matching Technique

The features extracted from the previous step including the lexical and status features are fed to the fuzzy string matching algorithm to detect the spam. The algorithm of the fuzzy is inspired from the findings of Chris et al. given in [25]. In his work Chris found that all spam contain words like free music, games, books, downloads etc. Table 1.1 shows all the spam content with the fraction of spam. Taking this as ground reality we designed a fuzzy string matching algorithm which detects these words in tokens. We created a spam word dictionary using the Table discussed before.

punctuations rithm 1 shows the fuzzy string matching technique, We can see that input for the algorithm are the twitter messages, the length threshold which is set keeping in mind the minimum length of the spam messages and the Spam dictionary. The output of the algorithm is indication for each message by value '1' or '0' to show spam or non spam. Each message is scan for url presence and whether the message is larger than the threshold, if false the message is discarded and if true each message is converted to lower case, all the punctuations from the message are removed and then it is split in tokens and stored in a list. Fuzzy string matching algo is applied over the words list and spam dictionary list, if there is a match then message is affixed with value '1' which show message is spam else with value '0' which show it is a non spam message.

Algorithm 1: Fuzzy String Matching Algorithm

```
Input  : Twitter messages Msg, i = 1,2,3,...n, Length threshold Len, Array of spam words Spam[n], variable x=0
Output: Msg1:0 or Msg1:1, Msg2:0 or Msg2:1, Msg3:0 or Msg3:1,...
Initialization;
for all i do
    if Msg contains url and length(Msg) > Len then
        msg ← lowercase(msg)
        msg ← remove(punctuations from msg)
        Words[] ← split(Msg)
        for all Words[] do
            for all Spam[] do
                if fuzzy_matching(Words[],Spam[]==True) then
                    x=1
                end
            end
        end
        if x == 1 then
            Return msg:1
        else
            Retrun msg:0
        end
    else
        Return NULL
    end
end
```

4.1.3 Multi-core and Multi-thread Variation

The results of fuzzy string are always better than exact string matching but the fuzzy string matching is very slow. To solve the problem and to speed up the process we implemented the fuzzy string matching algorithm over multi-core and multi-thread environment.

In the algorithm 2 we can see the implementation of multi-core fuzzy string matching. The output and input are same as normal fuzzy string matching algorithm. In this algorithm before scanning any message we find the available processor cores which are available for processing and we created same number of threads. For each thread a message is read and fuzzy string matching algorithm is applied for a message and spam and non spam messages are identified.

4.2 MapReduce Variation

Twitter is a very large online social network which accumulates around 6000 tweets per second, which add up to around 350,000 per minutes. If this huge number

Algorithm 2: Multi-core Fuzzy String Matching

```
    Input  : Twitter messages Msg, i = 1,2,3,...n, Length threshold Len, Array of spam words Spam[n], variable x=0
    Output: Msg1:0 or Msg1:1, Msg2:0 or Msg2:1, Msg3:0 or Msg3:1,...
 1  Initialization;
 2  Processors ← Number of available processors
 3  Create(Thread) ← Count(Processors)
 4  while msg.next !=Null do
 5      for each Thread do
 6          if Msg contains url and length(Msg) > Len then
 7              msg ← lowercase(msg)
 8              msg ← remove(punctuations from msg)
 9              Words[] ← split(Msg)
10              for all Words[] do
11                  for all Spam[] do
12                      if fuzzy_matching(Words[],Spam[]==True) then
13                          x=1
14                      end
15                  end
16              end
17              if x == 1 then
18                  Return msg:1
19              else
20                  Retrun msg:0
21              end
22          else
23              Return NULL
24          end
25      end
26  end
```

of messages is fed to a simple fuzzy string machine algorithm it will become a bottleneck and a normal system will not be able to handle such large load. To improve the efficiency, to distribute the load, to handle large number of messages, to decrease the response rate and to improve detection rate we implemented the code with Map Reduce.

The work flow of the technique is given in Figure 4.2. In Figure there are two main modules, extraction where the feature are extracted from the messages and next is fuzzy string matching algorithm which is implemented over Map Reduce platform.

