Identifying and Mitigating Cross-Platform Phone Number Abuse on Social Channels

By
Srishti Gupta

Under the supervision of Dr. Ponnurangam Kumaraguru

Indraprastha Institute of Information Technology Delhi
January, 2019

Identifying and Mitigating Cross-Platform Phone Number Abuse on Social Channels

By
Srishti Gupta

Submitted

in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

to the

Indraprastha Institute of Information Technology Delhi
January, 2019

Certificate

This is to certify that the thesis titled "**Identifying and Mitigating Cross-Platform Phone Number Abuse on Social Channels**" being submitted by **Srishti Gupta** to the Indraprastha Institute of Information Technology Delhi, for the award of the degree of Doctor of Philosophy, is an original research work carried out by her under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree / diploma.

Supervisor Name: Dr. Ponnurangam Kumaraguru, 'PK'

January, 2019

Department of Computer Science

Indraprastha Institute of Information Technology Delhi

New Delhi 110 020

Keywords: Online social networks, phone number, spam campaign, scam, cross-platform, heterogeneous networks

Abstract

Online Social Networks (OSNs) have become a huge aspect of modern society. Social network usage is on a rise, with over 2 billion people using it across the globe, and the surge is only expected to increase. Online Social Networks not only aid users to engage in online conversations, but also help them in staying updated with current news / trends, keep up with friends, and participate in online debates etc. Some experts suggest that OSNs will soon become the new search function – people will search lesser time navigating through Internet websites, but consume the content available on OSNs. A significant fraction of OSN spam research has looked at solutions driven by URL blacklists, manual classification, and honeypots. Since defence mechanisms against malicious / spam URLs have already matured, cybercriminals are looking for other ways to engage with users. Telephony has become a cost-effective medium for such engagement, and phone numbers are now being used to drive call traffic to spammer operated resources. The convergence of telephony and the Internet with technologies like Voice over IP (VoIP) is fueling the growth of Over-The-Top (OTT) messaging applications (like WhatsApp, Viber) that allow smartphone users to communicate with each other in myriad ways. These *social channels* (OSNs and OTT applications) and VoIP applications (like Skype, Google Hangouts) are used by millions of users around the globe. In fact, the volume of messages via OTT messaging applications has overtaken traditional SMS and e-mail. As a result, these social channels have become an attractive attack vector for spammers and malicious actors who are now abusing it for illicit activities like delivering spam and phishing messages.

Therefore, in this work, we aim to detect cybercriminals / spammers that use phone numbers to spread spam on OSNs. We divide this thesis into 4 parts – (1) Understanding the threat landscape of phone attacks on OTT messaging applications leveraging information from OSNs, (2) Uncovering the spam ecosystem on OSNs and identifying spammers which contribute in spreading spam, (3) Evaluating the trustworthiness of current caller ID services and machine learning models that identify spam calls / spammers, (4) Proposing a robust phone reputation score for identifying spam phone numbers on OSNs.

We first focus our attention on understanding various ways in which spammers can attack OTT messaging application users by leveraging information from OSNs. To understand the effectiveness of such attacks, we do an extensive online crowdsourced study to identify highly impactful phone based attack. Further, we list down the factors that govern why a user falls to phone based attack on OTT messaging applications. Our analysis revealed that social phishing attacks are most successful to lure victims. In addition, victims are deficit in regulating OTT messaging applications usage, hence vulnerable to attacks.

Next, we identify and characterize the spam campaigns that abuse a phone number on OSNs. We create ground truth for spam campaigns that operate in different parts of the world like Indonesia, UAE, USA, India, etc. By examining campaigns running across multiple OSNs, we discover that Twitter detects and suspends ∼93% more accounts than Facebook. Therefore, sharing intelligence about abuse-related user accounts across OSNs can aid in spam detection. According to our 6 months dataset, around ∼35K victims and ∼$8.8M could have been saved if intelligence was shared across the OSNs. In addition, we analyse the modus operandi of several campaigns to understand the monetization model of spammers behind such attacks. Finally, we compare the characteristic behavioral difference between the spam and legitimate phone based campaigns.

We further look at the effectiveness of caller ID applications that identify an incoming phone call as spam. These applications are vulnerable to fake registration and spoofing attacks which make them inefficient in correctly identifying spammers. Further, we explore that supervised machine learning models to identify spammers are prone to manipulation, therefore, not a reliable solution.

To build a robust solution to uncover spammers, we model OSNs as a heterogeneous network by leveraging various interconnections between different types of nodes present in the dataset. In particular, we make the following contributions – (1) We propose a simple yet effective metric, called Hierarchical Meta-Path Score (HMPS) to measure the proximity of an unknown user to the other known pool of spammers, (2) We design a feedback-based active learning strategy and show that it significantly outperforms three state-of-the-art baselines for the task of spam detection. Our method achieves 6.9% and 67.3% higher F1-score and AUC, respectively compared to the best baseline method, (3) To overcome the problem of less training instances for supervised learning, we show that our proposed feedback strategy achieves 25.6% and 46% higher F1-score and AUC respectively than other oversampling strategies. Finally, we perform a case study to show how our method is capable of detecting those users as spammers who have not been suspended by Twitter (and other baselines) yet. We finally use spammer metrics to design a phone reputation service, called SpamDoctor [1] that can flag a potential bad phone number.

In conclusion, this thesis aims to bring out methods to detect spammers abusing phone numbers on Online Social Networks. We propose methods to give a reputation score to phone numbers such that the score is tolerant against external manipulation. To summarize, the research contributions of this thesis are - (1) Building automated framework to evaluate the effectiveness of phone based attacks on OTT, (2) Building automated detection method to identify phone based spam campaigns and the users behind it, (3) Evaluating the trustworthiness of current caller ID services to detect spam calls, (4) Supervised detection method to identify spammers and building SpamDoctor to flag phone numbers abused on OSNs.

[1] `http://labs.precog.iiitd.edu.in/phonespam/`

Dedicated to my parents and sister

Acknowledgements

This thesis is a culmination of my PhD journey which has been a roller coaster ride accompanied with encouragement, hardship, trust, and frustration. As I see myself fulfilling this journey, I would like to thank many people who contributed in accomplishing this huge task.

Firstly, I would like to express my sincere gratitude to my advisor Prof. Ponnurangam Kumaraguru for providing continuous support and guidance throughout my PhD journey. He has been a tremendous mentor, I am grateful to him for allowing me to grow as a researcher. His advice on my PhD research and career goals has been priceless. I am grateful for his valuable advice, constructive criticism, positive appreciation, and counsel throughout the journey. His zeal for unflinching courage, passion, conviction, and perfection has always inspired me to do more and push my limits. He has made the process a smooth ride for me; I sincerely thank him from bottom of my heart and will be truly indebted to him throughout my life time.

I am thankful to my monitoring committee members, Prof. Mustaque Ahamad and Prof. Sambuddho Chakravarty for giving honest and timely reviews on my work and helping in shaping it better. Prof. Mustaque has been an incredible support of strength during turbulent times when I was away from IIIT-Delhi under his wings for a year. His positivity and you-can-do-it attitude helped me sail through the times when stakes were high. I found a family in the foreign land that I shall cherish forever; my earnest thanks to him for believing in me when it was most required. I would also like to thank Dr. Payas Gupta, a good friend, who I see and thank as a mentor in my journey. His vision, sincerity, passion, and dedication to do something different is contagious.

I have been lucky to have worked with smart students like Dhruv, Abhinav, Arpit, Gurpreet, and Saksham during my PhD journey. I thank them for the stimulating discussions we had, for the sleepless nights we worked together before deadlines, and for all the fun we had in the last couple of years. I would also like to thank my professional siblings, Dr. Paridhi Jain for being the best critique, Anupama Aggarwal for lending a helping hand in literally anything when I needed it, Dr. Niharika Sachdeva for giving positive energy, and Dr. Prateek Dewan for making me laugh with his jokes. A special thanks to Dr. Siddhartha Asthana for his words-of-wisdom that shall stay with me forever. I would like to thank my fellow labmates and all the members of Precog; they were

always beside me during the happy and hard moments to push and motivate me. I would also like to thank my friends Shalini, Arun, Vidushi, Aditi, and Tarun who have supported me in this journey, pepping me up with their positivity, and incented me to strive towards my goal. You guys were always real pals! I would also thank other faculty and administrative staff at IIIT-Delhi for their hard work in assisting students and for their timely help; a special shout out to Pooja Sagar. They help create an excellent learning and research environment at IIIT-Delhi.

Finally, I acknowledge the people who mean a lot to me, my mom and dad. A special shout out to my dear sister, Shaifali for being younger yet wiser than me and motivating me throughout the journey. Words cannot express how grateful I am to them for all the sacrifices they have made on my behalf. All their blessings and prayers has helped me sustain so far. I thank them for supporting me in the countless moments when there was no one to answer my queries. I would never be able to pay back the selfless love and care you gave me all these years.

I owe thanks to a very special person, my fiancè, Tushar, who has been a never-ending source of encouragement and motivation. You are the best thing that has happened to me; thanks for bringing a breath of fresh air in my life. My heartfelt regard goes to my father in law, mother in law, and sister in law for their love and moral support. I consider myself the luckiest in the world to have such a caring and loving family; I thank each one of them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Emergence of Social Channels

It is undeniable that Online Social Networks (OSNs) have become a huge aspect of modern society. Social network usage is on a rise, with over 2 billion people using it across the globe, and the surge is only expected to increase [37]. Online Social Networks not only aid users to engage in online conversations, but also help them in staying updated with current news / trends, keep up with friends, and participate in online debates etc. Some experts suggest that OSNs will soon become the new search function – people will search lesser time navigating through Internet websites, but consume the content available on OSNs. Social networking is one of the ways in which Internet marketers and website owners would boost the visibility of their websites. The benefits of social network marketing for business is being leveraged by business owners, large and small. Brands can use visual content on their Social networks to increase engagement and inspire sharing and viral marketing. Search engines now rank content based on social conversations and sharing, not just websites alone. While these social networks channels can be used to businesses for clever and effective marketing, the increasing popularity of social networks channels has attracted a cadre of criminals who craft large-scale phishing and spam campaigns targeted against OSN users. Traditionally, spammers have been driving traffic to their websites by luring users to click on URLs in their posts on OSNs [85,87,179]. A significant fraction of OSN spam research has looked at solutions driven by URL blacklists [85, 178], manual classification [52], and honeypots [122, 174]. Since defence mechanisms against malicious / spam URLs have already matured, cybercriminals are looking for other ways to engage with users. Telephony has become a cost-effective medium for such engagement, and phone numbers are now being used to drive call traffic to spammer operated resources (e.g., call centers, Over-The-Top messaging applications like WhatsApp). OTT messaging applications refer to the one where content providers distribute streaming media as a standalone product directly to viewers over the Internet, bypassing telecommunications, multichannel television, and broadcast television

platforms that traditionally act as a controller or distributor of such content.

The convergence of telephony and the Internet with technologies like Voice over IP (VoIP) is fueling the growth of Over-The-Top (OTT) messaging applications [1] that allow smartphone users to communicate with each other in myriad ways. These *social channels* [2] (like WhatsApp, Viber, WeChat), [3] and VoIP applications (like Skype, Google Hangouts) [4] are used by millions of users around the globe. In fact, the volume of messages via OTT messaging applications has overtaken traditional SMS [16] and e-mail [180]. As a result, these social channels have become an attractive attack vector for spammers and malicious actors who are now abusing it for illicit activities like delivering spam and phishing messages. For example, unsolicited messages like investment advertisements, adult conversation advertisements, and random contacts requests were seen to propagate on WhatsApp [35].

A phone number is a personally identifiable piece of information with which an individual can be associated uniquely, in most cases [197]. Although there exist burner phones [5] in some countries where a phone number may not be reliably associated with a person, in a large number of cases, phone numbers are linked to a wealth of information about their owners like name and place where he / she lives. OTT messaging applications use phone numbers for user authentication and communication. Authentication is typically done when a user registers with the application by providing his / her phone number and the validity of the phone number is verified by delivering an SMS message to it. Such phone numbers are a verified part of user identity because one needs to obtain a physical SIM card and complete the verification process of a service provider to obtain a phone connection. Service providers often require personal information to setup an account. Fraudulent communication carried out with phone numbers has already resulted in loss of millions of dollars to individuals and organizations where spammers / scammers reach out to their victims [143, 149, 173] despite the fact that phone numbers are considered private information and not easily exposed by many platforms. Attackers who want to target certain users may prefer phone numbers over other identities like e-mail addresses and online social identifiers due to multiple reasons – a) Phone numbers are ubiquitous due to smartphone penetration growing in all population segments of society, both rural and urban, and amongst all age groups too [99]. Therefore, attackers can expect more reachability, in terms of potential victims, while abusing phone numbers; b) In most countries, a verification process is required before someone can obtain a phone number. Because of this, attackers cannot obtain phone numbers as easily as e-mail addresses and online social identities which can be easily created and faked. As a result, there is a greater degree of trust associated with phone

[1] An over-the-top (OTT) application is any app or service that provides a product over the Internet and bypasses traditional distribution. Services that come over the top are most typically related to media and communication and are generally, if not always, lower in cost than the traditional method of delivery [15]

[2] In this thesis, we cumulatively call OSNs and OTT messaging applications as social channels

[3] http://marketingland.com/four-top-six-social-networks-actually-chat-apps-115168

[4] http://beebom.com/2015/09/best-voip-apps

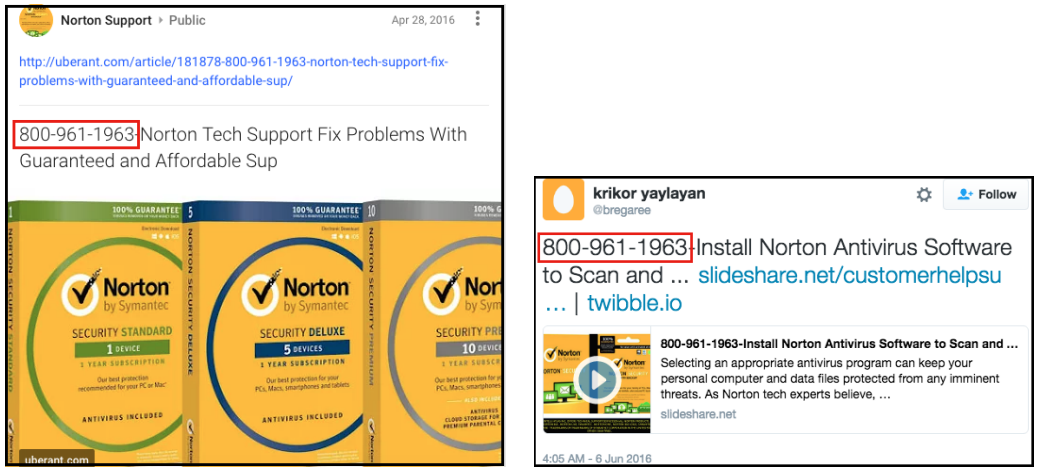
[5] https://www.puretalkusa.com/blog/what-is-a-burner-phone/

numbers as compared to other user identifiers; c) Phone numbers are personal and more persistent. People generally retain the same phone number for a long time due to the cost associated with it, where as, one can have multiple e-mail address and online identities. As victims would be using the same phone number, attackers can abuse it to increase the success rate of their attacks. On the other hand, due to multiplicity nature of e-mail address, the success rate for attackers to find and exploit currently being used e-mail addresses would be low; d) Phone calls and text messages are synchronous in nature and have faster response time than e-mail. This time sensitivity can be leveraged by the attacker to his / her benefit; e) We have mature and effective defenses against e-mail spam but this is not true for phone and messaging spam. There exist services like Truecaller that warn against potential spam but are ineffective because, first, the intelligence from the online channel is not getting integrated into these mobile application blacklists, like in the 26 case of 800notes.com and Malwarebytes, and secondly, call-blocking mobile applications are geared more towards blocking inbound phone communication (IPC), such as unverified robocalls, rather than blocking outbound phone communication (OPC) which is what happens when the victim is coerced into calling a phone number under the control of a spammer. Thus, attackers have an advantage when they exploit the telephony channel.

Telephony can be an effective tool for spammers because a recent study suggest that people fell victim to phone scams leading to a loss of $8.9 billion in United States alone, with the average person reporting receiving 23 spam calls per month [6]. Specifically, in the phone-based abuse of OSNs, spammers advertise phone numbers under their control via OSN posts and lure OSN users into calling these numbers. Advertising phone numbers reduce spammers' overhead of finding the set of potential victims who can be targeted via the phone. After that, they try convincing the victims that their services are genuine, and deceive them into making payments after a series of interactions [139]. To maximize their reach and impact, spammers disseminate similar content across OSNs. Figure 4.1 shows an instance of cross-posting behavior across OSNs in the case of Tech Support spam attack that has been in action for quite some time now.

While URLs help spammers attract victims to websites that host malicious content, phone numbers provide more leverage to spammers. Due to the inherent trust associated with the telephony medium and as spammers interact directly with victims over calls, spammers using phone numbers stand a better chance of convincing and hence are likely to make more impact. Besides, they can use fewer phone numbers as compared to URLs; a large number of URLs are required to evade filtering mechanisms incorporated by OSNs. [6] Moreover, the monetization and advertising channel, in this case, phone and OSN / Web respectively, are different in phone-based campaigns compared to a single channel (Web) used in URL-based campaigns. This requires correlation of abuse information across channels which makes it harder for OSN service providers to build effective solutions. Since the modus operandi in URL-based and phone-based spam campaigns is different, leaving phone-

[6] https://support.twitter.com/articles/90491



(a) GooglePlus (b) Twitter

Figure 1.1: Presence of the Tech Support campaign across OSNs (Twitter and GooglePlus).

based spams unexplored can limit OSN service providers' ability to defend their users from spam. Solutions have been built to educate users about URL-based spam [115], while limited education is available for phone-based attacks. This is evident from several well publicized and long running Tech Support spam campaigns (since 2008) that use phone numbers to lure victims leading to huge financial losses in the past, as reported by the Federal Bureau of Investigation [146]. Nevertheless, to the best of our knowledge, the space of OSN abuse using phone numbers is largely unexplored.

There are significant differences and unique challenges in the telephone ecosystem that require novel approaches. Many existing solutions have failed to overcome these challenges and, as a result, have yet to be widely implemented. We enumerate some challenges below –

- **Quick Response Time:** Unlike email, which can be queued for later analysis, a voice call has an immediacy constraint. The anti-spam system / solution should flag a incoming / outgoing call as spam within a short window of time to reduce the delay. Greater the call duration, more likely the malicious entity to coerce victim in his / her favor.
- **Difficulty in Handling Audio Streams:** The content of a voice call is difficult to parse and analyze: the content is an audio stream as opposed to the text of an email. While emails can be analysed even without the intervention of sender and receiver, content of a voice call is revealed for analysis only when the call is answered. In the former case, the anti-email system can take an action without the knowledge of sender / receiver, however, in the latter case, both the parties will get affected at the call level.
- **Lack of Useful Header Data:** Voice calls lack the rich header data of email. When a call arrives at the recipient, it does not contain much information. Caller ID application aid in

telling the name and location of the call, which might not be 100% accurate. An email header, on the other hand, contains well-defined, rich SMTP headers including the IP address and domain name of the email. In addition, it is hard to manipulate the email headers, a stark contrast to call / phone headers that can be omitted. For instance, phone spammers could use VoIP numbers to hide the entity or spoof numbers to forge the identity. One can identify the carrier, telecom circle, location from third party services, but these are rate-limited and not available for certain types of phone numbers, like toll-free phone numbers.