In Map Reduce the input is firstly split by the mapper, this input is scrambled and is sorted, this sorted data is then fed to reducers which do the processing of the data and produce output, the output from all the reducers is combined to give the final output. The number of mapper and reducer can vary according to need. There can be either one mapper and one reducer which shows implementation of Map Reduce over single machine which is generally known as single node Map Reduce, or we can have multiple mapper and reducer which show Map Reduce

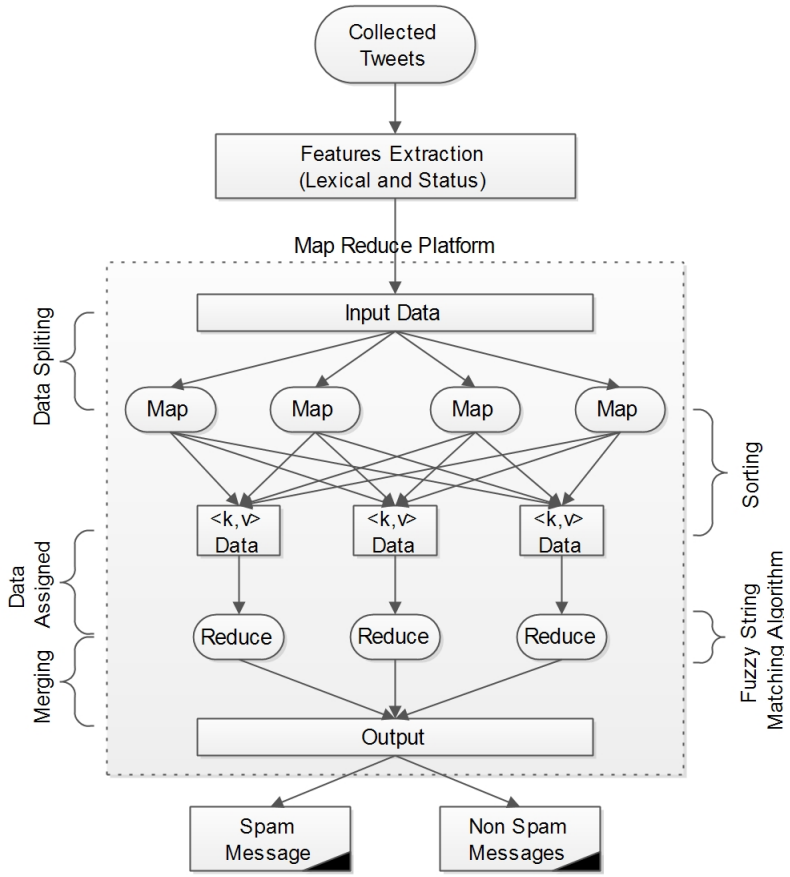

Figure 4.2: Work flow(Fuzzy String Matching algorithm over MapReduce platform)

implementation over multiple machine which is generally known as multi node Map Reduce. So depending on need, the nodes can be varied and the major advantage of Map Reduce is adding and removing of nodes can be done anytime. Thus the technique over Map Reduce is scalable too. The main two modules of the Map Reduce, mapper and reducer are discussed separately below

4.2.1 Mapper

As the name suggests mapper maps the input for the reducer. Mapper read the input files and split the input uniformly and writes this data over temporary storage. This temporary storage is scattered uniformly and then is sorted. This sorted input is fed to the reducer.

Algorithm 3 shows the mapper algorithm which helps in the preliminary filtering of spam messages. Here the messages are scanned for URL and if they have

URL and are greater than a threshold, they are affixed a value '0' else they are discarded. All the messages with affixed '0' are sorted and are fed to the reducer.

Algorithm 3: Mapper Algorithm

```
   Input  : Twitter messages Msg, i = 1,2,3,...n, Length threshold Len
   Output: Msg1:0, Msg2:0, Msg3:0,...
 1 Initialization;
 2 for all i do
 3 |   if Msg contains url and length(Msg) > Len then
 4 |   |   Return msg : 0
 5 |   else
 6 |   |   Return NULL
 7 |   end
 8 end
```

4.2.2 Reducer

As the name suggests reducer reduces the input fed by mapper and gives the output. It is the place where the actual processing is done and here the fuzzy string matching algorithm is implemented. The output from all the reducers is merged to give the final output.