- **Caller ID Spoofing:** The Caller ID service is an information service that provides the recipient with information of the caller before answering the phone, which could be useful for blocking spam calls. However, caller ID fundamentally has no authentication mechanism and is easily spoofed. Thus, a solution built over caller ID functionality is vulnerable to spoofing attacks.

- **Temporary Disposable Numbers / Burner Phones**: With the advent of cheap VoIP numbers, spammers use and dispose off numbers quickly. Having a VoIP number will bring some great benefits. First it hugely cuts on the phone bill. Spammers can have just one phone number and just forward all your calls to that particular phone number. This reduces the cost of purchasing multiple phone numbers and skips the physical verification step to obtain a SIM card. Due to volatility procured through VoIP numbers, it is hard for anti-phone spam system to keep a track of phone numbers in operation. Spammers tend to quickly move to a new phone number to avoid tacking. In addition, the telecom providers do not have any information about VoIP numbers, so the header information is missing.

- **Absence of Phone Blacklists**: While several URL blacklists exists that flag a particular URL as suspicious, no such service exists for identifying and labelling a phone number. URL blacklists are effective: a lot of spam emails don't appear in our Inbox folder and directly enter the Spam folder, so the victims are at lesser risk. Whereas, due to absence of phone blacklists, all inbound / outbound calls are answered which makes victims vulnerable to attacks.

1.2 Thesis Statement

In purview of these challenges, this thesis aims to develop a *robust* Phone Reputation Service for spammer identification on social channels. Limiting the scope of this thesis to attributes that are publicly accessible via the APIs, the thesis statement is:

Phone-based spam campaigns can be disintegrated across social channels by identifying and blocking spammers using relational similarity that thrives on identifiable and discriminative public attributes.

1.3 Thesis Contribution

- **Understanding the threat landscape of phone based attacks on OTT messaging applications:** We first focus our attention on understanding various ways in which spammers can attack OTT messaging application users by leveraging information from OSNs. To understand the effectiveness of such attacks, we do an extensive online crowdsourced study to identify highly impactful phone based attack. Further, we list down the factors that govern why a user falls to phone based attack on OTT messaging applications. Our analysis revealed that social phishing attacks are most successful to lure victims. In addition, victims are deficit in regulating OTT messaging applications' usage, hence vulnerable to attacks.

- **Identifying and characterizing the threat landscape of phone based attacks on Online Social Networks:** We identify and characterize the spam campaigns active on OSNs. We create ground truth for spam campaigns that operate in different parts of the world like Indonesia, UAE, USA, India, etc. By examining campaigns running across multiple OSNs, we discover that Twitter detects and suspends $\sim 93\%$ more accounts than Facebook. Therefore, sharing intelligence about abuse-related user accounts across OSNs can aid in better spam detection. According to our 6 months dataset, around $\sim 35K$ victims and $\sim 8.8M$ USD could have been saved if intelligence was shared across the OSNs. In addition, we analyse the modus of operandi of several campaigns to understand the monetization model of spammers behind such attacks. Finally, we compare the characteristic behavioral difference between the spam and legitimate phone based campaigns.

- **Investigating the effectiveness of state-of-the-art techniques for detecting spammers:** We further look at the effectiveness of caller ID applications that identify an incoming phone call as spam. These applications are vulnerable to fake registration and spoofing attacks which make them inefficient in correctly identifying spammers. Further, we explore that supervised machine learning models to identify spammers are prone to manipulation, therefore, not a reliable solution to identify spammers.

- **Building a robust phone reputation score for phone numbers on OSNs:** Building a robust phone reputation score for phone numbers on OSNs: To build a robust solution to uncover spammers, we model OSNs as a heterogeneous network by leveraging various interconnections between different types of nodes present in the dataset. In particular, we make the following contributions – (1) We propose a simple yet effective metric, called Hierarchical Meta-Path Score (HMPS) to measure the proximity of an unknown user to the other known pool of spammers, (2) We design a feedback-based active learning strategy and show that it significantly outperforms three state-of-the-art baselines for the task of spam detection. Our method achieves 6.9% and 67.3% higher F1-score and AUC, respectively compared to the

best baseline method, (3) To overcome the problem of less training instances for supervised learning, we show that our proposed feedback strategy achieves 25.6% and 46% higher F1-score and AUC respectively than other oversampling strategies. Finally, we perform a case study to show how our method is capable of detecting those users as spammers who have not been suspended by Twitter (and other baselines) yet. We finally use spammer metrics to design a phone reputation service (called SpamDoctor [7]) that can flag a potential bad phone number. The strength of this method is that it doesn't get manipulated by temporal signals.

1.3.1 Thesis Roadmap

The rest of the thesis is organized as follows. Chapter 2 discusses the literature review in the space of exploring the spam campaigns that abuse URLs and phone numbers on different mediums. We also discuss several techniques implemented by researchers to thwart such attacks. Chapter 3 describes our work on understanding the threat landscape of phone number abuse on OTT messaging applications, in addition to studying factors responsible in victims falling for phishing on such mediums. Chapter 4 contains our work on identification, characterization, and analysis of spam campaigns that abuse phone numbers across several social networks like Twitter, Facebook, GooglePlus, etc. Chapter 5 describes our work on studying the effectiveness of state-of-the-art tools (applications) to deal with phone based campaigns and how vulnerable such techniques are to manipulation. Chapter 6 entails our work on developing a phone reputation system which flags potential bad phone numbers by assimilating intelligence from Online Social Networks. We conclude our work and discuss the limitations, implications, and future directions in Chapter 8.

1.3.2 Acronyms

OTT : Over The Top
OSN : Online Social Networks
SNS : Social Networking Services
VoIP : Voice over IP
Social channels : OSN and OTT messaging applications
HMPS : Hierarchical Meta-Path Score
HIN : Heterogeneous Information Networks
TLD : Top Level Domain
IM : Instant Messaging
CDR : Call Data Record
DNC : Do Not Call

[7] `http://labs.precog.iiitd.edu.in/phonespam/`

SEO : Search Engine Optimisation
IPC : Inbound Phone Communication
OPC : Outbound Phone Communication
FTC : Federal Trade Commission
WA : WhatsApp
TC : Truecaller
FB : Facebook
MTurk : Amazon Mechanical Turk
DNC : Do Not Call
WOT : Web Of Trust
IRS : Internet Revenue Service
LCA : Least Common Ancestor
OCC : One Class Classification

Chapter 2

Background and Literature Review

Telephony scams have been going on for many years and scammers keep robbing innocent people sadly because their success ratio is still worth their time and effort. In this section, we briefly outline some of the prior research related to launching targeted attacks on Online Social Networks (OSNs), abusing address book syncing feature of smartphones for user profiling, and understanding the rationale behind people falling for phishing attacks. We also review the literature that looks at telephony attacks that abuse phone numbers. The aim of this chapter is to look at a range of research attempts which would help to explore the various types of phone and non-phone based spam that has surfaced OSNs. Then, we look at the various limitations that a phone number poses, which makes phone-based spam identification and mitigation a hard problem. Towards the end, we discuss the implications and research gaps in identifying and analyzing phone based spam attacks. [1]

2.1 User Profiling and Abusing Address Book Syncing in OTT Messaging Applications

Literature shows that collection of user profiles can be automated and yields a lot of personal information like phone numbers, display names, and profile picture [63, 108]. Bilge et al. launched automated identity theft attacks via profiling users on SNS (Social Networking Services) by employing friend relationship with the victims [54]. They showed that people tend to accept friend requests from strangers on social networks. In [50], Authors presented experiments conducted on "social phishing". They crawled social networking sites to obtain publicly available information about users and manually crafted phishing e-mails containing certain information about them. This study showed that victims are more likely to fall for phishing attempts if some information about their friends or about themselves is included in the phishing e-mail. Jagatic et al. showed that Internet

[1] Spam and scam are used interchangeably in this thesis.

users might be over *four times* more likely to become victims if the sender is an acquaintance [102]. This was done by harvesting freely available acquaintance data of a group of Indiana University students, by crawling social networking websites like Facebook, LinkedIn, MySpace, Orkut, etc. Gupta et al. showed that inference attacks can be employed to harvest real interests of people and subsequently break mechanisms that use such personal information for user authentication [91]. Huber et al. presented friend-in-the-middle-attack on Facebook which could leverage social information about users in an automated fashion [97]. They further pointed out the possibility of context-aware spam and social phishing attacks, where attacks were found to be cheap in terms of cost and hardware. Kurowski showed a manual attack on WhatsApp to retrieve personal information about victims and proposed the feasibility of social phishing attacks against victims [120]. Retrieving cell phone numbers from tools like Wireshark was demonstrated and manual attacks were carried out to gather more information about the victim.

Schrittwieser et al. analyzed popular messaging applications like WhatsApp, Viber, Tango etc. and evaluated their security models with a focus on authentication mechanisms [165]. The experimental results showed that major security flaws exist in most of the tested applications, allowing attackers to hijack accounts, spoof sender-IDs or enumerate subscribers. They also highlighted the enumeration and privacy-related attacks that are possible due to address book syncing feature of these applications. Antonatos et al. proposed HoneyBuddy, an active honeypot infrastructure designed to detect malicious activities in Instant Messaging applications like The Microsoft Network (MSN) [46]. [2] It automatically finds people using a particular messaging service and adds them to its contact list. Findings confirmed the ineffectiveness of existing security measures in Instant Messaging services. Cheng et al. showed the abuse of address book matching in privacy leakage [63].

2.2 Phishing Attacks after User Profiling

Many researchers have examined the statistics of suspicious URLs to understand what leads to phishing. Mc. Grath et al. performed a comparative analysis of phishing and non-phishing URLs [136]. They studied features like IP addresses, WHOIS records, geographic information, and lexical features of the URL (length, character distribution, and presence of predefined brand names) and found different lengths for phishing and non-phishing URLs, misuse of free hosting services by phishers. Similar features were used by Guan et al. to classify URLs that appeared in Instant Messaging (IM) [89]. Ma et al. built a URL classification system that processed a live feed of labelled URLs and collected features (lexical, WHOIS features) for these URLs in real time [131] with an accuracy of 99%. Zhang et al. built CANTINA, a tool which classified phishing URLs by analysing the content of the webpage [196]. They assigned a weighted sum to 8 features (4 content-related, 3 lexical, and 1 WHOIS-related) to build the classifier. Among lexical features, they looked at dots in

[2] `https://www.techopedia.com/definition/8343/microsoft-network-msn`

the URL, presence of certain characters, presence of IP address in the URL, and age of the domain. They further developed 8 discriminatory features and proposed CANTINA+ which explored HTML Document Object Model (DOM) and third party services to find phishing pages [189]. Miyamoto et al. used AdaBoost-based detection training sets to determine weights for the heuristics used in CANTINA and combined them using AdaBoost algorithm [140]. Fu et al. tried to classify phishing web pages based on visual similarity [84]. They compared potential phishing pages against actual pages and assessed visual similarities between them in terms of key regions, page layouts, and overall styles. The network characteristics of spam has been investigated by spammers. Anderson et al. focussed on the Internet infrastructure use to host phishing scams [44]. They found that large number of hosts are used to advertise Internet scams using spam campaigns, individual scams themselves are typically hosted on only one machine. Ramachandran et al. studied the network level behaviour of spammers, like IP address ranges that send out the most spam, common spamming modes, persistence of spamming hosts, and botnet spamming characteristics [154]. Casado et al. used passive measurements of packet traces captured from spam sources to estimate the bottleneck bandwidths of TCP flows from these spam sources [61]. Jung et al. studied the DNS blacklist traffic to monitor the IP addresses that were sending out spam. They also observed the activity distribution of spam source hosts [105].

Phishing Attacks on Smartphones. Felt et al. showed that web sites and mobile applications interact in a way that can be spoofed by malicious entities [80]. They performed analysis on 100 applications and 85 websites which require authentication credentials by users. They evaluated 15 phishing attack scenarios where authentication mechanism can be faked and misused. Xu et al. showed that customizable notification can allow third party applications to launch phishing attacks or spam notifications. They also proposed a notification view and logging service to aid in notifications review [190]. Xiu et al. showed that browser vulnerabilities made user prone to phishing attacks [145]. Marforio et al. demonstrated that personalized security indicators can help users locate phishing attacks on mobile platforms [132].

Why people fall for phishing? Highlighting the need for anti-phishing solutions, researchers first tried to understand why phishing works, its economic and psychological impact. Dhamija et al. gave a psychology based discussion on why people fall for phishing [73]. They analysed 200 phishing attacks and identified several reasons, ranging from pure lack of computer system knowledge, to visual deception tricks used by adversaries, due to which users fall for phishing attacks. They conducted a usability study to show that people generally don't look at the browser based cues like address bar, security indicators etc. Downs et al. explored the mental models used by the Internet users to evaluate potential phishing pages [75]. Some of their subjects used incorrect strategies to analyse potential scams, leaving them at risk. Fogg et al. studied the attributes of web pages

that make it credible [83]. They found that many features of a page's appearance enhance its perceived credibility, a fact that phishers routinely exploit. Moore et al. gave an economic model to characterize the trade-offs between advertising and malware as monetary vectors providing insights into the economic impact of phishing attacks [141]. Al-Momani et al. developed a model to classify e-mails into phishing e-mails and legitimate e-mails in online mode [41]. Spamassassin was built with a number of rules to detect features common in spam e-mail that go beyond the text of the email [191]. [3] Such text included things like the ratio of pixels occupied by text to those occupied by images in a rendered version of the e-mail, presence of certain fake headers, and the like.

Dhamija et al. provided the first empirical evidence of strategies which can be implemented to deceive people [74]. They found that visual deception and browser attacks can help in carrying out phishing attacks. Sheng et al. conducted an online role-based survey with 1,001 participants to study susceptibility to fall for phishing [167]. They demonstrated that females are more susceptible to phishing attacks than males, and participants between the age group 18 and 25 are relatively more susceptible. Downs et al. interviewed 20 non-expert computer users to understand their behavior in response to receiving suspicious e-mails [76]. They found that although users know about phishing, it does not reduce the success to phish them. Kumaraguru et al. conducted a study of 5,182 Internet users measuring the effectiveness of Anti-Phishing Phil, an interactive game that teaches users not to fall for phishing [119]. They found that the ability to distinguish between legitimate and phishing websites in males was higher than females. Lee calculated the odds ratio for people falling to phishing on Symantec's e-mail scanning service [124]. The results indicated that users with subjects "Social studies", and "Eastern, Asiatic, African, American, and Australasian Languages, Literature and related subjects" were positively correlated with targeted attacks with more than 95% confidence. Kumaraguru et al. showed that users with higher Cognitive Reflection Test (CRT) scores are more likely to click on links in phishing e-mails from unknown companies, than users with lower CRT scores [118].

While past research did not focus on some of the potentially confounding factors related to participants themselves, Vishwanath focused on Facebook habits and its determinants which influence individual susceptibility to social media phishing attacks [183]. Authors demonstrated that habitual Facebook use is the most substantial predictor of victimization in social media attacks. Rashtian et al. investigated human behavior in accepting friend requests from strangers on Facebook-like online social media [155]. With interviews and online surveys, they found that factors like, "knowing the requesters in real-world", "having common hobbies or interests", "having mutual friends", and the "closeness of mutual friends" governs user's decision in accepting friend requests.

[3] http://spamassassin.apache.org/

2.3 Vishing Attacks and Spam Over Internet Telephony.

Due to low cost and scalability of Voice over Internet Protocol (VoIP) based calling systems, scammers are using the telephony channel to make millions of call and expand the vishing ecosystem. Prior work has explored the detection and ways to combat scam on VoIP. Sahin et al. highlighted several frauds related to telephony like subscription fraud, retail and wholesale billing fraud etc [160]. Authors mentioned that these frauds happen primarily due to prevalence of cheap and free calls. Griffin et al. demonstrated that vishing attacks can be carried out using VoIP [88]. They illustrated how several vishing attacks can be crafted in order to increase information security awareness. Chiappetta et al. analyzed VoIP CDRs (Call Detail Records) to build features that can classify normal or malicious users during voice communication [64]. Features were built using mutual interactions and communication patterns between the users. Past literature has also demonstrated detection of spam over VoIP through semi-supervised clustering [188], constructing multi-stage spam filter based on trust and reputation of callers [70], comparing human communication patterns with hidden Turing tests to detect botnets [152], building a system using features like call duration, social networks, and global reputation [47], proposing protection model based on user-profile framework such as users' habits [164], placing telephone honeypots to collect intelligence about telephony attacks [90, 93], and using call duration and traffic rate [109]. Caller ID spoofing is being used by scammers to hide their real identity and make fraudulent calls. Researchers have implemented various solutions to detect caller ID spoofing, using covert channels built on timing estimation and call status for verification [142], identifying the caller by tracing the calls to the corresponding SIP-ISUP interworking gateway [170], to depict if the display name in call is spoofed or not [170], using customer's phonebook feature for storing white and black lists for filtering unwanted voice calls [57], and detecting audio codecs in call path, calculating packet loss and noise profiles to determine source and path of the call [48].

2.4 Combating Spam using Automated Techniques

Spam is a growing problem for OSNs, and several researchers have looked at different ways to combat it.

Combating non-phone number based spam. There has been some contemporary work that reports the existence of spam on several OSNs like YouTube [52], Twitter [87], and Facebook [85]. On Twitter [87], authors studied the characteristics of accounts that send spam originating from previously compromised accounts. They developed techniques to identify spammer accounts from compromised accounts. On Facebook, authors found that 6% of posts were malicious and posted by compromised accounts. Thomas et al. studied the characteristics of suspended accounts on

Twitter [179]. With an in-depth analysis of several spam campaigns, they reported that 77% spam accounts suspended by Twitter were taken down on the day of their first tweet. Apart from this, there has been work done to differentiate a spammer from a non-spammer [43, 51, 123, 185, 192]. Lumezanu et al. studied the spread of URL campaigns on email and Twitter and found that spam domains receive better coverage when they appear both on Twitter and email [130]. In addition to characterizing URL-based spam, efforts have been made in detecting [67, 122, 186] and preventing [79, 153] URL-based spam campaigns.

Blum et al. proposed a method to detect phishing URLs based on SVM [55]. They used 23 features to train the SVM based on protocol, domain, and path features of the URL. They achieved an accuracy of 99%. Fette et al. used machine learning to classify phishing messages [82]. They used the properties of URLs present in the message (e.g., the number of URLs, number of domains, and number of dots in a URL) and could identify suspicious URLs with 96% accuracy. Bergholz et al. further improved the accuracy of Fette et al. by introducing models of text classification to analyse e-mail content [53]. They trained the e-mail features using Dynamic Markov Chains and Class - Topic models. Whittaker et al. analysed URL and contents of the page to determine whether a page is phishing or not [187]. They used features like presence of IP address, string characteristics of the URL and could classify more than 90% phishing pages. Kolari et al. used URLs found within a blog page as features to determine whether the page is spam with good accuracy [111]. Kirda et al. developed a browser extension AntiPhish, that aimed to protect users against spoofed website-based attacks [110]. Several other toolbars like SpoofGaurd [65], TrustBar [96], PhishZoo [40], Netcraft [33], and SiteAdvisor [135] were developed to warn users about phishing attacks. Dhamija et al. developed “trusted paths" for the Mozilla web browser that were designed to assist users in verifying that their browser has made a secure connection to a trusted site [72].

Another approach has been to educate and train users about phishing. Kumaraguru et al. used online training materials to teach people how to protect themselves from phishing attacks [116]. Robila et al. educated users using phishing IQ tests and class discussions [158]. They displayed legitimate and fraudulent e-mails to users and had them identify the phishing attempts from authentic e-mails. It helped users in knowing what to look in the e-mails. Jagatic et al. developed a contextual training approach in which users sent phishing e-mails to probe their vulnerability [103]. At the end of the study, users were typically given additional materials informing them about phishing attacks in general. This approach had been used at Indiana University in studies conducted on students about contextual attacks making use of personal information.

Researchers have also looked at meta-path based approaches to detect spam. Sun et al. [176] first proposed the idea of meta-path in heterogeneous network. Since then, it has been used extensively in various applications such as classification [112, 126], clustering [177], and similarity measures [169, 176]. Sun et al. proposed a measure called “PathSim” which outperformed Path Constrained Random Walk (PCRW) proposed by [121]. It takes into account the nodes that are not closely

connected, but also share similar visibility with each other. Meng et al. introduced biased constraint random walk to handle both symmetric and non-symmetric meta-paths [137]. Since finding all meta-paths is an NP-hard problem, they proposed Forward Stagewise Path Generation algorithm (or FSPG), which derives meta-paths that best predict the similarity between a node pair. Shi et al. proposed "HeteSim" to measure the relevance of any node pair in a meta-path [168]. To overcome the computational and memory complexity of HeteSim, Meng et al. proposed "AvgSim" that measures similarity score through two random walk processes along the given meta-path and the reverse meta-path [138]. Besides these similarity measures, Zhang et al. found node similarity based on connections between centers in X-star network [194].

Spammer Detection. Previous literature has addressed the problem of spam and spammers on Twitter and other OSNs [56, 58, 59, 67, 69, 78, 85, 125, 127, 175, 178, 179, 192]. Benevenuto et al. used OSN based features like video, user, and social relationships to detect spammers on YouTube, with an accuracy of 98% [52]. They further considered Twitter to uncover spammers based on user and network features and could correctly identify 78.5% spammers [51]. Lee et al. used features like content, friend information, posting patterns to identify spammers from an unknown pool of Twitter spammers, achieving an accuracy of 88.98% [122]. Khan et al. segregated spammers from bloggers by finding the fraction of URLs in the tweets, average number of hashtags as the prominent ones, with a precision score of 0.70 [107]. Previous literature has also looked into identifying fake accounts on OSNs by examining characteristics of user profiles (e.g., the fraction of messages posted that contain a URL, a system called SPAMDETECTOR) [174], by learning typical behavior of an account and flagging an account as suspicious in case of deviation (a system called COMPA [77]). Another system, called BOTorNOT leverages features to understand if a Twitter account exhibits similarity to the known characteristics of social bots [71, 81], or building a system that considers differences in which legitimate and malicious messages propagate through the network [144]. Liu et al. calculated user topics with LDA, and then employed supervised learning to identify spammers based on topics of discussion [129]. Link farming in Twitter where spammers acquire a large number of follower links has been investigated by Ghosh et al. [86]. By analyzing over 40,000 spammer accounts, they discovered that a majority of farmed links comes from a small number of legitimate and highly active users. Viswanath et al. applied Principal Components Analysis (PCA) to find patterns among features extracted from spam accounts [184].

Combating phone number based spam. A large fraction of phone spam includes robocalling and spoofing, wherein spammers call the victims and trick them into giving personal or financial information [34]. Studies have shown that, in spam activities, phone numbers are more stable over time than email, and hence can be more helpful in identifying spammers [68, 101]. Christin et al. analyzed a type of scam targeting Japanese users, threatening to reveal the users' browsing history,

in case they do not give them money [66]. Authors took advantage of the phone numbers made available by spammers to cluster multiple spam campaigns. In studies mentioned above, authors relied on publicly available datasets to perform their analyses. Researchers have investigated phone number abuse by analyzing cross-application features in Over-The-Top applications [94], cross-channel SMS abuse [171], and by characterizing honeypot numbers [49, 90, 93, 134]. Pandit et al. showed that several services like Truecaller, FTC, Call Data Record (CDR) datasets are able to filter only 55% of incoming spam phone numbers [150]. These techniques, however are not appropriate for caller ID spoofing attacks and cannot determine outgoing spam phone numbers. While looking at the effectiveness of Do-Not-Call (DNC) registries, Sahin et al. highlighted that such lists are used both in a positive and negative manner [159]. On one hand, spammers are exploiting DNC lists to carry out targeted attacks, on the other hand, users who have subscribed to DNC lists receive lesser spam calls. Liu et al. worked on augmenting telephone spam blacklists using CDR datasets provided by a leading telephone provider in China [128]. They observed that calls made with a spam phone number have large volume and have similar calling destinations. Finally, using volume and destination as features, authors clustered phone numbers together which were eventually marked as part of the same spam campaign.

Gupta et al. studied Twitter campaigns that abused phone numbers for both, incoming and outgoing phone spam communication [92]. Authors developed an end-to-end system to collect tweets from Twitter streaming API, used a set of regular expression to filter out phone numbers from tweets, and finally clustered phone numbers that were a part of same campaign. Miramirkhani et al. studied the Tech Support campaign that abuse phone numbers, from the perspective of domains that were used to host malicious content [139]. Authors also interacted with spammers to understand their social engineering tactics, studied URLs and domains abused by spammers along with monetization techniques. Srinivasan extended the study to understand the abuse of tech support campaigns (TSS) on search and ad channels [172]. Authors built a search-engine-based system for discovering tech support spam campaign; could identify more than 9,000 TSS-related domains and 3,365 phone numbers operated by technical support scammers, present in both organic search results as well as ads located on search-results pages. They analysed Search Engine Optimization (SEO) techniques that allowed scammers to rank well on search engines, and the long-lived support domains which allowed TSS domains to remain hidden from search engines. As a countermeasure to fight back voice spam, Sahin et al. developed Lenny, a chatbot that initiates a conversation with the telemarketer to deal with phone spam [161]. The idea behind the computer bot is to increase the interaction time with the spammers to stall fraudsters and slow down economics of voice spam, by directly and indirectly increasing the cost of a failed telemarketing or scam call. To spend 15 minutes or more of a working time with a Lenny-like bot represents a direct cost for spammers. Authors claim that spammers will not be able to target other legitimate customers during this time; it increases the call costs until reaching a valid customer and decreases the volume of calls a single spammer can

generate in a certain time period.

2.5 Research Gaps

It is not hard to find news articles / reports detailing the current phone spam problem plaguing people across the US, UK, Canada, and in many other parts of the world. Phone numbers are an attractive target for spammers due to following reasons – (i) Phone numbers are ubiquitous due to smartphone penetration growing in all population segments of society, both rural and urban, and amongst all age groups. Therefore, spammers can expect more reachability, in terms of potential victims, while abusing phone numbers; (ii) phone numbers are not easy to fake, due to verification process associated with it. On the other hand, multiple fake e-mail addresses and online social identities can be easily created. As a result, there is a greater degree of trust associated with phone numbers as compared to other user identifiers; (iii) phone numbers are personal and persistent. People generally retain the same phone number for long time due to the cost associated with it, while there can be multiple e-mail address and online identities. Hence, spammers can ensure greater success as victims would be actively using the identifier (phone number) they wish to exploit; (iv) phone numbers have faster response time than e-mail. This time sensitivity can be leveraged by the attacker to his / her benefit. As much as using a phone number is advantageous for a spammer, developing solutions against phone spam is hard. The biggest hurdle to understanding the phone spam problem has typically been the fact that data (data of spam campaigns that abuse a phone number) is hard to collect. We enumerate several challenges in dealing with phone spam:

- **Outgoing spam communication harder to detect than incoming spam communication:** In order to get a true understanding of the problem, one would need to be able to scrape everyone's call history and match it against databases like those available from Truecaller. Telecom providers probably already has such a capability, but there are obvious privacy and security risks and liabilities to that. Some spam phone numbers lists generated by websites like Malwarebytes are specific to certain campaigns, like the popular Tech Support campaign that has been going on for a while [36]. Over time there were a total of 1,705 phone numbers listed on the Malwarebytes list. On comparing the numbers with our dataset, we found that only 20.3% from our TSS dataset were also present in the telephony blacklist provided by Malwarebytes. This finding reinforces the previous one in suggesting that phone numbers used in cross-channel abuse are going largely undetected by existing telephony channel blacklists. We also checked popular call-blocking applications, such as TrueCaller and Mr. Number and found that less than 10% phone numbers from our lists are present in them. The low coverage in such applications / services could be because – (i) the intelligence from the online channel is not getting integrated into these mobile application blacklists, like in the

case of 800notes.com and Malwarebytes; (ii) call-blocking mobile applications are geared more towards blocking in-bound phone communication (IPC), such as unverified robocalls, rather than blocking outbound phone communication (OPC) which is what happens when the victim is coerced into calling a phone number under the control of a spammer.

- **Challenges with Phone carriers and Service Providers:** From the perspective of telecom services, telecommunications is highly regulated by the government to create competition and fairness, but this also slows down innovation and reduces the risks carriers are willing to take. Placing phone calls is inexpensive for telemarketers and spammers; they can make millions of illegal calls at little cost and with almost no risk. Unfortunately, phone companies make money when they connect these calls to victims' phone; they will lose good share of fortune by building anti-phone spam solutions.

- **Caller ID Spoofing:** Spammers are leveraging the enormous inventory of inactive phones to simulate the illusion that incoming calls are originating from your local neighborhood. Spoofing is easy to implement, and blocking those number, or developing and applying a simple rule is unworkable because the phone number inventory dramatically outstrips even the most ambitious blocking. This facilitates outnumbered spoofed calls made by telemarketings while conceding their identity well.

Is there a solution to phone spam? There are multiple ways a user can employ to help mitigate the phone spam issue, but there's no solution on the market (yet) that will completely stop calls from coming to one's phone. This thesis aims to build an effective, robust solution against phone number. Here, we highlight some of the techniques that can be used:

- **Changing the phone number:** A crude way to avoid phone spam is changing the existing phone number in use. However, it could be possible that one is inheriting someone else's phone spam when getting a new phone number. Phone companies tend to recycle numbers; person X's "new" number might be person Y's "old" number. If that number has been subjected to any kind of breach, X is essentially getting transferred to a new kind of spam.

- **Installing a call blocker:** It won't stop the calls but can minimise the frequency. However, as mentioned above, these applications can stop incoming calls but are not effective for outgoing spam communication. The coverage of spam phone numbers in such applications is bleak.

- **Joining the Do Not Call Registry:** This helps in mitigate a very small percentage of the phone spam, but the DNCR only helps mitigate sales calls. Spammers also abuse the personal information given by users while registering to such lists. Further, businesses can easily get around the Registry rules as well. For example, non-profits can still make donation calls or

calls that are in effect sales calls even if they masquerading as though they're for non-profit purposes.

- **Reporting suspected spam numbers:** Crowdsourcing is a potential way for blacklisting services to aggregate data about potential bad phone numbers. Federal Trade Commission (FTC) has a page for reporting spam and scam caller, but, the FTC has a very hard time stopping spoofed numbers, which seem to be comprising the majority of spam calls these days.

None of the techniques mentioned above are full proof in developing a concrete solution against phone spam. This thesis aims to solve this problem by deriving intelligence from the web, specifically from Online Social Networks.

Chapter 3

Vulnerability to Phone-based Attacks

In this chapter, we highlight the feasibility, scalability, and success of phone-based attacks on social channels using publicly available information. We show how non-targeted, spear, and social phishing attacks can be crafted against social channels (WhatsApp) users by exploiting cross-application features from multiple applications (Truecaller and Facebook). We present the success of these attacks by evaluating users' response using an online roleplay study with Amazon Mechanical Turk participants. First we discuss the advantage of picking WhatsApp as the attack vector by spammers followed by the automated attack simulation, and the final attack results. Although similar results were found for other mediums like e-mail, we demonstrate that due to the significantly increased user engagement via social channels and ease with which phone numbers allow collection of pertinent information, there is a clear need for better protection of this medium.

3.1 WhatsApp as a Vulnerable Attack Target

> Dear User,
>
> Your incoming mails are pending due to the recent upgrade to our database, In order to receive this messages. Please <badlink> click here <badlink> to submit the form and wait for responds via Email. We apologies for any inconvenience and appreciate your understanding.

Not a day passes by without receiving a similar spam or a phishing e-mail. According to Jakobbson, phishing is nothing but a form of social engineering in which an attacker attempts to fraudulently acquire sensitive information from a victim by impersonating a trustworthy third party [102]. Recently, there has been a tremendous growth of similar phishing attempts on the traditional telephony channel. New forms of phishing attacks have emerged exploiting traditional text messaging services, i.e., SMS (SMiShing [19]) and voice (vishing [25]).

The convergence of telephony and the Internet with technologies like Voice over IP (VoIP) is fueling the growth of Over-The-Top (OTT) messaging applications that allow smartphone users to communicate with each other in myriad ways. OTT messaging applications (like WhatsApp, Viber, WeChat), [1] and VoIP applications (like Skype, Google Hangouts) [2] are used by millions of users around the globe. In fact, the volume of messages via OTT messaging applications has overtaken traditional SMS [16] and e-mail [180].

These OTT messaging applications use phone numbers to uniquely identify users and allow them to find their friends who also use the same application. Attackers are using the same functionality and use a phone number as a unique identifier to exploit cross-application features for launching phishing attacks ranging from non-targeted to targeted (spear [167] and social [102]) and plausibly whaling attacks [27]. As a result, OTT messaging has become an attractive attack vector for spammers and malicious actors who are now abusing it for illicit activities like delivering spam and phishing messages. For example, unsolicited messages like investment advertisements, adult conversation advertisements, and random contacts requests were seen to propagate on WhatsApp, as shown in Figure 3.1. Resende et al. also recorded the spread of misinformation on WhatsApp during political campaigns where fake images are shared on this medium [157].

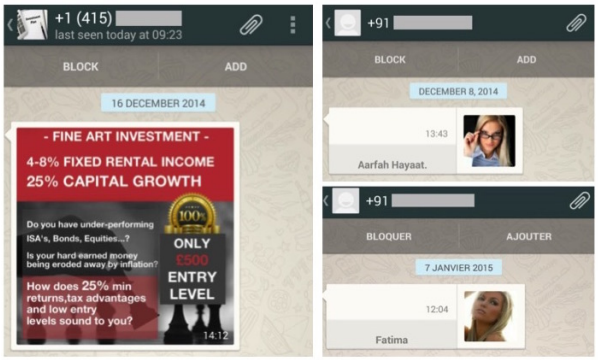


Figure 3.1: Unsolicited advertisements and contacts as phishing messages abusing WhatsApp.

OTT messaging applications use phone numbers for user authentication and communication. Authentication is typically done when a user registers with the application by providing his / her phone number and the validity of the phone number is verified by delivering an SMS message to it. A phone number is a personally identifiable piece of information with which an individual can be associated uniquely, in most cases [197]. Although there exist burner phones [3] (i.e., temporary phone number that can be provided instead of the user's actual cell number) in some countries where a phone number may not be reliably associated with a person, in an overwhelming number of

[1] http://marketingland.com/four-top-six-social-networks-actually-chat-apps-115168
[2] http://beebom.com/2015/09/best-voip-apps
[3] https://www.puretalkusa.com/blog/what-is-a-burner-phone/

cases, phone numbers are linked to a wealth of information about their owners like name and place where he / she lives. Such phone numbers are a verified part of user identity because one needs to obtain a physical SIM card and complete the verification process of a service provider to obtain a phone connection. Service providers often require personal information to setup an account.

Fraudulent communication carried out with phone numbers has resulted in loss of millions of dollars to individuals and organizations [143, 149, 173] despite the fact that phone numbers are considered private information and not easily exposed by many platforms. Attackers who want to target certain users prefer phone numbers over other identities like e-mail addresses and online social identifiers due to multiple reasons - a) Phone numbers are ubiquitous due to smartphone penetration growing in all population segments of society, both rural and urban, and amongst all age groups too [99]. Therefore, attackers expect more reachability, in terms of potential victims, while abusing phone numbers; b) In most countries, a verification process is required before someone can obtain a phone number. Because of this, attackers cannot obtain phone numbers as easily as e-mail addresses and online social identities which can be easily created and faked. As a result, there is a greater degree of trust associated with phone numbers as compared to other user identifiers; c) Phone numbers are personal and more persistent. People generally retain the same phone number for a long time due to the cost associated with it, where as, one can have multiple e-mail address and online identities. As victims would be using the same phone number, attackers can abuse it to increase the success rate of their attacks. On the other hand, due to multiplicity nature of e-mail address, the success rate for attackers to find and exploit current e-mail addresses would be low; d) Phone calls and text messages are synchronous in nature and have faster response time than e-mail. This time sensitivity can be leveraged by the attacker to his / her benefit; e) We have mature and effective defenses against e-mail spam but this is not true for phone and messaging spam. There exist services like Truecaller that warn against potential spam but are ineffective because, first, the intelligence from the online channel is not getting integrated into these mobile application blacklists, like in the 26 case of 800notes.com and Malwarebytes, and secondly, call-blocking mobile applications are geared more towards blocking inbound phone communication (IPC), such as unverified robocalls, rather than blocking outbound phone communication (OPC) which is what happens when the victim is coerced into calling a phone number under the control of a spammer. Thus, attackers have an advantage when they exploit the telephony channel.

The proliferation of messaging, voice, and other related smartphone applications result in collection and access to a wealth of information about owners of smartphones. In this chapter, we explore if malicious actors can exploit these applications to collect and aggregate information about intended victims to craft more targeted attacks. In contrast, most prior research has explored phone number abuse with either voice (vishing) / SMS spam, which aim to either collect personal information or direct users to fraud websites [88]. Although various e-mail tricks have been seen in the past, they are not well known on messaging apps.