Algorithm 4 shows the reducer algorithm which is applied to each message one at

Algorithm 4: Reducer Algorithm

```
   Input  : Msg1:0,Msg2:0,Msg3:0,...¿, Array of spam words Spam[n], variable x=0
   Output: Msg1:0 or Msg1:1, Msg2:0 or Msg2:1, Msg3:0 or Msg3:1,...
 1 Initialization;
 2 for all msg do
 3 |   msg ← lowercase(msg)
 4 |   Words[] ← split(Msg)
 5 |   for all Words[] do
 6 |   |   for all Spam[] do
 7 |   |   |   if fuzzy_matching(Words[],Spam[]==True) then
 8 |   |   |   |   x=1
 9 |   |   |   end
10 |   |   end
11 |   end
12 |   if x == 1 then
13 |   |   Return msg:1
14 |   else
15 |   |   Retrun msg:0
16 |   end
17 end
```

a time. Firstly punctuations are removed from the tweet message and the message is converted into token. These set of tokens are compared with the spam word dictionary using the fuzzy string matching algorithm. If there is a match the

message is spam and the affixed value '0' of the message is changed to '1', else if there is no match the affixed value remain unchanged. The output of all the reducers is merged together to get the final output where all the spam messages have affixed value '1' and all the non-spam messages have affixed value '0'.

4.3 Environment Setup

We implement the technique on the machine with Core - i7 processor, with 12 GB ram in master machine and 6 GB ram in other nodes. The operating system used was Ubuntu 14.04, JDK version used was 1.8.0_45 and the hadoop version was 1.2.1 (Table 4.1). On Master machine NameNode, TaskTracker, DataNode, Jobtracker, Secondary-NameNode are run and on other nodes machines DataNode and TaskTracker are run.

Table 4.1: Machine's details

Processor	Intel Core 3rd Generation i7-3770S Processor
Ram(Master/Slave)	(12 GB /6 GB) DDR3 1600MHz
Operating System	Ubuntu 14.04
Java JDK	JDK 1.8.0_45
Hadoop	1.2.1

Chapter 5

Results

In this section we evaluate our technique and find out how effective our technique is in detecting spams. We first create our dataset and then we test the dataset with the experiments.

5.1 Evaluation of Fuzzy String Matching algorithm

5.1.1 Dataset

Our dataset was created using the Twitter stream during March 2015 using Twitter API which is used to harness the tweets from the twitter and data was collected from random sample of all twitter messages. As the messages contained all the tweets including Users Post and Twitter control messages so we filterd the messages which had a url embedded in it. We also eliminated all the tweets which were in non-English language as our technique is based on english language only. This leaves us with 13,17,720 tweets which contain both spam and normal messages (Table 5.1).

Total Messages	Spam Messages	Non Spam Messages
13,17,720	41080	1276640

Table 5.1: Dataset for Fuzzy String Matching algorithm

Technique	Total Messages	Spam Detected	Time Needed(in sec)
Exact String Matching	13,17,720	41080	3.144
Fuzzy String Matching	13,17,720	43800	55.688

Table 5.2: Results of Exact and Fuzzy String Matching

Technique	Total Messages	Spam Detected	Time Needed(in sec)
Exact String Matching	13,17,720	0	4.246
Fuzzy String Matching	13,17,720	43800	55.688

Table 5.3: Results of Exact and Fuzzy String Matching on modified dataset

5.1.2 Ground Truth Of Dataset

Our dataset has around 13,17,720 tweets, when fed to exact string matching algorithm identifies 41080 spam messages out of all the tweets (Table 5.2). Now we convert the whole dataset to test the fuzzy technique. We change the alphabets of the spam words like(free to fre@, Discount to Disc0unt, etc.) to copy the obfuscation done by the spammers. Now we ran the converted dataset with exact string matching and it identified 0 spams.

Table 5.4: Fuzzy String Matching Variations(Multi-Core, Multi-Thread)

Fuzzy String Matching Variation	Spam Detection	Time Required(in sec)
Multi-Thread	43800	13.046
Multi-Core	43800	11.964

5.1.3 Performance Measure

Now testing the original dataset and modified dataset with fuzzy string technique. The techniques identified 43800 spams in both the cases which is much close to the exact string match (Table 5.2,5.3) with false positive less than 1% and false negative of 0% shown in Table 5.5 . When compared with exact string technique due to fuzzy nature the technique is much slower.

	False Positive (%)	False Negative (%)
Fuzzy String Matching	0.02	0.0

Table 5.5: False Positive and False Negative(Fuzzy String Matching Algorithm)

Testing the dataset with multi-core and multi-thread variant, they both identified the same number of spams from all the messages with improved time efficiency because of better utilization of available CPU resources. The results can be seen in Table 5.4 which shows multi-core was faster than multi-thread with same detection rate.

In general it wholely depend on the CPU architecture, operating system design and the application structure which decides whether multi core would be faster or multi thread. In our case multi core was faster than multithread.

5.2 Evaluation Of MapReduce Variation

5.2.1 Dataset

A new dataset is created using the Twitter stream during May'15 with the help of Twitter API which is used for getting the tweet messages from the twitter. The messages with a URL are filtered out and all non-English tweet are also removed from them as the proposed technique is based on english language only. This leaves us with 31,000 messages which have both spam and non-spam messages.

Table 5.6: Dataset for MapReduce Variation

Total Messages	Spam Messages	Non-Spam Messages
31,000	1829	29171

5.2.2 Dataset Ground Truth

The dataset is fed directly to exact string matching technique to find the existing spams in the dataset and the technique filtered 1829 spam messages. Same dataset was also fed to fuzzy based string matching technique to find any existing altered spam in the dataset but no such spam message was found and thus the dataset of 31,000 tweets have 1829 spam messages and no altered spam (Table 5.6). Additional 5000 alter spams are added to the dataset which include known number of one alphabet, two alphabets and three alphabets altered spams. Finally the

dataset have 36,000 tweets in total with 1829 non altered spams and 5000 altered spams.

5.2.3 Performance Measure

Single Node Cluster

Firstly technique was run on a single machine, the fuzzy based matching technique was set to catch spam with similarity of (.80) and was run against the dataset. It identified 4205 spam messages which include the spam messages with no alteration and spam messages with 1 alphabet alteration and some false positive and false negative too (Table 5.7). To further improve the detection rate the fuzzy based

Table 5.7: Results for the MapReduce Variation

Fuzzy String Matching Variation	Total messages	Total Spams	Spam Detected	False Positive	False Negative
Similarity(0.80)	36000	6829	4205	374	2998
Similarity(0.70)	36000	6829	5903	558	1484

technique was set to catch spam with similarity of (.70) which identified 5903 spams which significantly improved the detection of spam messages as now the technique was able to catch spam messages with no alteration, 1 alphabet alteration and 2 alphabet alteration but it also increased the false positive rate(Table 5.7). But the technique was not able to catch the 3 alphabet altered spams . If we further try to improve the detection rate by reducing the similarity further, it exponentially increased the false positive rate of the technique. The accuracy when the similarity is set to (.80) is 90.6% which increases to 94.3% when the similarity is lowered to (.70). The FP and FN for the similarity (0.80) is 0.12 and 0.439 respectively. Whereas FP and FN for the similarity (.70) is 0.19 and 0.217 respectively. It can be seen that when we tried to decrease the false negative rate the false positive rate increased. The variation of True positive and False positive can be seen in Figure 5.1 and in Figure we can see that the value of True positive increase significantly when we decreased the similarity to (.70).

The Figure 5.2 shows the result of precision and recall for the technique. It shows the results for both the variation and it can be seen in the Figure that the tech-

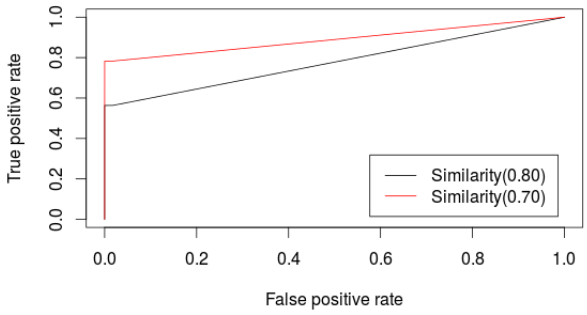

Figure 5.1: True Positive vs False Positive

nique achieves high recall and precision when similarity is set to (.70).

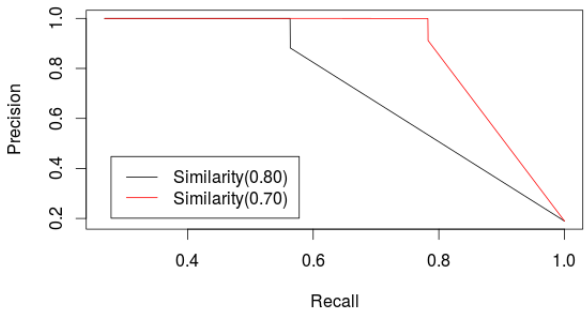

Figure 5.2: Precision vs Recall

The Figure 5.3 show the sensitivity and specificity variation for the technique and it can be seen in the Figure that similarity (0.70) outperforms similarity (0.80). It can be noted that the value of sensitivity increase for the same specificity when we lower the similarity to (.70).