Security implications of phone number abuse can be classified into three major categories:

- **High-impact attacks:** This includes creating emergency panic situations where people can be told about some mis-happening with relatives / dear ones. People commit denial of service attacks against phone numbers by sending bulk SMSs and paralyzing the phone for few seconds. Most cell phone plans limit the number of text messages one can send and receive. If an attacker spams with text messages, phone number's owner may be charged additional fees. Spammers can also send malicious code as a text message, which can hijack the phone, the moment it is received, similar to Stagefright vulnerability observed recently. [4]

- **Medium-impact attacks:** Spammers know enough legitimate information about an individual to make themselves seem trustworthy and deceive the victim into divulging sensitive information. They can imitate bank officials and deceive people in giving out their personal information like SSN, bank account number, credit card number etc. In addition, they can send extremely urgent-sounding text messages posing as a trusted organization, and get your information when phone number's onwer click on a link in the message.

- **Low-impact attacks:** Social engineering attacks can be carried out against phone numbers. Personalized calls can increase the success of these attacks. Spammers can extract money by making calls for charitable reasons. By exploiting the sentiments of people, they can lure them to pay handsome sums of money in the name of charity. Telemarketing fraud calls can be made to obtain personal information in the name of selling gift items or fake lottery prizes.

As phishing attacks on mobile platforms are not widespread, lack of awareness makes mobile channels more vulnerable to phishing attacks. Past work has shown that small size of the device and other hidden factors make it difficult to detect phishing sites on the mobile device [32]. The continuous usage and presence of people on smartphone devices increases the chance of looking at whatever is being received. Desktop users, on the other hand, get information only when they have access to their machines. Instant messaging applications build connections with close friends, colleagues, etc., thereby being trusted more than other web sources. All the above mentioned factors govern phone channel as potential vulnerable targets.

In this chapter, we explore how attackers can exploit phone numbers to launch targeted attacks over phone and other communication channels. We first demonstrate how a phone number can be used across multiple applications to collect private and personal information which can later be aggregated for targeted attacks. Reverse-lookup contact feature used by caller ID applications like Truecaller [5] can be exploited to find more details (e.g. name) about the owner of the phone number. Furthermore, by correlating this with *public* information present on online social networking

[4] http://fortune.com/2015/07/27/stagefright-android-vulnerability-text/
[5] http://truecaller.com/

platforms (e.g., Facebook), attackers can determine the social circle (friends) of the victim. Finally, address book syncing feature of OTT messaging applications allows attackers to determine what applications certain users are using on their smartphones. Based on this, attackers can identify the specific OTT messaging applications (e.g., WhatsApp) that can be used to reach the users. Finally, we provide early evidence of crafting whaling attacks [27] against the owners of vanity numbers [17], phone numbers generally owned by people with high influence or high-net-worth individuals.

By developing an automated and scalable system that uses phone numbers to facilitate targeted attacks to be crafted at scale, we make following contributions:

- This is the *first* attempt to systematically understand the threat posed by the ease of correlating user information across caller ID lookup application (Truecaller) and social networking application (Facebook) using phone numbers as unique identifiers. We show the attack is feasible with easily available computational resources, and poses a significant security and privacy threat.
- To carry out attacks on a large scale, the entire attack cycle from determining the attack channel to the launch of an attack should be automated. We design and implement an automated system that takes a phone number as an input and targets the victim on the attack channel.
- The attack strategy should be scalable, and our system is scalable to a large user population. This is based on the level of information that is available about the users. For 1,162,696 random pool of Indian phone numbers that we enumerated, it is possible to launch social and spear phishing attacks against 51,409 and 180,000 users respectively. Vishing attacks exploiting caller ID applications can be launched against 722,696 users. We also found 91,487 highly influential victims who can be attacked by crafting whaling attacks. This emphasizes the magnitude and significance of the attack.

Given that the telephony medium is not as well defended as e-mail, we believe that these contributions offer a promising new direction and demonstrate the urgent need for better security for such applications.

3.2 System Overview: Feasibility and Automation

In this section, we demonstrate the *feasibility* and the *ease* with which different phishing attacks can be crafted on OTT messaging applications. To *automate* the whole process, we build a system that exploits cross-application features to collect information about a user and determines the attack channel (OTT messaging applications) and phishing attack vectors (non-targeted, spear,

and social) (see Figure 3.2). Specifically, the system has three main steps. Based on a numbering plan, phone numbers are randomly generated and inserted into an address book of a smartphone. This address book is on a device that is under the control of the attacker. Once this is done, the system determines the attack channel, i.e., which OTT messaging applications can be used to send phishing messages to the victim. This is done by identifying what OTT messaging applications are running on the smartphone having a given phone number and exploiting their address book syncing feature. Once the attack channel is determined, the system fetches data from other applications to determine whether any additional information can be used to launch either a spear phishing or a social phishing attack.

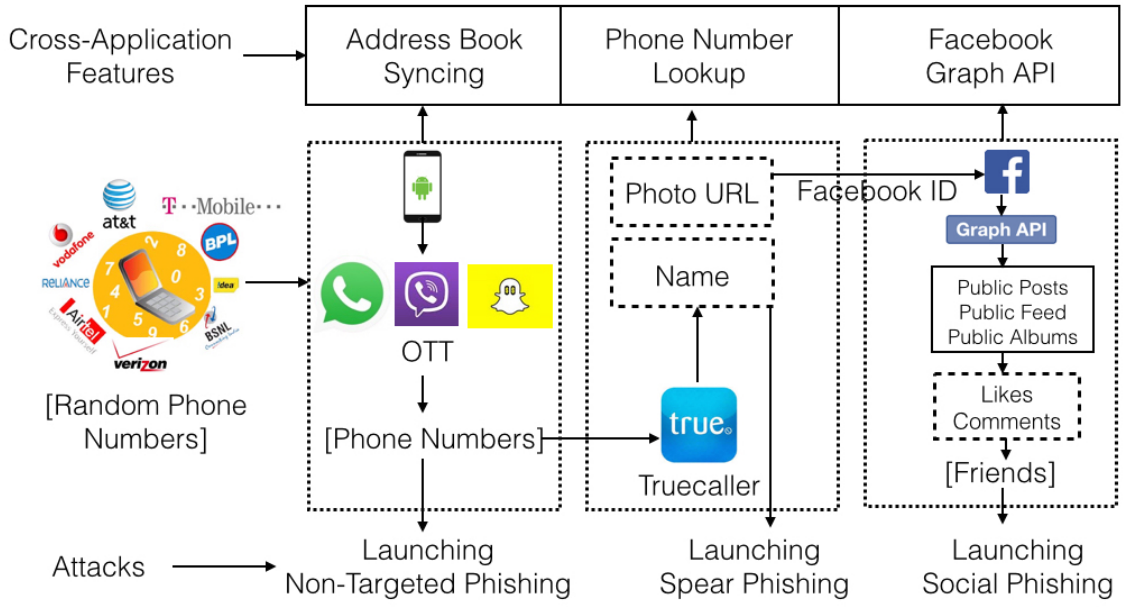


Figure 3.2: System for cross-application information gathering and attack architecture.

3.2.1 Step 1: Setting up a Forged Address Book

This section elaborates phone number generation. The system generates a large pool of phone numbers which could be exploited by an attacker to launch phishing attacks. Unlike e-mail addresses, the phone number set is finite, therefore, an entire range can be enumerated and inserted into the contact address book. This may give a few misses as some phone numbers either may not be allocated for general use or users may not be using any of these applications. However, due to finite nature of phone numbers, sufficient pool of allocated numbers can be found. To avoid getting noticed and blocked by OTT messaging applications' service providers for syncing a large address book, an attacker can devise a strategy to do it in phases [108].

3.2.2 Step 2: Identifying Attack Channel

Once phone numbers are inserted into the address book, the system determines whether the victim is present on an attack channel like WhatsApp, Viber, or Snapchat. This is achieved by exploiting the address book syncing feature in OTT messaging applications, where once a user registers himself on these applications, his contacts in the address book are uploaded (automatically, for some applications) to the OTT messaging applications' service provider and are matched against the users of the application to find already existing contacts. Only the information for numbers present in the address book is retrieved. These applications make no suggestions / recommendations for people who might be using these OTT messaging applications, like Facebook, Twitter etc. While this makes it easy and convenient for users to discover friends on these applications rather than adding them manually, it poses a security threat as well, as an attacker can use this to find the presence or absence of the victim on these applications (i.e., the attack channel).

3.2.3 Step 3: Collecting Information for an Attack Vector

Once a victim's presence is established on the attack channel, the system next determines which attack vector (non-targeted, spear, or social) can be used to target the victim. Both targeted, and non-targeted phishing attacks can be crafted against victims on OTT messaging applications. We describe the attack vector generation details for each of the attacks below.

Launching Non-targeted Phishing Attacks

Non-targeted phishing attacks are undirected attacks which are aimed to target as many users as possible. The goal is to reach out to a large audience and not to target a particular individual. Since it only requires the knowledge whether the victim is present on the channel, this can be achieved by crafting a non-targeted phishing message and sending it to the victim.

Launching Spear Phishing Attacks

Spear phishing attacks are directed at specific individuals or companies. These attacks are crafted using some a-priori knowledge of either victim's name, location, or interests to make it more believable and increase the likelihood of its success. We focus on generating spear phishing attack vectors using victim's name. To obtain information about the victim, we used Truecaller, an application that enables searching contact information using a phone number [21]. Its legitimate use is to identify incoming callers and block unwanted calls. It is a global collaborative phone directory that keeps data of more than 1 billion people around the globe. We used Truecaller as an example, but any such application can be used to determine this information. Truecaller also maintains data

from social networking sites and correlates this information to create a large dataset for people who register on it. Also, due to its address book syncing feature, it retrieves information about contacts (friends) of the "owner of the phone number" who installed it too. The 'search' endpoint of Truecaller application provides details of an individual like:

> *name, address, phone number, country, Twitter ID, e-mail, Facebook ID, Twitter photo URL,* and *photo URL*

However, the private information obtained is according to the privacy settings of users.

We automated the whole process of fetching information about phone numbers from Truecaller. We used the search end-point (used to search information about a random phone number) to obtain the registration ID corresponding to a particular phone number. [6] This was necessary to make authenticated requests and retrieve the information from their servers. We extracted the registration ID from the network packet sent while searching a random phone number on Truecaller application installed on our iPhone as shown in Figure 3.3. Once the registration ID was obtained, we programmatically fetched information for phone numbers in our dataset. Multiple instances of the process were initiated, on a 2.5 GHz Intel i5 processor, 4GB RAM at the rate of 3000 requests / min. We worked with only one registration ID for not abusing the Truecaller servers and effecting its services, however, it is easy for an attacker to scale the process by collecting multiple registration IDs to bypass rate limits imposed by Truecaller.

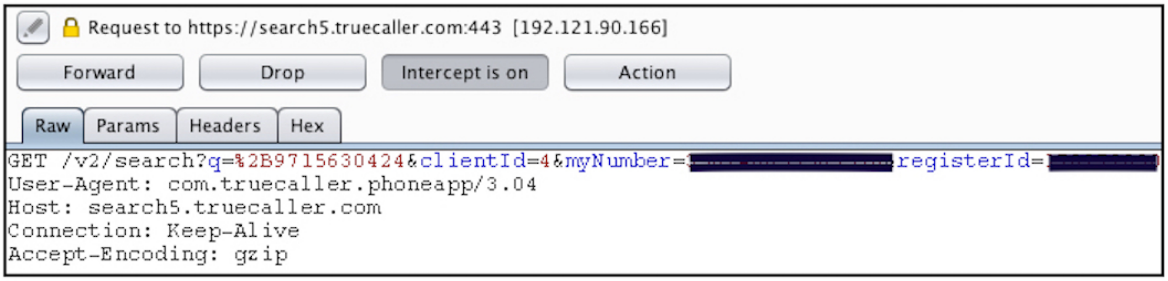


Figure 3.3: Screenshot of a network packet which is used to obtain the registration ID from Truecaller to fetch information from its servers.

Launching Social Phishing Attacks

Although phishing is a social engineering attack, here we discuss social phishing [102], i.e., how phishing attacks can be better targeted by making them appear to be coming from a friend. Friends' information can be conveniently chosen to gain trust, therefore, the attacker uses victim's name and one of his friend's information (i.e., friend's name) to craft the attack vector.

[6] We used this phone number only for research purposes and nothing else.

We use Facebook, the largest social network of friends and family to obtain the social circle of the victim [1]. We assume that friends obtained will be related to the person in some way or the other which can increase the probability of success of a social phishing attack. However, we do not differentiate between the affinity of a friend Alice with the victim as compared to another friend Charlie. Though, there may be greater affinity with one friend as compared to the other in the real-world, however, in this chapter, we treat all friends equal and leave affinity determination as future work. Truecaller aggregates data from various social networking websites and sometimes provides a link to the public profile image of the victim on Facebook. We extracted Facebook ID from these links to retrieve friends of the victim on Facebook.

Extracting friends from victim's profile is a non-trivial task, since everyone do not have their friendlist set as public. Therefore, we decided to use public sources like victim's public feed, victim's public photo albums, and victim's public posts on Facebook to obtain friends information [8], assuming users liking / commenting on any of these public sources are friends of the victim. To validate the above hypothesis, we performed a small experiment to determine if friends obtained from public sources on Facebook are a subset of public friendlist. Even though normal access token from Facebook does not provide these details, we were able to fetch the information using a never-expiring mobile OAuth token obtained from iPhone's Facebook application. [7]

We collected a random sample of 122,696 Facebook IDs and obtained 95,756 friends from public sources and 80,979 friends from public friendlist (see Figure 3.4). There were only 62,574 users for whom we were able to find friends from both public sources and public friendlist. Out of which, we found that 42,552 (68%) user-IDs liking and commenting on public sources were part of victim's friendlist with more than 95% matching rate. As observed in Figure 3.4, in some cases friends from public sources were not a complete subset of friends from public friendlist. We obtained 5,881 friends with 90 - 95% matching, 3,754 friends with 85 - 90% matching, and 10,387 friends with less than 85% matching. This could be because some users might have disabled all platform applications from accessing their data. In this case, they might not appear anywhere in any Facebook API [11]. To launch attacks using friends information, friends can be picked from public friendlist, if available, else, the attacker can rely on public sources to extract friends. Therefore, we extract the Facebook ID from the photo URL from Truecaller JSON response, and obtain public sources using Facebook Graph API to find friends on Facebook to craft social phishing attack vector. For example, following JSON object was obtained for one of the phone numbers in the dataset –


```
{
"NAME": "XXXXX",
"NUMBER": "+91XX0000000X",
"COUNTRY": "India",
```


[7] Recently, we noticed that Facebook has patched this bug and new mobile OAuth tokens do not give away this information.

```
"PHOTO_URL": "http://graph.facebook.com/XXXXXX/picture?width=320&
   height=320",
"e-mail": " "
}.
```

The Facebook ID was parsed from `PHOTO_URL` and used to make further requests. E-mail addresses for some users were also available which can be used to target them.

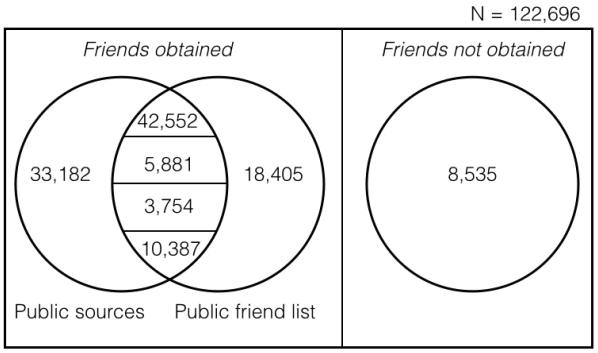


Figure 3.4: Relation between friends obtained from public sources and public friendlist. Friends from public sources are found to be a subset of friends from public friendlist in 68% cases (with more than 95% matching).

3.2.4 Scalability

To demonstrate the *scalability* of our attack, we enumerated through a list of 1,162,696 random Indian phone numbers as shown in Figure 3.5. We forge the address book of an Android device by inserting all these numbers in multiple phases. To check the presence of these numbers on an attack channel, they were synced with WhatsApp application (WA) using address book syncing feature. We found 255,873 (22%) users on WhatsApp. Numbers which were not found on WhatsApp either may not be allocated to any user or may not be registered on it. The next step was to collect attributes associated with the owner of the phone number (victim). Truecaller (TC) was used to collect more information about the victims. Detailed information for 231,409 (90.43%) users was collected using Truecaller; name was obtained for all the users. Using the name and phone number, spear phishing attacks can be crafted against these users. For rest 9.6% users whose information cannot be obtained from Truecaller, non-targeted phishing attacks can be launched against them. Finally, to craft more targeted and personalized attacks, i.e., social phishing attacks, friends information was leveraged from Facebook (FB). Social circle information was obtained for 51,409 (20.1%) users; 34,595 from public friendlist and 16,814 friends from public sources. Social phishing attacks can be launched against these users whereas spear phishing attacks can be launched against other 180,000 users whose social circle was not obtained.

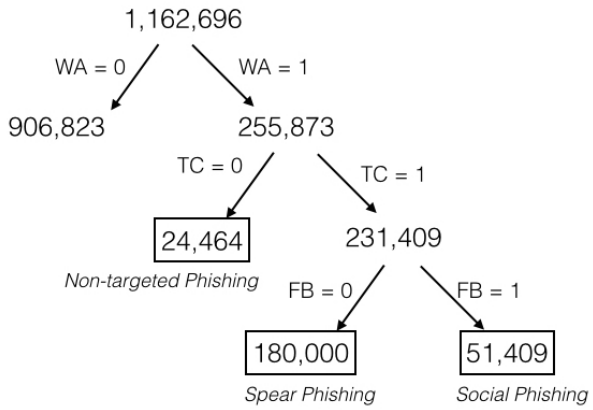


Figure 3.5: Data collection to demonstrate scalability of the system, WA–WhatsApp, TC–Truecaller, FB–Facebook. Signficant random phone numbers can be attacked using spear and social phishing.

3.2.5 Limitations in Carrying Out Large Scale Targeted Attacks

There are certain factors that can raise the difficulty bar for an attacker, and prevent large scale attacks. Applications like Truecaller and WhatsApp have incorporated certain measures to prevent large scale attacks. For instance, Truecaller limits the number of queries that can be made to search a random phone numbers at a particular time. Recently, we noticed that WhatsApp blocked the account if repeated large number of contacts are uploaded in the address book. Although an attacker can device strategies to mitigate these issues, like sending requests in batches, however it increases the computational load for him / her. WhatsApp has recently added spam reporting feature which allows a user to report a phone number as spam. [8] These nudges can help a user take an informed decision about an incoming phone number and reduce the success rate of attacks proposed in this chapter.

3.2.6 Ethical and Legal Considerations

Crawling data is an ethically sensitive area. We did the data collection just to demonstrate the feasibility and scalability of phishing attacks possible on OTT messaging applications. The goal of this work was not to collect personal information about individuals, but to explore how such applications can be abused to collect personal information. As a proof of concept, we collected information about owners of random phone numbers ensuring that the collected information is not made available to any other organization or individual. All the conducted experiments were approved by the Institutional Review Board (IRB). Data collected from the participants was anonymized and protected according to the procedures described in the corresponding IRB submission documents. At the end

[8] `http://www.ibtimes.co.uk/whatsapp-rolls-out-new-spam-blocker-feature-1497715`

of our experiments and analysis, phone numbers of all the profiles were delinked to maintain privacy. We collected only the public information available on Facebook using it's Graph API.

3.3 Other Attack Vectors

In this section, we focus on crafting more targeted whaling attacks that can be crafted on specific set of phone numbers, called vanity numbers.

3.3.1 Launching Whaling Attacks on OTT Messaging Applications and Traditional Telephony Channels

Whaling attacks [27] that are directed specifically at senior executives or other high-profile individuals within a business, government, or other organization, can be crafted on OTT messaging applications. It uses the same technique as targeted phishing (including vishing) attacks but the intended victims are people with high influence or high-net-worth individuals. In India, there is a particular set of phone numbers reserved by mobile operators for politicians, bureaucrats, and people willing to invest large amount of money to get a phone number. They are called Vanity / VIP / Fancy numbers and follow a specific pattern [23]. It could be one digit repeated several times, 99999-xxxxx or xx-8888-xxxx; two digits, xx-85-85-85-xx; or in different orders, xx-123-123-xx or xx-11-112233. The main advantage of vanity phone numbers over standard phone numbers is increased memorability. Since they are bought at higher price, owners of these phone numbers can be assumed as people with high influence [17]. For very special numbers, network providers host auctions online where people can purchase these numbers [38]. Using only vanity numbers in the address book, attackers can launch whaling attacks that only targets HNIs (High-net-worth individuals) by sending them targeted or non-targeted phishing messages. We looped through the "patterns" available from an e-auction website to enumerate vanity numbers pool. Device's address book was initialized with 171,323 vanity numbers. They were synced with WhatsApp and 40,223 (23%) were found on it. Details for 36,232 (90.1%) users were obtained using Truecaller who could be attacked via targeted phishing attacks. Rest 3,991 can be phished using non-targeted attack techniques. Furthermore, friends circle was obtained via Facebook and details for 5,756 users were found; 3,803 from public friendlist and 1,953 from public sources. These users are vulnerable to social phishing attacks, while rest 30,476 can be attacked using spear phishing attack techniques. Crafting vishing attacks that requires basic information about the phone number could be launched against 91,487 out of 171,323 vanity numbers; details about these vanity numbers was obtained from Truecaller.

Out of 11,286 vanity numbers that were found on Truecaller and Facebook; we obtained personal information (using Facebook) about owners as follows: gender (10,246), relationship status (3,733),

birthday (726), work details (6,729), school details (10,994), employer details (9,801), and hometown (6,952). E-mail address for 11,013 vanity number owners were gathered from Truecaller. We manually analyzed Facebook profiles of 100 random vanity number owners to find their occupation details and found director / CEO / chairman (10), student (10), engineer (12), consultants (2), business (5), accountant / officer (8), lecturer (5), manager (8), bank officials (12) for 70 user profiles. Detailed analysis of the content (posts and likes) generated by these users can help attackers craft better phishing and vishing attacks against them. Even though people are paying handsomely to protect their privacy, a combination of applications like Truecaller and Facebook can be used to deanonymize these profiles [18].

Apart from phishing attacks on OTT messaging applications and other attack channels mentioned above, feasibility of attacks on other channels was encountered during the course of this work. E-mail addresses for 81,389 users were obtained from Truecaller against 1,162,696 random phone numbers searched. Using Facebook Graph API, we could aggregate information like gender, birthday, hometown, work / school / education details, and relationship status for the user associated to a phone number. Using this information, attacker can choose e-mail or sms as other attack channel to launch phishing attacks.

3.4 Study Design: Vulnerability to Fall for Phishing

As mentioned previously, in this section, we only focus on our phishing attacks on OTT messaging applications. In this chapter, we do not focus on the success rate of the other attacks and channels demonstrated in previous sections.

To demonstrate the *success* of phishing attacks on OTT messaging applications, we conducted an online roleplay user study on Amazon Mechanical Turk (MTurk). We used roleplay to measure the effectiveness of phishing attacks on participants in our user study. It has been shown that such roleplay tasks have good internal and external validity [76, 119, 167]. Roleplay experiment was built as a javascript framework that enabled web interactions where participants' real actions were recorded in the database. The benefit of the roleplay is that it enables researchers to study effect of phishing attacks without conducting an actual phishing attack. Gathering data about the victims, which was required to launch attacks as proposed in this chapter, involves user deception. Deception studies are generally not preferred in the research community due to unforeseeable consequences [104]. A study that launched actual phishing attacks without informing their users reported that users got infuriated with the attacks after they were debriefed [102]. Owing to these constraints and limitations, we validated our attacks using roleplay experiment rather than a real study.

The experiment consists of three phases (see Figure 3.6), a) Briefing: to ensure that participants have concrete information and clear role description. b) The Play: to assess susceptibility to phishing

attacks. Participants were exposed to one of the three phishing attack vectors: non-targeted (e_1), spear (e_2), and social (e_3). c) Survey: to model victim's vulnerability to phishing attacks on OTT messaging applications.

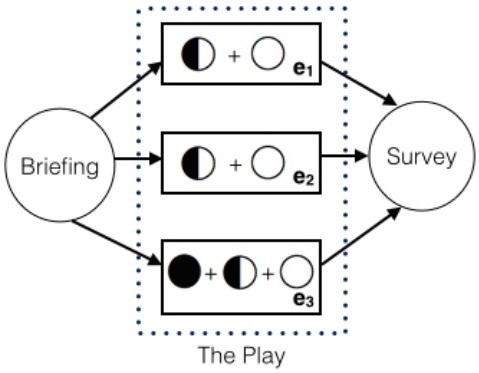


Figure 3.6: User Study Design. ◐- probably a phishing message, ●- definitely a phishing message, ○- legitimate message.

3.4.1 Recruitment

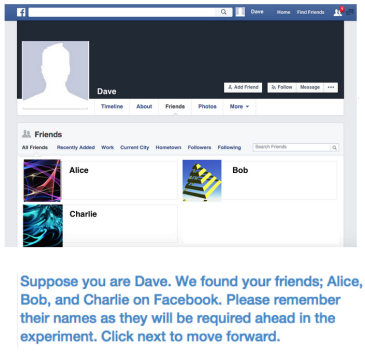
Participants were recruited through Amazon's Mechanical turk (MTurk), a crowdsourcing platform used to conduct human intelligence tasks. Participation in this study was restricted to only those who were above 18 years of age and had been using WhatsApp on regular basis. There was no restrictions on the participants due to region or country. Participants who completed all three phases were paid $0.30.

3.4.2 Briefing

Susceptibility to phishing attacks was measured with response to a roleplay task. This phase of the study was common across all the participants to familiarize them with the real-world scenario. It was conducted as explained in the following steps (see Figure 3.7).

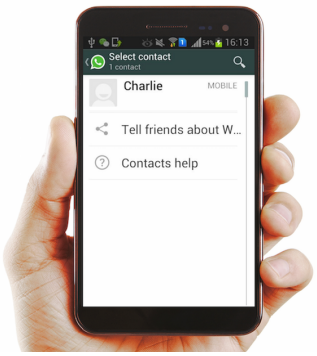
1. **Bootstrapping using Facebook (Figure 3.7(a)):** A hypothetical situation for the participant to assume himself as "Dave", and has friends "Alice", "Bob" and "Charlie". We used Facebook as a medium to bootstrap this and to let the participant familiarize with the roles. An attacker can extract this information from Facebook and create a social phishing attack vector to phish the victim, which is modeled in this step.

2. **WhatsApp registration (Figure 3.7(b)):** Next screen shows the registration on WhatsApp, as "Dave", using a random phone number. The aim is to make the participant under-



(a) Dave's Facebook profile, with Alice, Bob and Charlie as his friends.

(b) Dave's registration on WhatsApp using a random phone number.



(c) Contacts on WhatsApp with Charlie as his *only* friend.

Figure 3.7: Roles and character familiarization to participants (Briefing).

stand that the user (i.e., Dave) has an account on WhatsApp.

3. **Charlie as *the only* friend on WhatsApp (Figure 3.7(c)):** Once registered, WhatsApp syncs Dave's address book and found only Charlie. Note that other Facebook friends of Dave (i.e., Alice and Bob) were not present on WhatsApp. There could be two reasons, either Dave does not have Alice's and Bob's number in his smartphone or Alice and Bob do not have an account on WhatsApp. The primary idea is to introduce Charlie as the **only** friend who is present on WhatsApp and other friends (Alice and Bob) are not present on WhatsApp. This is to mimic the attack model when an attacker might not know which of victim's friend is present on WhatsApp. Therefore, the success of the attack should be independent of this knowledge.

4. **Elimination:** We eliminated anyone from our final results a) who provided wrong answers to the following two questions asked immediately after showing the above screens, "*Who was your friend on Facebook?*", and "*Who was your friend on WhatsApp?*" (since these screens were shown just before asking the questions, users who would be paying attention will not give a wrong answer.) b) who completed the survey in less than 30 seconds or took more than 5 minutes.

3.4.3 The Play

In this phase of the user study, participants were exposed to one of the three cases of phishing: non-targeted, spear or social. Each of the case was randomly assigned to the participants. In all three cases, the legitimate case was shown to the participant to ensure that his responses were as

expected as in a real-world scenario i.e., given a message m and a trust function T,

$$T(m)\ from\ known\ no. \geq T(m)\ from\ an\ unknown\ no. \tag{3.1}$$

The order of phishing and legitimate messages was randomized to avoid learning bias during the course of the experiment. Some participants were removed from the analysis due to unexpected behavior as discussed in Section 3.5. At the end of each WhatsApp message shown to the participant, the participant was asked the following question and corresponding options: *"What would you like to do with the message?"* with following options: click, reply, delete, or do nothing. Now we describe the three experiments to test the success of phishing attacks. Since the names and message content was kept same in all the three experiments, we do not foresee any bias. Table 3.1 summarizes all three attack vectors and scenarios.

e_1**: Testing non-targeted phishing attack's success** Non-targeted phishing is defined as an attack scenario where no additional information about the victim is known beforehand, except the phone number. In the **play phase**, participants were exposed to two scenarios in a random order to avoid learning bias; *probably phishing message* ($\mathbf{\text{◐}}_1$) and *legitimate message* (○_1) . In $\mathbf{\text{◐}}_1$, the sender is a random phone number, whereas, in ○_1, the message is from Dave's friend Charlie. The former scenario is *probable phishing* because from Dave's perspective, the sender could be one of his friends who is not present in his WhatsApp contacts. However, the latter case is *legitimate* because Charlie was already in Dave's address book, as mentioned during the briefing phase (see Section 3.4.2).

e_2**: Testing spear phishing attack's success** Spear phishing is defined as an attack where some information about the victim is known beforehand, in addition to phone number. In our experiment, name of the participant was "Dave", as described in the briefing phase. This information was used to craft a spear phishing attack vector. Similar to non-targeted phishing, participants were exposed to two scenarios; *probably phishing message* ($\mathbf{\text{◐}}_2$) and *legitimate message* (○_2). In $\mathbf{\text{◐}}_2$, the sender is a random phone number, whereas in ○_2, the message appears to be coming from the friend Charlie. However, with one notable difference, that the name of the victim (Dave) was added to the message to make it more personalized as compared to e_1.

e_3**: Testing social phishing attack's success** Social phishing is defined as an attack where social information (friends, acquaintances, colleagues, etc.) associated with the victim is gathered, in addition to known basic information about the victim (name and phone number). In this part of the experiment, participants were exposed to three scenarios (as compared to two in e_1 and e_2) in a random order; *probably phishing message* ($\mathbf{\text{◐}}_3$), *legitimate message* (○_3), and *phishing message* ($\mathbf{\text{●}}_3$). In $\mathbf{\text{◐}}_3$, the sender is a random phone number, however, mentioning the name of "Alice" (one

Table 3.1: Attack vectors with examples used in our user study.

Attack Vectors	Scenarios		Message Content	Sender Phone no.	Figure
Non-targeted	◐$_1$	Probably Phishing	Hey, Check this link out, `http://bit.ly/1JWOPhv`	Random no.	3.8(a)
	○$_1$	Legitimate	Hey, Check this link out, `http://bit.ly/1JWOPhv`	Charlie	3.8(b)
Spear	◐$_2$	Probably Phishing	Hey Dave, Check this link out, `http://bit.ly/1JWOPhv`	Random no.	3.8(c)
	○$_2$	Legitimate	Hey Dave, Check this link out, `http://bit.ly/1JWOPhv`	Charlie	3.8(d)
Social	◐$_3$	Probably Phishing	Hey Dave, Check this link out, `http://bit.ly/1JWOPhv` – Alice	Random no.	3.8(e)
	○$_3$	Legitimate	Hey Dave, Check this link out, `http://bit.ly/1JWOPhv` – Charlie	Charlie	3.8(f)
	●$_3$	Phishing	Hey Dave, Check this link out, `http://bit.ly/1JWOPhv` – Charlie	Random no.	3.8(g)

of Dave's friend on Facebook but not in the WhatsApp contact list, see Section 3.4.2). From Dave's perspective, this could probably be a legitimate message because Alice is not in Dave's address book and plausibly in real-world scenario Alice is trying to initiate a conversation with Dave. On the other hand, this could be a phishing message, because friend's name (Alice) could be forged and an attacker could imitate Alice and send a message to Dave. In ○$_3$, the message appears to be coming from the friend Charlie and in ●$_3$, the message is coming from a random phone number having Charlie as the friend's name. Since, Charlie is anyways a friend of Dave on WhatsApp, this is definitely a phishing attack because the sender phone number should have shown Charlie and not a random number.

3.4.4 Survey

Following the roleplay study, to understand the characteristics of individuals vulnerable to phishing attacks, we administered an online survey to the same set of participants. We demonstrate the necessity and the impact of the questionnaire in Section 3.5.

3.5 Results

In Section 3.2, we demonstrated the *feasibility*, *automation*, and *scalability* of phishing attacks on OTT messaging applications exploiting cross-application features. In this section, we demonstrate the *success* of phishing attacks performed during our user study.

3.5.1 Vulnerability to Fall for Phishing

In total, 460 participants completed the entire user study, out of which 129 participants were filtered out based on answers to two questions asked from the participants during the briefing phase (see Section 3.4.2). We present the results based on remaining 331 participants. We used Kruskal Wallis and Mann Whitney statistical tests to check the behavior of population subject to different order of messages show to them. We did not find statistical difference between the random groups in falling victims to all three kinds of phishing attacks.

Table 3.2 summarizes the results obtained in three experiment scenarios e_1, e_2 and e_3 as mentioned in Section 3.4.3. In this work, we define the success of a phishing attack by either a participant decided to click the link or replying to the message, whereas, attack is unsuccessful if the participant decided to delete or does nothing about the message. *Note that, these are potential participants who may fall for phishing attacks. Actual phishing attack happens when the participant goes through all the steps in the phishing attack i.e. by providing his/her sensitive details like credit card information on the phishing web page.* However, previous studies have established that a very high percentage of participants who click on the link continue to provide information to the phishing websites [117, 119, 167]. We believe that users who choose to reply to the message are *potential victims* too, as the attacker can verify active usage of the phone number. Also, attacker can lure the victim to give out personal information in subsequent messages. Extra cautious users would have preferred to either delete / do nothing with the message received. We denote,

$$\overset{\curvearrowleft}{\CIRCLE}, \overset{\curvearrowleft}{\Circle}, \overset{\curvearrowleft}{\bullet} \implies \text{clicked / replied to the message, and}$$
$$\overline{\CIRCLE}, \overline{\Circle}, \overline{\bullet} \implies \text{deleted / did nothing about the message}$$

For example, $\overset{\curvearrowleft}{\CIRCLE}$ means participant chose to click or reply to a (*probably phishing message*), while $\overline{\CIRCLE}$ means participant chose to delete or do nothing about the (*probably phishing message*). We remove those participants from our further analysis who chose to click on phishing / vulnerable message

but not on the legitimate message. Because according to equation 3.1,

$$T(\bigcirc) \geq T(\bullet) \quad and \quad T(\bigcirc) \geq T(\mathbin{\rotatebox{180}{◐}})$$

We denote these participants as *Unknown* (see Table 3.2).

We denote those participants as *Vulnerable* (i.e. falling for phishing attacks) who chose to click / reply on either phishing (●) or probably phishing messages (◐) or both. All other participants were part of *Cautious* group, i.e., who chose to delete or do nothing about both phishing (●) and probably phishing messages (◐).

We define the success rate of phishing attack as:

$$Success(\%) = \frac{Vulnerable}{Vulnerable + Cautious} * 100$$

In total, we have 314 out of 331 participants who were either vulnerable or cautious. We found that phishing attacks on OTT messaging applications were successful as, $e_1 = 34.5\%$ (37 out 107), $e_2 = 54.3\%$ (56 out of 103), and $e_3 = 69.2\%$ (72 out of 104). This is consistent with prior work that social phishing is the most effective out of the three. Furthermore, in social phishing as observed from Table 3.2, equal number of participants (63 = 54+9) fell for phishing when the name mentioned in the message text was Charlie (i.e., it is coming from a friend who is in Dave's WhatsApp contacts) and when the name was Alice (i.e., it is coming from a friend who is not in Dave's WhatsApp contacts). This shows that including friend's name in the message (irrespective of whether the friend is present or absent on WhatsApp) increases the success rate of phishing attacks. We repeated the analysis for random 25%, 50%, and 75% of the total participant population (to establish that the participant pool size is sufficient for our analysis) and found the success results to be consistent.

Table 3.2: Possible outcomes of phishing attacks for each experiment.

Case	Vulnerable	Cautious	Unknown
e_1	◐ ○ (37)	◐ ○ (24) ◐ ○ (46)	◐ ○ (7)
e_2	◐ ○ (56)	◐ ○ (19) ◐ ○ (28)	◐ ○ (4)
e_3	◐ ○ ● (54) ◐ ○ ● (9) ◐ ○ ● (9)	◐ ○ ● (12) ◐ ○ ● (20)	◐ ○ ● (2) ◐ ○ ● (1) ◐ ○ ● (3)

3.5.2 Vulnerability Factors Assessment

To understand a victim's vulnerability to phishing attacks on OTT messaging applications, we administered an online survey to the participants after completing the roleplay (see Section 5.2.1). Table 5.2 shows the demographics of survey participants.

Table 3.3: Demographics of survey participants (N = 314).

Variable	Class	Count (N = 314)
Age	18 - 24 years	78
	25 - 30 years	110
	31 - 40 years	99
	41 - 50 years	18
	51 - 60 years	7
	Above 60 years	2
Gender	Male	196
	Female	118

We designed the questions in our survey specifically for WhatsApp, however, it can be generalized for any other OTT messaging application since the way of operating and receiving messages is the same. In this section, we would like to understand and answer "What kind of people fall for phishing attacks on OTT messaging applications?". We use six measures to answer the above question as shown in Table 3.4, combining the responses from all 314 participants.