Multi Node Cluster

Now the technique is run on multinodes as well to see the results of map reduce. Map Reduce is used to handle large dataset. Table 5.8 shows the results of running the technique with increasing the number of nodes. The original dataset used in previous evaluation is very small, to see the advantages of the map reduce multiple instances of same dataset is used which can mimic the large dataset. Here x10

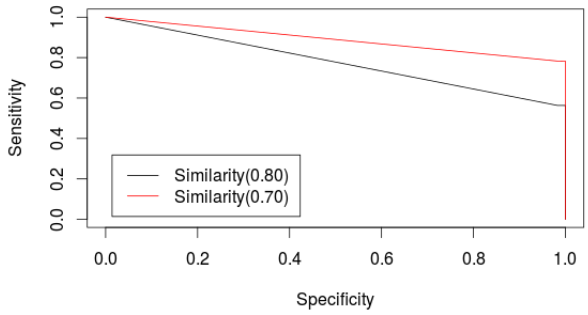

Figure 5.3: Sensitivity vs Specificity

means that 10 instances of same dataset are fed to the technique and so on. In the Table 5.8 we can see that if we increase the size of dataset the time increases linearly which is very much needed if we want to handle large databases.

In Figure 5.4 we can see the linear growth of time when we increase the dataset in

Table 5.8: Map Reduce and Dataset Variation results

Nodes ↓ /Dataset →	x10	x20	x30	x40	x50	x100	x150	x200	x250	x300	x350	x400	x450
Master	**0:14**	0:24	0:32	0:36	0:43	1:18	1:44	2:29	2:54	3:33	4:03	4:40	5:22
Master+1Node	0:16	**0:19**	**0:21**	**0:24**	**0:28**	1:05	1:11	1:53	1:57	2:35	2:45	3:05	3:35
Master+2Node	0:17	0:20	0:23	0:27	0:31	**0:49**	**1:07**	**1:27**	**1:45**	2:25	2:45	3:01	3:27
Master+3Node	0:18	0:20	0:25	0:27	0:31	0:53	1:11	1:32	2:00	**2:21**	2:40	3:01	3:28
Master+4Node	0:18	0:21	0:26	0:30	0:36	0:54	1:13	1:34	2:03	2:24	2:39	3:03	3:31
Master+5Node	0:18	0:21	0:25	0:27	0:35	0:56	1:15	1:36	2:00	2:23	**2:36**	**2:56**	**3:24**
Master+6Node	0:19	0:21	0:26	0:30	0:34	0:56	1:15	1:37	2:01	2:25	2:47	3:04	3:34
Master+7Node	0:19	0:22	0:26	0:30	0:34	0:55	1:16	1:37	2:03	2:25	2:45	3:05	3:33

case of master with 0 nodes, Same nature is also shown by all the other variations too. The time required by the technique also decrease significantly if the number of nodes are decreased, as same work can be parallely divided onto many machine. But it can be seen that after this decreasing time phase the time required tend to increase again as overhead of dividing the same work to more machines increases and thus the time required also increases (Figure 5.5).

Various finding of running the technique over map reduce platform, varying the nodes and dataset are-

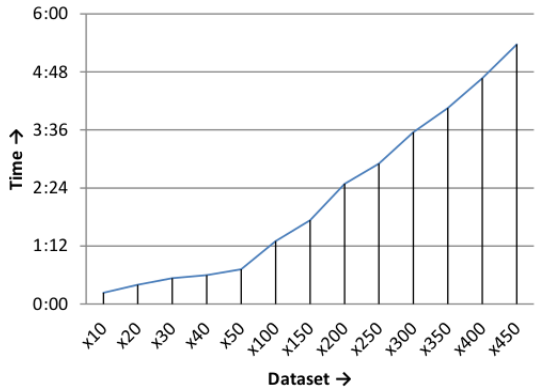

Figure 5.4: Performance for dataset vs Time on Master node

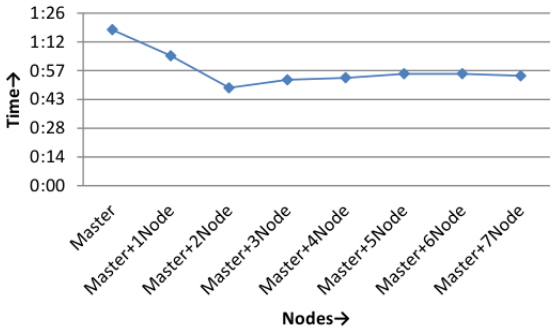

Figure 5.5: Performance of No. of Nodes over a fix dataset

- If the size of dataset increases, the time required for a Map Reduce variation to run the technique also increases linearly.
- If the number of nodes is increased, the time required to handle the same dataset decreases significantly.
- But if we further decrease the nodes, the time required tend to increase because of increased overhead to handle more machines for same piece of work.
- Many times it was seen that the time required for a variation tend to fluctuate enormously, main reason for such fluctuation can be irresponsive node or a node cannot reply in fixed time to master or if the data is lost over the network and hence the master cannot give result on time. Master either returns error or processes the work on its own which eventually results in much increase in time required when compared with normal execution.

Chapter 6

Conclusion and Future Scope

The thesis explains the problem of spams in the online social networks like twitter and how spammers can evade the existing techniques by just simple obfuscation of spam content. We proposed a new technique which is based on fuzzy string matching which can even catch the obfuscated spams and more importantly spams from the compromised accounts. As the majority of spams originate from compromised accounts, the technique proposed is a major breakthrough for spam catching in online social networks.We first proposed a model to run a simple fuzzy string match algorithm. To improve the time efficiency altered the code to use multi-core and multi-thread. The technique uses Map Reduce platform for implementation which makes the technique able to handle large dataset easily without becoming a bottleneck for the network. The FP and FN for the technique was very low and the variation in number of the nodes also showed that the time required to handle a dataset also decrease when we increase the number of nodes.

As future work, the technique can be altered to use machine learning at place of using static dictionary for catching spams. Moreover the technique can also be implemented using GPU which can speed up the process and further improve the efficiency of the technique. Till date no technique has been given which is combination of any two technique, either(linked based with content based or linked based with machine learning or content based with URL based). So in future new techniques can be formulated which uses combination of any two detection techniques which will eventually help in decreasing the inherated drawbacks of each technique and moreover help in detection accuracy and efficiency.

References

[1] Fake accounts in facebook - how to counter it. http://tinyurl.com/5w6un9u, 2010.

[2] Stolen facebook accounts for sale. http://tinyurl.com/25cngas, 2010.

[3] Why the number of people creating fake accounts and using second identity on facebook are increasing. http://tinyurl.com/3uwq75x, 2010.

[4] Twitter usage statistics. http://tinyurl.com/5w6un9u, 2013.

[5] http://code.google.com/apis/opensocial/, 2014.

[6] Web-of-trust. http://www.mywot.com/, 2015.

[7] Google safe browsing api, June 26, 2014.

[8] Malwarepatrol- malware is everywhere!, June,2015.

[9] Linkedin q3 2013 earnings call, slideshare, October 29, 2013.

[10] Facebook company info. https://newsroom.fb.com/company-info, September,2009.

[11] L. A. Adamic, J. Zhang, E. Bakshy, and M. S. Ackerman. Knowledge sharing and yahoo answers: everyone knows something. In *Proceedings of the 17th international conference on World Wide Web*, pages 665–674. ACM, 2008.

[12] L. Akoglu, M. McGlohon, and C. Faloutsos. Oddball: Spotting anomalies in weighted graphs. In *Advances in Knowledge Discovery and Data Mining*, pages 410–421. Springer, 2010.

[13] J. Baltazar, J. Costoya, and R. Flores. The real face of koobface: The largest web 2.0 botnet explained. *Trend Micro Research*, 5(9):10, 2009.

[14] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th international conference on World wide web*, pages 551–560. ACM, 2009.

[15] D. Boyd and J. Heer. Profiles as conversation: Networked identity performance on friendster. In *System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on*, volume 3, pages 59c–59c. IEEE, 2006.

[16] G. Brown, T. Howe, M. Ihbe, A. Prakash, and K. Borders. Social networks and context-aware spam. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 403–412. ACM, 2008.