From the 10 items (questions) used in the survey, we use *factor analysis* to reduce the dimensionality of the factors measuring a victim's vulnerability to phishing attacks. Factor analysis is a statistical method used to study the dimensionality of a set of variables. Using principle component and varimax rotation, we were able to create four constructs; Technical savviness (2 items), Online privacy concern (3 items), Deficient self-regulation (2 items), and Frequency of use (2 items). To measure the internal consistency of items measuring the factors obtained, we used alpha reliability test [163]. It measures how well a set of items measure a single, one-dimensional aspect of individuals. In our regression analysis, we used factors having the value for Cronbach alpha (α) > 0.7, as it is known to be an acceptable reliability coefficient [163]. Table 3.4 provides details for each item used in the survey along with alpha reliability scores.

Age

Prior research shows that age of victims is a factor in determining the likelihood of falling for e-mail phishing [117, 167]. Authors found that participants falling in 25 - 30 years of age group are most susceptible to e-mail phishing attacks. We tested this measure with our survey participants, forming the following hypothesis:

Table 3.4: Factors predicting a victim's vulnerability to phishing attacks on OTT messaging applications. Columns with 1 - 5 represent Likert scale, μ denotes mean and σ denotes standard deviation of all the items.

Measures	Factors	Likert Scale					μ	σ	α
		1	**2**	**3**	**4**	**5**			
Technical Savviness	How long have you been using the Internet?	0	4	12	32	266	3.58	0.27	0.041
	How often do you read articles related to technology to keep yourself updated with latest trends and computer security issues?	5	65	102	86	56			
Frequency of Use	On an average, with how many friends do you chat on WhatsApp in a day?	58	86	89	53	28	2.86	0.22	0.889
	How often do you use WhatsApp to communicate with others on a given day (sending, replying to messages)?	54	69	68	63	60			
Deficient Self-Regulation	I feel comfortable in adding new contacts received on WhatsApp, even if it is coming from a random / unknown person.	73	106	61	59	15	2.44	0.06	0.746
	I feel comfortable in checking out links received on WhatsApp, even if it is coming from a random / unknown person.	99	82	56	65	12			
Online Privacy Concern	Are you concerned that strangers might know too much about you on WhatsApp?	75	83	65	66	25	2.59	0.04	0.820
	Are you concerned that a WhatsApp message you send someone may be inappropriately forwarded to others?	81	78	71	58	26			
	Are you concerned that a WhatsApp message you send might be read by someone else besides the person you sent it to?	77	89	73	51	24			

Hypothesis 1 *Younger individuals are more likely to fall for phishing attacks on WhatsApp.*

Gender

According to previous research on e-mail phishing, authors found that females are more vulnerable to phishing attacks than males [102, 167]. To test the effect of this factor on victims falling for phishing attacks on OTT messaging applications (in our case, WhatsApp), we build the following hypothesis:

Hypothesis 2 *Females are significantly more likely to fall for phishing attacks on WhatsApp.*

Technical savviness

Past research has shown that phishing education is an effective tool in reducing susceptibility to phishing attacks [167]. To understand the effect of being technical savviness of participants on victimization to phishing attacks, we build the following hypothesis: We believe our questions to measure technical savviness might not be appropriate.

Hypothesis 3 *Individuals with less Internet savviness are significantly more likely to fall for phishing attacks on WhatsApp.*

We measured the technical savviness on a 1 - 5 response scale with values ranging from *"Never"* to *"Very often"* on two factors shown in Table 3.4.

Frequency of Use

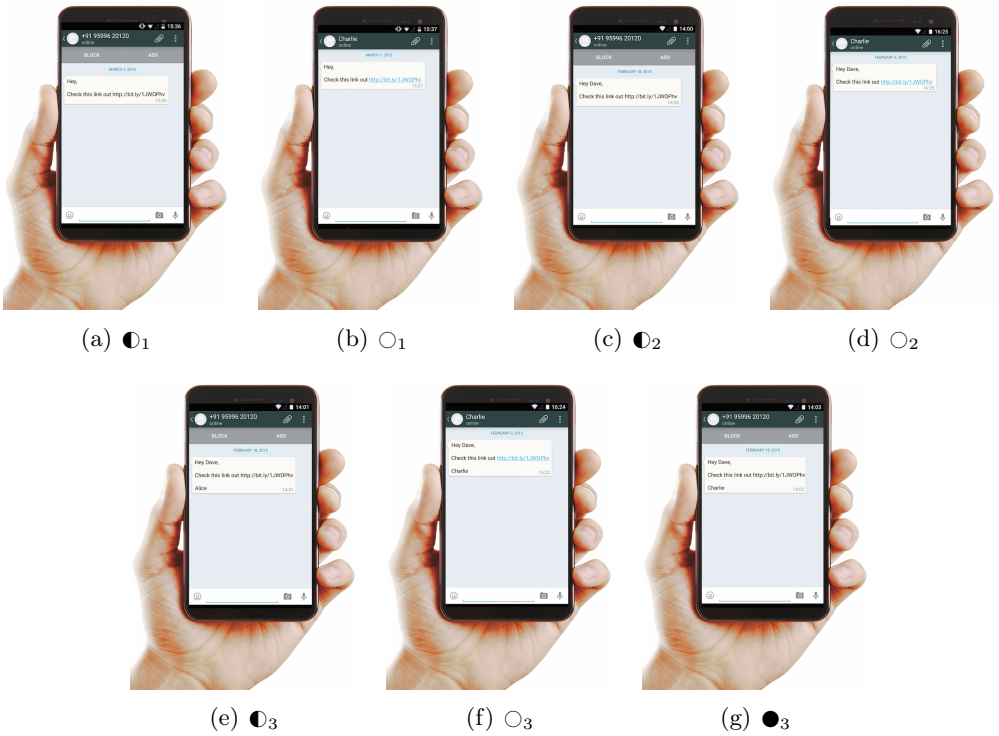
Frequency of use is defined as continuous consumption and use of an application that captures an individual's repeated and routine access, interaction, and utilization of the application. It has been shown that the frequency of using a particular medium affects behavior on that medium. For example, Vishwanath showed that people who extensively used Facebook tend to accept friend requests, even from strangers [183]. Based on this we hypothesize:

Hypothesis 4 *Individuals with higher frequency of WhatsApp use are significantly more likely to fall for phishing attacks on WhatsApp.*

We measured the frequency of use on a 1 - 5 response scale with values ranging from *"Never"* to *"Very often"* on two factors shown in Table 3.4.

Deficient Self-regulation

Unconscious and un-regulated actions often trigger habitual usage of an application. Vishwanath used the term *deficient self-regulation* to describe the state in which the individual self-control over media use is diminished resulting in a lack of awareness, attentiveness, and control over their actions [183]. For instance, individuals lacking self control might go out of the way to check their messages on WhatsApp even when they are busy, driving etc. The premise of the previous work is that individuals self-deficient in regulating their behavior on the application are more likely to fall for phishing attacks. This might be due to relaxed cognitive involvement while accessing the medium because of continuous usage. Therefore, we hypothesize:



(a) $\mathbf{\mathbb{O}}_1$ (b) $\mathbf{\mathbb{O}}_1$ (c) $\mathbf{\mathbb{O}}_2$ (d) $\mathbf{\mathbb{O}}_2$

(e) $\mathbf{\mathbb{O}}_3$ (f) $\mathbf{\mathbb{O}}_3$ (g) $\mathbf{\mathbb{O}}_3$

Figure 3.8: Three phishing attack scenarios – a) denotes random, potential spam message; b) denotes random, legitimate message; c) denotes spear, potential spam message; d) denotes spear, legitimate message; e) denotes social, potential spam message; f) denotes social, legitimate message; and g) denotes social phishing message.

Hypothesis 5 *Individuals deficient in self-regulation are significantly more likely to fall for phishing attacks on WhatsApp.*

To measure deficient self-regulation, we used a 1 - 5 response scale ranging from *"Strongly Disagree"* to *"Strongly agree"* on two factors shown in Table 3.4.

Online Privacy Concern

Online privacy concern is another condition which could influence self-victimization to phishing attacks on a medium. Online privacy concern refers to apprehensions about personal information disclosure by an individual online. Behaviorally, concern for privacy restricts access to personal information, as shown by Joinson et al. [100]. Thus, it is likely that individuals who are concerned about their privacy will restrict engaging in conversation with strangers, avoid clicking on suspicious content, and pay attention to each and every conversation initiated on WhatsApp. This leads to following hypothesis:

Hypothesis 6 *Individuals with lower levels of privacy concern online are significantly more likely to fall for phishing attacks on WhatsApp.*

To measure the effect of online privacy concern, we use a 1 - 5 response scale ranging from *"Not at all concerned"* to *"Extremely concerned"* on three factors shown in Table 3.4.

3.5.3 Binary Logistic Regression

We use binary logistic regression to model the success of phishing attack, our dependent variable (Y) which is either 0 or 1. We use age, gender, frequency of use, deficient self-regulation, and online privacy concerns as independent variables $(x_1, x_2, \ldots, x_n)$. Based on the α score (≤ 0.7), technical savviness was removed from the regression analysis (see Table 3.4).

Logistic model is defined as,

$$logit(Y) = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n \tag{3.2}$$

where $logit(Y)$ is the logistic transformation of Y and the 'β' values are the logistic coefficients that can be used to create a predictive equation [151]. where x_1, x_2, ...x_n are the set of independent variables.

Taking log on both sides,

$$\text{odds(Y)} = e^{(\beta_0+\beta_1 x_1+\ldots+\beta_n x_n)}$$

The equation shows that for 1 unit increase in one independent variable (say x_1), the odds of Y happening increases by a factor of $e^{\beta_0+\beta_1}$.

Since Y is a binary variable, we used binary logistic regression in our analysis to predict the success to fall for phishing attacks. Binary logistic regression estimates the probability that a characteristic is present (i.e., probability of "attack is successful", in our case), given the value of independent variables. We use age, gender, frequency of use, deficient self-regulation, and online privacy concerns as independent variables. X = $(X_1, X_2, ..., X_k)$ be a set of independent variables which can be discrete, continuous, or a combination and x_i is the observed value of the independent variable X_i, taking into account cumulative probabilities of all N .

Model:

$$\begin{aligned} \pi_i &= Pr(Y_i = 1|X_i = x_i) \\ &= \frac{exp(\beta_0 + \beta_i x_i)}{1 + exp(\beta_0 + \beta_i x_i)} \\ or & \\ logit(\pi_i) &= log\frac{\pi_i}{1 - \pi_i} \\ &= (\beta_0 + \beta_i x_i) \end{aligned}$$

The regression supported hypotheses 2, 4, 5, but not 1, 3, and 6. The logit model in our case is represented by the following equation:

$$\begin{aligned} Phishing_{Success} = -2.947 - 0.109\ (age) - 1.342\ (gender) + 0.337\ (frequencyofuse) \\ +0.059\ (onlineprivacyconcern) + 1.311\ (deficientself - regulation) \end{aligned}$$

Table 3.5 summarizes the results from regression analysis. The regression supported hypotheses 2, 4, 5, but not 1, 3, and 6. The values of Wald test (based on z-value) determines the significance of independent variables in predicting the dependent variable. Gender, frequency of use, and deficient self-regulation are found to be significant based on Wald's z-score [9] (significance values less than .05 as depicted in Sig. column). Exp(β) denotes the relative odds or odds ratio for a particular independent variable, given the other independent variables in the model. For example, the odd ratio of males falling for phishing attacks compared to females is 0.261. This means males are 0.26 times as likely as females to fall for phishing attacks. The constant value used in regression guarantees that the residuals don't have an overall positive or negative bias.

The overall regression was significant (χ^2 (df = 5, N= 314) = 117, p < 0.001) showing that the addition of independent variables significantly improved the model. The outliers were assessed by converting the independent variables to standardized z scores, and removing values below -3 or greater than 3 [98]. We did not find any outlier in our dataset. As none of the independent variables in the analysis have standard error $SE > 2$ (see Table 3.5), we ensured the absence of multicollinearity between independent variables.

Based on our regression model, P(model predicted success | attack was successful), also known as *sensitivity* = 122 / 158 = 0.77, while P(model predicted not successful | attack was not successful), also known as *specificity* = 113 / 156 = 0.72. Receiver Operating Characteristic (ROC) curve

[9] https://www.statisticshowto.datasciencecentral.com/wald-test/

between sensitivity and (1 - specificity) was used to assess the predictive ability (goodness-of-fit) of the regression model. The area under the curve is observed to be 0.828 with 95% confidence interval (0.784, 0.873), which is greater than 0.5, implies that logistic regression predicted the group significantly better than by random chance.

Table 3.5: Binary Logistic Regression to model victim's vulnerability.

Variables	β	SE	Wald	Sig.	$Exp(\beta)$
Age	-0.109	0.141	0.596	0.440	0.897
Gender	-1.342	0.314	18.223	0.000*	0.261
Frequency of use	0.337	0.139	5.672	0.016*	1.396
Deficient self-regulation	1.311	0.174	56.937	0.000*	3.709
Online privacy concern	0.059	0.138	0.185	0.667	1.061
Constant	-2.947	0.643	20.974	0.000	0.053

The results show that factors like users frequently using WhatsApp, and being deficient in their ability to regulate such behaviors, predict a victim's vulnerability to fall for phishing attacks on OTT messaging applications. We also found males are 0.26 times as likely as females to fall for phishing attacks on OTT messaging applications. Perhaps mobile phone platforms are a trusted source, as compared to web, and since phishing attacks have never been seen before on OTT messaging applications, users have higher trust in messages coming on these applications. Hence privacy apprehensions and concerns do not regulate their behavior on this medium.

3.5.4 Observations

To understand why people fall for phishing attacks on OTT messaging services, we took cues from the user study (play). In the play, according to the *four* options given for each phishing attack, i.e., click, reply, delete, and do nothing, we asked each participant the reason behind the selection. To find out the reason why people fall / not-fall for phishing, we will only focus on the options for 'click the message' and 'delete the message'. The complementary options for both the case were:

- Click the message: I trust the sender, I click all the links, I feel the link is interesting.
- Delete the message: I don't trust the sender, I don't trust the link in the message, I find the message as spam.

As participants cannot click on the link in the play, we assume that trust in the sender influences clicking the option, 'I feel the link is interesting'. As participants fell the message is coming from a friend, it urges them to feel it must be something interesting. The feature set now becomes {Trust

on the sender, Trust on the link}. For each option in the study, participants had to choose either of the radio buttons. Therefore, we only rank the two features from the study to govern their behavior.

Table 3.6 demonstrates value of two features in each of the phishing case scenarios.

Category	e_1 (20)	e_2 (37)	e_3 (37)
Trust the sender	17	26	24
Trust the link	3	11	2
Don't trust the sender	20	11	7
Don't trust the link	14	8	4

Table 3.6: Feature vector to depict why people fall for phishing.

As observed from the table, since e_1 (non-targeted) phishing is not successful, majority of the participants chose to not to click the messages as they did not trust the sender, followed by less trust on the links. In case of e_2 (spear) phishing, participants clicked on the incoming message as they trusted the sender. The number of participants who clicked due to the trust on links were comparatively lesser. In the third kind of phishing attack, i,e., e_3 (social phishing), participants trusted the sender more in comparison to links. In all the three cases, we found the proportion of users selecting a particular option. We found that proportion of participants clicking the message due to trust in the sender were found to be maximum in case of social phishing. The proportion of participants trusting the link in the phishing message were found to be maximum in case of spear phishing.

e_1: Non-targeted phishing As Figure dash shows, majority of people chose to click the phishing message as they trust the sender, whereas they do not click since they don't trust the links in the message. As compared to non-targeted phishing, trust on the sender is the least in this case. This means including some form of personal information about the victim increases the success rate in falling to phishing attacks.

e_2: Spear phishing As observed form Figure dash, majority of the people click on the phishing link as they trust the sender and choose not to click as they find the message as spam.

e_3: Social phishing Majority of the people clicked on the message as they trust the sender. They chose not to click as they did not trust the content of the message, i.e., either they don't trust the link, or find the message as spam.

3.5.5 Information Gain

To determine the factors that influence the phishing attack scenario, we define our feature set as {Trust on sender, Trust on link, Message content}. In each case, we assume presence of the factor as 1 and other factors as 0. For example, to calculate the information gain for trust on sender, the feature vector will look like (1, 0, 0), the other two factors viz. trust on links and message content is coded as 0. This is because if a participant chooses to click due to trust on sender, it means he does not trust the link and find the message interesting.
Let $\{c_i\}_{i=1}^{m}$ be the category in the target space. Let each factor be denoted as t, attack successful as c_1, attack unsuccessful as c_2.
Information gain is defined as:

$$G(t) = -\sum_{i=1}^{m} P_r(c_i) log P_r(c_i) + P_r(t) \sum_{i=1}^{m} P_r(\frac{c_i}{t}) log P_r(\frac{c_i}{t}) + P_r(\bar{t}) \sum_{i=1}^{m} P_r(\frac{c_i}{\bar{t}}) log P_r(\frac{c_i}{\bar{t}}) \tag{3.3}$$

e_2: Spear phishing The probability value (each feature as success, hence c_1) are calculated as follows:

$$P_r(\frac{c_1}{t}) = \frac{(14/37)*(37/100)}{(14/37)+(18/63)}$$
$$P_r(\frac{c_1}{t}) = \frac{(18/63)*(63/100)}{(14/37)+(18/63)}$$
$$P_r(\frac{c_1}{t}) = \frac{(23/37)*(37/100)}{(23/37)+(1/63)}$$
$$P_r(\frac{c_1}{\bar{t}}) = \frac{(1/63)*(63/100)}{(27/37)+(1/63)}$$

G(Trust on sender) = -0.07. Similarly, G(Trust on links) = -0.06, and G(Message content) = 0.02

3.6 Discussion

Due to inherent trust on phone numbers and the fact that spammers are moving towards abusing this unique identifier, there is a dire need to protect its abuse. Content-based filters have been created to filter legitimate and spam content on e-mails. However, due to the lack of sufficient data on OTT messaging applications, for instance, encrypted data (as in WhatsApp), it is difficult to implement similar approach. Moreover, content-based approach might not work well with phone spams due to short text in the messages and use of informal text. Also, the lack of real and public databases can compromise the evaluation of different approaches. Academic researchers do not have large corpus of good or bad phone numbers which can be used to develop a solution.

To fill this gap, there is a need to use existing infrastructure to develop solutions and filter bad phone numbers. We built a phone reputation system, called SpamDoctor that can model bad phone numbers using the intelligence from Online Social Networks, as discussed in Chapter 6. Although IP and domain reputation systems have been in existence, something similar related to phone numbers

does not exist [45, 95]. Several industry based solutions have been proposed in this direction. These services provide information about phone numbers that appear in e-mails, online complaint sites, or directly from applications (like WhitePages Pro). However, there are large number of other sources, for instance, online social networks, that have traces of campaigns related to phone numbers, which go unnoticed by these services. Services like Truecaller also label a phone number as spam, however, it is fairly easy to add noise in these crowdsourced platforms, as discussed in Chapter 5.