[17] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *NSDI*, pages 197–210, 2012.

[18] D. Chakrabarti. Autopart: Parameter-free graph partitioning and outlier detection. In *Knowledge Discovery in Databases: PKDD 2004*, pages 112–124. Springer, 2004.

[19] P. Chirita, J. Diederich, and W. Nejdl. Mailrank: Global attack-resistant whitelists for spam detection. In *Conference on Information and Knowledge Management (CIKM)*. Citeseer, 2005.

[20] D. J. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. Mapping the world's photos. In *Proceedings of the 18th international conference on World wide web*, pages 761–770. ACM, 2009.

[21] G. Danezis and P. Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *NDSS*, 2009.

[22] A. Felt and D. Evans. Privacy protection for social networking apis. *2008 Web 2.0 Security and Privacy (W2SP'08)*, 2008.

[23] M. Flores and A. Kuzmanovic. Searching for spam: detecting fraudulent accounts via web search. In *Passive and Active Measurement*, pages 208–217. Springer, 2013.

[24] P. W. Fong. Preventing sybil attacks by privilege attenuation: A design principle for social network systems. In *Security and privacy (SP), 2011 IEEE symposium on*, pages 263–278. IEEE, 2011.

[25] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 35–47. ACM, 2010.

[26] L. A. Goodman. Snowball sampling. *The annals of mathematical statistics*, pages 148–170, 1961.

[27] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @ spam: the underground on 140 characters or less. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 27–37. ACM, 2010.

[28] G. Gu, J. Zhang, and W. Lee. Botsniffer: Detecting botnet command and control channels in network traffic. 2008.

[29] J. Huang, Y. Xie, F. Yu, Q. Ke, M. Abadi, E. Gillum, and Z. M. Mao. Socialwatch: detection of online service abuse via large-scale social graphs. In *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, pages 143–148. ACM, 2013.

[30] T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer. Social phishing. *Communications of the ACM*, 50(10):94–100, 2007.

[31] M. Jiang, P. Cui, A. Beutel, C. Faloutsos, and S. Yang. Catchsync: catching synchronized behavior in large directed graphs. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 941–950. ACM, 2014.

[32] G. Keizer. Worm spreads on facebook, hijacks users' clicks. *Computerworld*, 2008.

[33] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[34] C. Kreibich and J. Crowcroft. Honeycomb: creating intrusion detection signatures using honeypots. *ACM SIGCOMM Computer Communication Review*, 34(1):51–56, 2004.

[35] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.

[36] R. Layton, P. Watters, and R. Dazeley. Automated unsupervised authorship analysis using evidence accumulation clustering. *Natural Language Engineering*, 19(01):95–120, 2013.

[37] K. Lee, J. Caverlee, K. Y. Kamath, and Z. Cheng. Detecting collective attention spam. In *Proceedings of the 2nd Joint WICOW/AIRWeb Workshop on Web Quality*, pages 48–55. ACM, 2012.

[38] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers: social honeypots+ machine learning. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 435–442. ACM, 2010.

[39] S. Lee and J. Kim. Warningbird: Detecting suspicious urls in twitter stream. In *NDSS*, 2012.

[40] L. Liu and K. Jia. Detecting spam in chinese microblogs-a study on sina weibo. In *Computational Intelligence and Security (CIS), 2012 Eighth International Conference on*, pages 578–581. IEEE, 2012.

[41] L. Liu and M. T. Zsu. *Encyclopedia of database systems*. Springer Publishing Company, Incorporated, 2009.

[42] D. K. McGrath and M. Gupta. Behind phishing: An examination of phisher modi operandi. *LEET*, 8:4, 2008.

[43] A. Nazir, S. Raza, and C.-N. Chuah. Unveiling facebook: a measurement study of social network based applications. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, pages 43–56. ACM, 2008.

[44] NielsenWire. Twitter's tweet smell of success. `http://blog.nielsen.com/nielsenwire/online_mobile/twitters-tweet-smell-of-success/`, 2009.

[45] F. J. Ortega, C. Macdonald, J. A. Troyano, and F. Cruz. Spam detection with a content-based random-walk algorithm. In *Proceedings of the 2nd international workshop on Search and mining user-generated contents*, pages 45–52. ACM, 2010.

[46] P. Oscar and V. Roychowdbury. Leveraging social networks to fight spam. *IEEE Computer*, 38(4):61–68, 2005.

[47] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. 1999.

[48] C. Perez, B. Birregah, R. Layton, M. Lemercier, and P. Watters. Replot: Retrieving profile links on twitter for suspicious networks detection. In *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*, pages 1307–1314. IEEE, 2013.

[49] C. Perez, M. Lemercier, B. Birregah, and A. Corpel. Spot 1.0: Scoring suspicious profiles on twitter. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, pages 377–381. IEEE, 2011.

[50] M. B. Prince, B. M. Dahl, L. Holloway, A. M. Keller, and E. Langheinrich. Understanding how spammers steal your e-mail address: An analysis of the first six months of data from project honey pot. In *CEAS*, 2005.

[51] M. S. Rahman, T.-K. Huang, H. V. Madhyastha, and M. Faloutsos. Efficient and scalable socware detection in online social networks. In *USENIX Security Symposium*, pages 663–678, 2012.

[52] Z. Shan, H. Cao, J. Lv, C. Yan, and A. Liu. Enhancing and identifying cloning attacks in online social networks. In *Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication*, page 59. ACM, 2013.

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

[54] J. Song, S. Lee, and J. Kim. Spam filtering in twitter using sender-receiver relationship. In *Recent Advances in Intrusion Detection*, pages 301–317. Springer, 2011.

[55] L. Spitzner. The honeynet project: Trapping the hackers. *IEEE Security & Privacy*, 1(2):15–23, 2003.

[56] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 1–9. ACM, 2010.

[57] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 1–9. ACM, 2010.

[58] E. Tan, L. Guo, S. Chen, X. Zhang, and Y. Zhao. Unik: unsupervised social network spam detection. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 479–488. ACM, 2013.

[59] K. Thomas, C. Grier, D. Song, and V. Paxson. Suspended accounts in retrospect: an analysis of twitter spam. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 243–258. ACM, 2011.

[60] K. Thomas, C. Grier, D. Song, and V. Paxson. Suspended accounts in retrospect: an analysis of twitter spam. In *Proceedings of the 2011 ACM*

SIGCOMM conference on Internet measurement conference, pages 243–258. ACM, 2011.

[61] D. N. Tran, B. Min, J. Li, and L. Subramanian. Sybil-resilient online content voting. In *NSDI*, volume 9, pages 15–28, 2009.

[62] N. Tran, J. Li, L. Subramanian, and S. S. Chow. Optimal sybil-resilient node admission control. In *INFOCOM, 2011 Proceedings IEEE*, pages 3218–3226. IEEE, 2011.

[63] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove. An analysis of social network-based sybil defenses. *ACM SIGCOMM Computer Communication Review*, 41(4):363–374, 2011.

[64] A. H. Wang. Don't follow me: Spam detection in twitter. In *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, pages 1–10. IEEE, 2010.

[65] D. Wang, D. Irani, and C. Pu. A social-spam detection framework. In *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, pages 46–54. ACM, 2011.

[66] S. Webb, J. Caverlee, and C. Pu. Characterizing web spam using content and http session analysis. In *CEAS*. Citeseer, 2007.

[67] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov. Spamming botnets: signatures and characteristics. *ACM SIGCOMM Computer Communication Review*, 38(4):171–182, 2008.

[68] Y. Xie, F. Yu, Q. Ke, M. Abadi, E. Gillum, K. Vitaldevaria, J. Walter, J. Huang, and Z. M. Mao. Innocent by association: early recognition of legitimate users. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 353–364. ACM, 2012.

[69] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 3–17. IEEE, 2008.

[70] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Sybilguard: defending against sybil attacks via social networks. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 267–278. ACM, 2006.

[71] Q. Zhu, A. Clark, R. Poovendran, and T. Basar. Deployment and exploitation of deceptive honeybots in social networks. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 212–219. IEEE, 2013.

[72] A. Zinman and J. S. Donath. Is britney spears spam? In *CEAS*, 2007.

List of Publications

[1] A. Kumar and M. Singh. *A Survey on Social Spammers Detection Techniques.* IEEE Communicataion and Survey [Communicated].

[2] A. Kumar and M. Singh. *Social Media Spam Detection Using Fuzzy String Matching.* Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions), 2015 4th International Conference on. IEEE, 2015 [Accepted].

Video Presentation

video link is –