Specifically, our reputation system (see Figure 3.9) will help users in making a conscious decision (real-time) before responding to text or message received from that particular phone number on OTT messaging applications or online social networks. The reputation system includes a learning model which will learn features from past fraudulent numbers, and detect new phone numbers.To accomplish this task, we developed self-adaptive algorithms capable of filtering bad phone numbers, information about which can be leveraged from online social networks. This would involve identifying spam campaigns that involve phone numbers, propagating on multiple platforms, and building features to create a model to classify bad phone numbers.

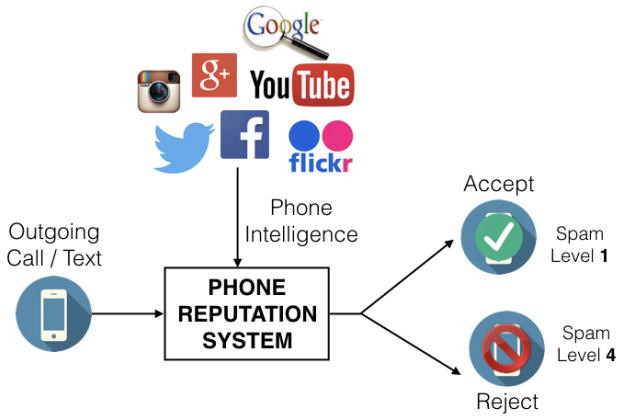


Figure 3.9: Schematic diagram of a Phone Reputation System to model bad phone numbers.

The phone intelligence derived from theses reputation systems can also be used by OTT messaging applications and caller ID applications. For instance, OTT messaging applications can do an initial filtering while registering a bad phone number. In addition to sending an SMS code to verify the number, they can incorporate additional measures to verify the owner entity. Similarly, caller ID applications can associate a spam score to the bad phone number, and display to the user making him / her alert. In addition to reputation system which can be used by OTT messaging applications and caller ID applications, we propose some recommendations on how to alleviate (if not eliminate) the security risks created by these exploits.

Recommendations to OTT Messaging Applications

Given the plausibility of phishing attacks on OTT messaging applications, this medium needs to be defended. Often users ignore the risks associated with such applications, due to ease of use, however, the platform should implement some solutions. OTT messaging applications can put certain checks; restrict address book sync feature, such that people can be added only based on requests (like Facebook), or something unique like BBM pin. In addition, we suggest that OTT messaging applications should not provide any personal information (profile picture, online status etc.) about new friends (contacts) added after automated address book syncing. Only after a sanity check, where people verify knowing each other, more information about them should be updated. Perhaps these recommendations increase user load in handling such services, trade-off between security and usability always remains a challenge.

Apart from this, end-to-end encryption (specifically in applications like WhatsApp) poses a major challenge to identify a phishing message at zero hour. Even though it makes it hard for an attacker to eavesdrop messages, WhatsApp can't build solutions to combat the problem. In order to effectively defend against phishing message, one solution could be assigning crowd-sourced score (phishing) to a phishing message. OTT messaging applications can filter messages with high phishing score. However, introducing noise in the dataset, remains a challenge, which can be overcome using phone reputation systems.

In this chapter, we demonstrated the *feasibility, automation, and scalability* of targeted attacks that can be carried out by abusing phone numbers. We presented a novel, scalable system which takes a phone number as an input, leverages information from Truecaller (to obtain victim's details) and Facebook (to obtain social circle), checks for the presence of phone number's owner on the attack channel, and finally targets the victim.

Chapter 4

Cross-Platform Intelligence On Spam Campaigns Abusing Phone Numbers Across Online Social Networks

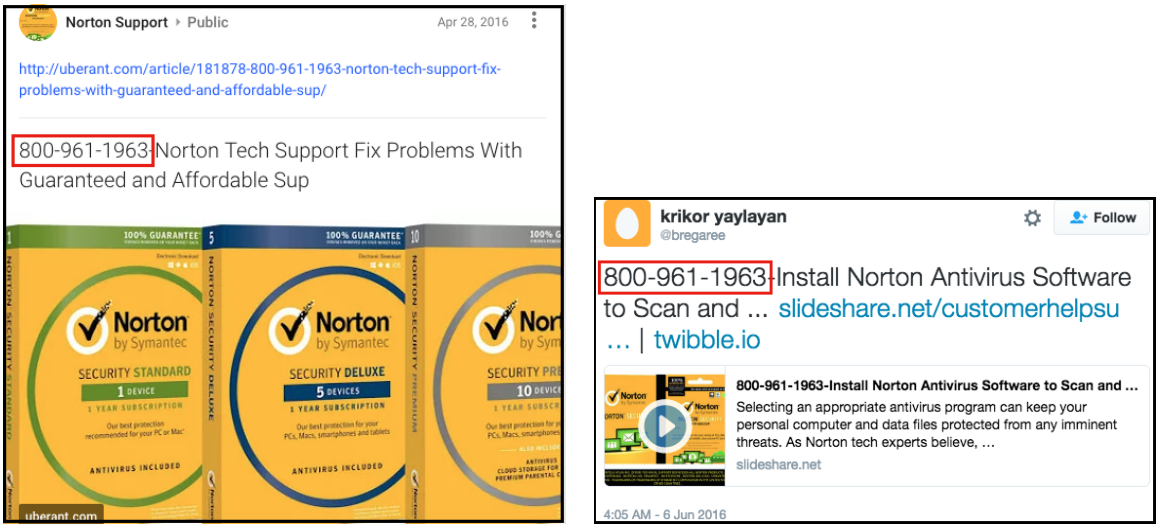
With the convergence of telephony and the Internet, the phone channel has become an attractive target for spammers to exploit and monetize spam conducted over the Internet. This chapter presents the first large-scale study of cross-platform spam campaigns that abuse phone numbers. We collect ∼22 million posts containing ∼1.9 million unique phone numbers from Twitter, Facebook, GooglePlus, Youtube, and Flickr over a period of six months. Using text clustering, we identify 202 campaigns operating across the globe with Indonesia, United States, India, and United Arab Emirates being the most prominent originators. We show that even though Indonesian campaigns generate close to 3.2 million posts, only 1.6% accounts posting them have been suspended so far. By examining campaigns running across multiple OSNs, we show that Twitter can detect and suspend 93.3% more accounts than Facebook. Therefore, sharing intelligence about abuse-related user accounts across OSNs can aid in spam detection. Using our 6 months dataset, we find that around 35,000 victims and 8.8M USD could have been saved if intelligence was shared across the OSNs. Through this data-driven view of analyzing phone number based spam campaigns running on OSNs, we highlight the unexplored phone-based attacks surfacing on OSNs.

4.1 Introduction

Increasing popularity of Online Social Networks (OSNs) has attracted a cadre of actors who craft large-scale phishing and spam campaigns targeted against OSN users. Traditionally, spammers have been driving traffic to their websites by luring users to click on URLs in their posts on

OSNs [85, 87, 179]. A significant fraction of OSN spam research has looked at solutions driven by URL blacklists [85, 178], manual classification [52], and honeypots [122, 174]. As more effective defenses are developed against malicious / spam URLs, cybercriminals are looking for other ways to engage with users. Telephony has become a cost-effective medium for such engagement, and phone numbers are now being used to drive call traffic to spammer operated resources (e.g., call centers).

In this chapter, we explore a data-driven approach to understand OSN abuse that makes use of phone numbers as action tokens in the realization / monetization phase of spam campaigns. Telephony could be an effective tool for spammers because Internet crime reports suggest that people fell victim to phone scams leading to a loss of $7.4B in 2015, and a projected loss of $11B in 2018 for Americans alone [6]. Specifically, in the phone-based abuse of OSNs, spammers advertise phone numbers under their control via OSN posts and lure OSN users into calling these numbers. Advertising phone numbers reduce spammers' overhead of finding the set of potential victims who can be targeted via the phone. After that, they try convincing the victims that their services are genuine, and deceive them into making payments after a series of interactions [139]. To maximize their reach and impact, spammers disseminate similar content across OSNs. Figure 4.1 shows the cross-posting behavior across OSNs for one such campaign found in the dataset, the Tech Support campaign.



(a) GooglePlus (b) Twitter

Figure 4.1: Presence of the Tech Support campaign across OSNs (Twitter and GooglePlus).

While URLs help spammers attract victims to websites that host malicious content, phone numbers provide more leverage to spammers. Due to the inherent trust associated with the telephony medium and as spammers interact directly with victims over calls, spammers using phone numbers stand a better chance of convincing and hence are likely to make more impact. Besides, they can use fewer phone numbers as compared to URLs; a large number of URLs are required to evade filtering

mechanisms incorporated by OSNs. [1] Moreover, the monetization and advertising channel, in this case, phone and OSN / Web respectively, are different in phone-based campaigns compared to a single channel (Web) used in URL-based campaigns. This requires correlation of abuse information across channels which makes it harder for OSN service providers to build effective solutions. Since the modus operandi in URL-based and phone-based spam campaigns is different, leaving phone-based spams unexplored can limit OSN service providers' ability to defend their users from spam. Solutions have been built to educate users about URL-based spam [115], while limited education is available for phone-based attacks. This is evident from several well publicized and long running Tech Support spam campaigns (since 2008) that use phone numbers to lure victims leading to huge financial losses in the past, as reported by the Federal Bureau of Investigation [146]. Nevertheless, to the best of our knowledge, the space of OSN abuse using phone numbers is largely unexplored.

In this chapter, we address this gap by *identifying* and *characterizing* spam campaigns that abuse phone numbers across multiple OSNs. Studying phone-based spam across multiple OSNs provides a new perspective and helps in understanding how spammers work in coordination to increase their impact. From 22M posts collected from Twitter, Facebook, GooglePlus, YouTube, and Flickr, we identify 202 campaigns are running in different countries around the world, abusing 806 unique phone numbers. Studying these campaigns, we make the following observations:

1. We find that the cross-platform spam campaigns which rely on phone numbers originate from many countries, but most of them come from Indonesia, United States of America (USA), India, and United Arab Emirates (UAE). These campaigns exploit fewer distinct phone numbers than URLs. Intuitively, this could be due to the cost associated with buying a phone number. Victims that fall prey to these campaigns are offered banned filmography, personal products and a variety of other services; but the services are not delivered even after successful payment.

2. As reported in earlier research [86], we also find evidence that suggests user accounts collude to maximize their reach either by creating multiple accounts or promoting other spammers' content. To evade suspension strategies of each OSN, spammers keep the volume per account low. Our results show that accounts are suspended after being active for 33 days (on average); while literature suggests that spammers involved in URL-based spam campaigns, on the other hand, could survive only for three days after their first post [179]. Again, this suggests a crucial need to build effective solutions to combat phone-based spam.

3. Our analysis also suggests that OSN service providers should work together in the fight against phone-based spam campaigns. Examining phone numbers involved in campaigns across OSNs, we find that although all OSNs are consistently being abused, Twitter is the most preferred

[1] https://support.twitter.com/articles/90491

OSN for propagating a phone campaign. By analyzing spammers' multiple identities across OSNs, we find that Twitter can suspend 93.3% more accounts than Facebook. Thus, quick user linking and *cross-platform intelligence* can be useful in preventing the onset and reducing the lifetime of a campaign on a particular network. We estimate that cross-platform intelligence can help protect 35,407 victims across OSNs, resulting in potential savings of $8.8M.

Our results shed light on phone-based spam campaigns where spammers are using one channel (OSN) to spread their content, and other channel (voice / SMS / message via phone) to convince their victims to fall prey to their campaigns. Given that no timely and effective filters exist on either channel to combat such spam, there is an imperative need to build one. We believe that spammers would provide real phone numbers, at which their victims can reach them. Therefore, this dataset is less polluted with fake or spoofed numbers, which makes our results reliable.

4.2 Dataset

In this section, we discuss our methodology for collecting phone numbers, posts and other metadata; later cluster them to find campaigns on OSNs. These campaigns are then heuristically tagged as benign or spam. Figure 4.2 shows our comprehensive system that is used to collect phone numbers across multiple OSNs. We picked Twitter as the starting point to find phone numbers, as it provides easier access to large amounts of data as compared to other online social networks [22, 147]. We set up a framework to collect a stream of tweets containing phone numbers. For each unique phone number received every day, a query was made to other OSNs viz. Facebook, [2] GooglePlus, Flickr, and YouTube, and for every search, we stored the following details - user details (user ID, screen name, number of followers and friends), post details (time of publication, text, URL, number of retweets, likes, shares, and reactions), and whether the ID is suspended. The data collection ran over a period of six months, between April 25, 2016 and October 26, 2016. Our system collected 22,690,601 (22M) posts containing 1,845,150 (1.8M) unique phone numbers, posted by 3,365,017 (3.3M) unique user accounts on five different OSNs. After removing noise, the filtered set was further used for finding campaigns.

Initial Seed of Phone Numbers

To study cross-platform phone number spam abuse, we picked Twitter as the starting point to find phone numbers, as it provides 1% of public stream data, which is large as compared to data available from other online social networks [22, 147]. We already had a system set up to collect a

[2]Collecting data from Facebook was challenging. In April 2015, Facebook deprecated their post-search API endpoint [3], so we used an Android mobile OAuth token to search content using the Graph API [94].

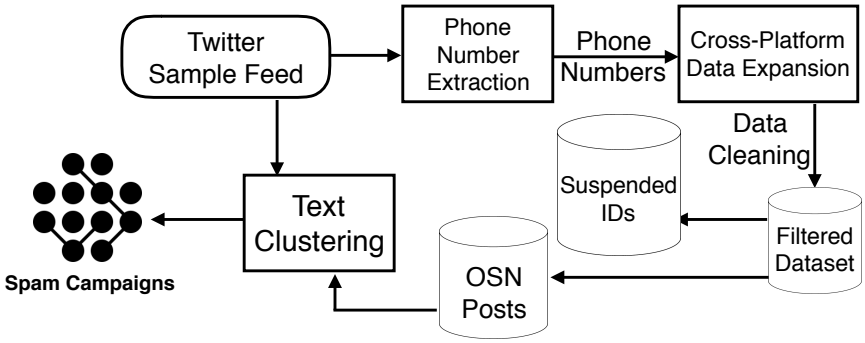


Figure 4.2: System architecture for data collection across multiple OSNs.

stream of tweets containing only phone numbers. We used a curated list of 400 keywords like call, SMS, WhatsApp, ring, contact, dial, reach, etc. to filter relevant tweets from Twitter's Streaming API. We received daily snapshots from that system, aggregated all the unique phone numbers and searched for them across other OSNs viz. Facebook, GooglePlus, Flickr, and YouTube. For every result that had a phone number, we stored the following details - user details (user ID, screen name, followers, friends), post details (published time, text, URL, retweets, likes, shares, reactions), and suspended ID information. In April 2015, Facebook deprecated their post-search end point [4], but, we used an Android mobile OAuth token to search for content on Facebook [94] using the Graph API. In addition, to ensure maximum coverage, we used Google search results to obtain data from Facebook. We refined the Google search query as '*number* site:facebook.com', since Google indexes all publicly available posts, not just a sample. This gave us links to all Facebook posts containing that particular phone number. Finally, we parsed all the user IDs from the Facebook post URL, and used Facebook's Graph API to find all the posts created by that user account. There were some posts that did not contain phone numbers, which we filtered out in the post-processing phase of the data collection, described later in this section.

Priority Queue Implementation

On an average, we received 0.5 million unique phone numbers daily from Twitter's data collection snapshot. Due to infrastructure and API limitations, it was challenging to process all 0.5 million unique phone numbers everyday. Therefore, we implemented a priority queue where each phone number was assigned a priority. Priority was assigned based on the two goals of this work: 1) getting spam campaigns which are being spread in significant volume, 2) getting recent spam campaigns. The priority of all the numbers was revised every two weeks to make sure that the quality of the dataset was maintained. We picked a two week window as Facebook Search API doesn't provide

[4] `https://developers.facebook.com/docs/graph-api/using-graph-api/v2.0#search`

data older than 14 days, as observed during experiments. To achieve the first goal (collecting voluminous campaigns), we looked at the volume generated by a phone number (p_i) normalized by the total volume generated by all n phone numbers

$$X(p_i) = \frac{V_{p_i}}{\sum_{i=0}^{n} V_{p_i}}$$

To achieve the second goal (aggregating the most recent tweets), we defined a relevance score (W) for each day (t) for a phone number (p_i) in a time period (T). For instance, for calculating the priority of a phone number in a two week window, tweets received on the 14th day (more recent) were given a higher score than tweets received on the 10th day (less recent). The weight of each day (W^t) was modeled as an exponential decay function (λ), as literature shows tweets exhibit exponential decay [162].

$$W^t = \lambda \exp^{-\lambda t}$$

The total score was calculated as summation of normalized weighted tweets received for each day. The above score was then normalized by the weighted score for all days in the time period T.

$$Y(p_i) = \frac{\sum_{t=0}^{T} (\frac{V_{p_i}^t}{V_{p_n}^t} * W^t)}{\sum_{t=0}^{T} W^t}$$

Finally the total score for a phone number (p_i) is given by the following equation:

$$\alpha_1 * X(p_i) + \alpha_2 * Y(p_i) \tag{4.1}$$

Based on experimental results, we found that $\alpha_1 = 0.9$ and $\alpha_2 = 0.1$ gave the most recent and the tweets with the highest volume.

4.2.1 Post-processing

Once the priority queue was implemented, we limit the size of the priority queue to $50k$ phone numbers. Top $50k$ unique phone numbers having the highest priority were searched across all OSNs. A phone number can have several variations, for instance, the number 1-888-551-2881 can be represented as 1(855)276-2781, 1(855) 276-2781, 1.888.551.2881, or 1 888 551 2881 (all the variations were were individually searched on each OSN to ensure completeness). When these phone numbers were searched for on multiple OSNs, we observed a lot of noise due to variations like 1(855) 276-2781, 1 888 551 2881. The space in between the digits was replaced with by arbitrary characters, thus a lot of unrelated posts were being collected. We filtered out this noise by post-processing the data, where a couple of regular expressions were used to obtain a valid phone number from the text obtained from each post are listed below:

```
Reg exp 1. ('(?<= )\d{6}-\d{3}(?= )|
(?<=\[)\d{6}-\d{3}(?=\])|(?<=\()\d{6}-\d{3}(?=\))')
Reg exp 2. (' (\d[\d ]{5,13}\d{2}) ')
Reg exp 3. ('\$ *\d+[\.]*\d+|\d+[\.]*\d+\$')
Reg exp 4. ('^\d+\s|\s\d+\s|\s\d+$')
```

Finally, after removing the noisy posts we end up having 1.4 million posts. We manually verified the richness in the final dataset obtained after post-processing. For manual inspection, we took 10,000 posts and ran our post-processing algorithm, finding valid phone numbers in the post-processed sample with negligible noise. The filtered set was further used for clustering and detecting spam clusters (campaigns).

4.2.2 Campaign Identification

A *campaign* is defined as a collection of posts made by a set of users sharing similar text and phone numbers. To make sure that we do not tag any benign campaign as spam, we filtered out the phone numbers used by even one verified account. Specifically, we consider a collection of $N = \{n_1, n_2, \ldots, n_n\}$ users across multiple OSNs, where each user n_i may post a series of time-ordered k posts, $P_{ni} = \{p_{i1}, p_{i2}, \ldots, p_{ik}\}$. A campaign P_c, thus, can be defined as $P_c = \{p_{ij}, n_i \mid n_i \in N \cap p_{ij} \in P_{ni} \cap theme(p_{ij}) \in t_k\}$ such that the campaign messages belong to a coherent theme t_k. We define themes by picking a set of common, frequently occurring words across all the posts made by all the users, along with human labelling in the end. Every phone number, say *ph1*, is represented by a set of frequent unigram tokens which occur around the phone number. All posts that contain atleast 33% tokens from the representative token set are put together in a cluster; indicating posts related to the phone number. Different phone numbers, say *ph1* and *ph2*, are put together in the same cluster if the average Jaccard coefficient between the corresponding set of posts is greater than 0.7 (we calculated different values of Jaccard coefficient and average silhouette scores to measure quality of clusters [42], and found 0.7 as knee point for corresponding value of silhouette score as 0.8). All users that post about any phone number in the clustered set (*ph1*, *ph2*) are put together. A cluster thus formed is marked as a campaign. Using this method, we found 22,390 (22K) campaigns in the dataset, collectively amounting to 10,962,350 (10.9M) posts.

Spam Campaigns: By definition, spam is an *unsolicited* message that is usually sent in *bulk*. In this chapter, we flag a campaign as *spam* if it meets the following criteria: a) the presence of phone numbers involved in the campaign in the United States Federal Trade Commission's Do Not Call (DNC) dataset [10], or b) if even one OSN account involved in the campaign is suspended. With

this, we identified 6,171 out of 22,390 campaigns as spam. However, it did not mean that others were not spamming, they just did not fit our filtering criteria. To understand the characteristics of spam campaigns, we focused only on campaigns with atleast 5000 posts. Heuristics are then followed by a thorough manual inspection to flag a campaign as spam. This results in a working dataset of *202 campaigns* comprising of *4,890,958 (4.9M) posts*; these were manually identified from 6,171 campaigns. In addition, topics were assigned to the 202 campaigns, where two campaigns could be assigned the same topic. For instance, a campaign selling shoes and other selling jackets would be assigned the topic - "Product Marketing". Since the space of exploring outgoing phone spam is new, no prior information and description about phone spam is available. To make sure that the campaigns are annotated correctly, we, being an expert in the domain, labeled and assigned a topic to the campaign. During our manual validation of spam campaigns, we found that only 3% campaigns couldn't be assigned any topic and were left uncateorized. Rest, all other campaigns could be assigned one, coherent topic. Figure 4.3 shows an example of a campaign which couldn't be assigned any topic, even though it was generating spam.

Figure 4.3: One of the spam campaign that couldn't be assigned a topic due to it's unclear nature.

4.2.3 Dataset Limitations

Our dataset collection methodology suffers from following limitations:

- **Data Sampling:** We acknowledge that our dataset may contain some bias, as we get only 1% Twitter sample available from the REST API. It can underestimate the spam campaigns observed on Twitter, however, we observed different kind of campaigns abusing phone numbers in our dataset. However, it is extremely challenging to obtain an ideal, unbiased dataset.
- **Limited Campaigns:** We started our data collection from Twitter. We might miss some

of the popular campaigns prevalent on other social networks, but, Twitter provides a good sample of public content.

- **Dataset Completeness:** We relied on the data provided by APIs from all the networks. Therefore, we might miss some posts related to the campaigns. Given that scraping is not allowed, using APIs was the only viable option to gather data. Irrespective of this, we have been able to receive quite a significant amount of posts related to phone numbers. Further, if a search parameter is not supported by the API, it is challenging to retrieve posts relevant to a phone number.

4.3 Characterizing Spam Campaigns

Having identified spam campaigns, we now characterize their propagation across OSNs. We look into how campaigns are spread in different parts of the world and how they differ in several metrics.

4.3.1 Where does phone-based spam originate?

It is important to know from which countries do the spam originate. We assume that the country associated with a phone number is the source country. For the analysis, we need to extract the country of the spam phone number. This is done either by identifying a) the language of the post containing the spam phone number via a metadata field in the tweet object, or b) by the country code using Google's phone number library. [5] These two methods helped in identifying countries for 127 campaigns. For rest of the campaigns, we called top two frequently occurring phone numbers in the campaign using Tropo [6], a VoIP software that can be used to make spoofed calls. We recorded all the calls and used Google's Speech API [7] to detect language and country of the campaign. We could identify countries for 26 more campaigns; for the remaining 49, the country is unknown. Table 4.1 presents different campaigns (column #C) run in different countries along with the number of posts (column #posts) being made in each campaign.

Popular source countries: Top four source countries selected by the volume of campaigns viz. Indonesia, United States of America (USA), India, and United Arab Emirates (UAE) show interesting characteristics. First, from Figure 4.5, we observe that there is a good overlap of campaign categories across countries, while some countries have specific categories of campaigns running. Among all the campaign categories, volume generated by Indonesian campaigns is significantly higher than any other country. While investigating further, we found that 99.3% pairs of consecutive posts related to the same campaign appeared on Twitter in less than 10 minutes. Given that a major

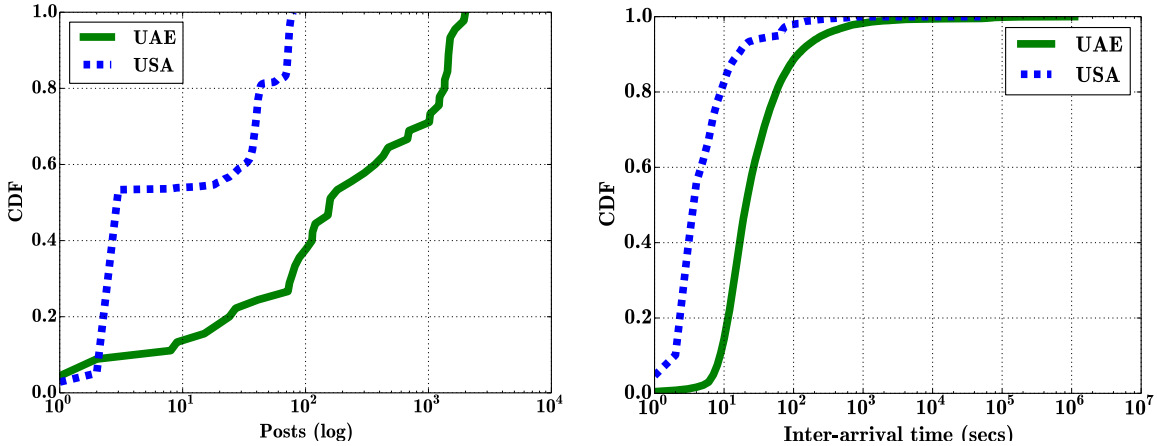
[5] https://github.com/googlei18n/libphonenumber
[6] https://www.tropo.com/
[7] https://cloud.google.com/speech/

Table 4.1: Campaigns' distribution across source countries (arranged in alphabetical order).

Country	Campaign Topics	#C	Posts
Argentina	Party Reservations	1	39,476
	Adult (porn)	1	30,751
Chile	Delivering goods	1	6,691
Columbia	Hotel Booking	1	18,228
	Adult (porn)	1	5,324
Ghana	Deception (solve marriage, depression)	1 1	12,825
Guatemala	Product marketing	1	8,821
India	Hotel Booking	1	10,986
	Deception (solve marriage, depression)	1	15,128
	Tech Support	1	43,552
Indonesia	Hotel Booking	1	8,291
	Product Marketing	75	2,689,616
	Pornography	4	164,382
	Deception (solve marriage, deception)	7	101,799
	Followers	15	406,713
	Finance, real estate	3	23,700
	Selling adult products	5	48,109
	Uncategorized	3	29,043
Kuwait	Charity (donation)	1	46,494
Mexico	Pornography	1	8,204
Nigeria	Deception (solve marriage, depression)	1	29,226
Pakistan	Finance, real estate	1	16,058
Spain	Charity (donation)	1	14,311
UAE	Pornography	5	65,593
USA	Party Reservations	8	172,090
	Product Marketing	1	22,804
	Pornography	1	19,653
	Deception (solve marriage, depression)	1	12,936
	Escorts	1	9,652
UK	Escorts	1	9,268
	Charity (donation)	2	17,184
Venezuela	Hotel Booking	1	6,813
	Free games, downloads	1	9,028
Unknown	Party Reservations	10	323,565
	Hotel Booking	2	11,334
	Product Marketing	10	108,634
	Free games, books, downloads	1	8,834
	Pornography	17	211,714
	Uncategorized	2	10,266
	Deception (solve marriage, depression)	5	48,093
	Finance, loans, real estate	2	34,226
	Charity (donation)	2	29,740

fraction of content appeared within a few minutes, it is likely that content generation is automated. To ascertain this, we looked at the information of the client (provided by the Twitter API) used by spammers to interact with the Twitter API or their web portal. We found that most of the content was generated using 'twittbot.net', a popular bot service, known to be used by spammers [179]. Apart from the bot service, several other clients like RoundTeam (0.25%), IFFTT (0.03%), Buffer (0.017%) and Botize (0.016%), etc. were used for Twitter. Besides, we found that volume per phone number was also high in Indonesian campaigns; 80% phone numbers had more than 1000 posts. One would assume that volume per phone number would be low since there are humans at the other end to service the requests. However, by processing the text in the posts created in this campaign, we found that it heavily relies on sending SMS or WhatsApp ($\sim$ 71% posts). This explains why spammers would be able to handle the load of interacting with victims. There are many other advantages of using these messaging services – spammers can further send phishing messages to victims and communicate with them unmonitored.

Second, we find that visibility (number of likes, shares, and retweets) of a post is positively correlated with the volume of posts (Pearson coefficient value was 0.97, the p-value was recorded as 0.000). While this may sound intuitive, the number of accounts that were suspended within a campaign were not positively correlated. We noticed that even though the volume generated by Indonesian campaigns is 98.2% higher than Indian campaigns, the fraction of users suspended in Indian campaigns was 85.6% higher. This indicates that the account suspension is dependent on the nature of campaigns; campaigns providing escort services or technical support services had more accounts suspended. Surprisingly, for similar escort service campaign running in two different countries, USA and UAE, there was a significant difference in the number of accounts suspended. The number of posts generated by escort campaign running in the USA (9,652) was lower than that running in UAE (69,263), but 55.6% user accounts were suspended in the USA in comparison to only 9.1% accounts suspended in UAE. We looked at several reasons which could potentially lead to account suspension – volume generated per user or URLs used in the posts. We noticed that volume per user was higher for UAE users (Figure 4.4(a)), number of URLs shared in UAE campaign was higher, and words used in both the campaigns had a good overlap. While we can not cite the exact reason for suspension, from Figure 4.4(b), we observed that inter-arrival time between two consecutive posts in the USA (41s on an average) is lesser than that of posts made in the UAE campaign (3922s on an average) which could be a potential reason for suspension. Nevertheless, this disproportional rate of suspensions for USA users as compared to UAE users remains a surprising find; OSN service providers should have stringent algorithms in place which can detect some aberrations.



(a) Volume per account low for USA accounts.

(b) Inter-arrival time lesser for USA accounts.

Figure 4.4: Comparing Escort service campaign in USA vs. UAE.

4.3.2 Modus operandi

To ascertain the attack methodology the victims faced, we performed an experiment after receiving our institute's IRB approval. Pretending to be a potential victim, we made calls to campaigns running in USA and UAE selling adult (Viagra) pills, in Indonesia, selling herbal products, and in India promoting tech support and astrology services providing solutions to marriage and love problems. To avoid time zone conflict, we called the same set of numbers at different points of time across 24hrs of the day. Apart from Indonesia, campaigns from other countries had an IVR deployed, before reaching a spammer. We posit this can help in load balancing between limited spammers. Due to language limitation in Indonesia, spammers preferred chatting over platforms like WhatsApp, where they were extremely responsive. A transcript of an interaction with a USA based spammer selling Viagra pills is shown below:

```
IVR: Press 1 to know about our products, 2 to check the status of previous order
and 3 for other inquiries
Victim:  *pressed 1*
IVR: Press 1 to know more about <company-name> viagra pills and 2 for
other products.
Victim: *pressed 1*
IVR: *call forwarding to human*
Scammer: Hello, I'm <name>, speaking from <company-name>, what would you like
to know about the <brand> viagra pills.
Victim: What are the various packs I can buy and how much does it cost?
```

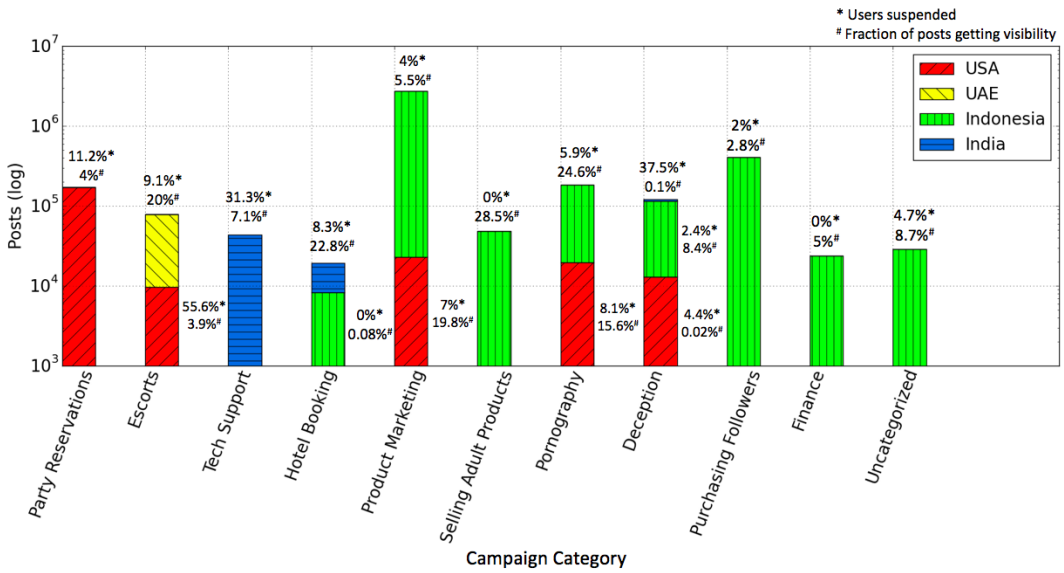


Figure 4.5: Comparison of campaigns running in the top 4 countries – Indonesia, USA, India, and UAE across different campaign categories. Indonesia generates maximum spam campaigns (volume) but fraction of accounts suspended in India is higher.

```
Scammer: We have only one variant which costs $99 - $119 for
the pills and $20 for delivery.
Victim: Okay. How can I pay for the order if I decide to order? Do you have a web
portal where I can make an online transaction?
Scammer: No sir, currently, we're operating only over phone, so you can provide
your VISA card details to me, and I'll be happy to place the order for you.
Victim: Is phone the only option? I would like to make the payment through the
web portal.
Scammer: Sorry sir, but we operate only over phone.
Victim: Okay, what are the product guarantees you offer?
Scammer: Yes, sir please be assured that we provide 100 percent return guarantee.
Victim: Can I get some samples before placing the order?
Scammer: I am sorry sir, we don't provide samples. Should I place an
order for you?
Victim - No, thank you for the information.
```

The campaigns in USA and UAE were not limited by any delivery location; they had a usual delivery time of 2-4 weeks. These campaigns were operating solely over the phone and had no option of visiting an online portal to make the transaction. The attackers confidently asked for the credit card details over the phone even though banks suggest otherwise. Spammers from Indonesia

told that they would start delivery only after receiving the payment, which was to be done via bank transfer. During the interactions, spammers were persuasive in selling products by claiming their products to be the best as compared to similar products in the market. Tech support campaigns in India were providing service to users remotely over the Internet and charged over call once the issue was 'fixed'. Spammers pretended that there is a problem with victims' computer and then try to convince the victim to pay them to fix it, as reported in several complaints [20]. Another astrology based spam running in India tricked by promising to fix our marriage and love related problems within 48 hours [12]. We called 3-4 numbers in different states from India. Interestingly, all the spammers had a similar way of dealing with the problem, where they asked to send personal details over WhatsApp.

It is evident that spammers running campaigns in different countries deploy similar mechanisms to let the victim reach them (posts on social media), to set up the product delivery operation (product delivery post payment and service delivery prior to payment), and model of payment (details transfer via phone, WhatsApp, verbal). It is the product delivery operation that creates deliberate confusion; intuitively, the delivery mechanism is similar for benign campaigns. Spammers take advantage of such confusion, offer fake promises and later do not deliver.

4.4 Characterizing Cross-Platform Spam Campaigns

After understanding the distribution of phone spam campaigns across different geographical areas, we now study spammers' maintenance efforts to retain accounts against suspension, maximize visibility and thus, maximize their target audience. Usually, we observe that spam campaigns do not limit themselves to one OSN and are rather present on multiple networks. Evidence also clearly suggests that campaigns do cross-pollinate across social networks. Study of the distribution of spam categories (topics) across OSNs is aggregated in Table 4.2.

The distribution of posts in top 3 spam campaigns: Loveguru (from Deception category), Tech Support, and Indonesian Herbal Product (from Product Marketing category) is shown in Table 4.3. Even though Twitter has the largest fraction, all OSNs are abused to carry out spam campaigns. In the *LoveGuru* campaign, astrologers promise victims to fix their love and marriage related problems. In the *Tech Support* campaign, spammers pose as technical support representatives or claim to be associated with big technological companies (like Amazon, Google, Microsoft, Gmail, Quebec, Norton, Yahoo, Mcafee, Dell, HP, Apple, Adobe, TrendMicro, and Comcast) and offer technical support fixes. This campaign had incurred financial losses of $2.2M to victims, as per FBI [146]. In the *Indonesian Herbal Product* campaign, spammers sell a variety of products ranging from beauty products, growth pills for children to organ enlargement pills. In their posts, spammers ask victims to text them via SMS / WhatsApp on the phone numbers under their control. While selling

Table 4.2: Distribution of all campaign categories across OSNs. Tech Support, Deception, and Product Marketing campaigns have significant volume across OSNs (arranged in decreasing order of spamicity).

Category	TW	FB	G+	YT	FL
Tech Support	28,984	1,974	12,303	2,850	1,737
Deception	204,819	2,542	4,257	101	63
Product Marketing	2,827,867	70	1,514	82	342
Charity (Donation)	106,593	224	911	1	0
Adult (products, escorts)	572,583	41	39	4	1
Party Reservations	535,054	74	3	0	0
Finance, real estate	73,968	16	0	0	0
Uncategorized	45,993	5	0	2	0
Purchasing Followers	406,713	0	0	0	0
Hotel Booking	55,652	0	0	0	0
Free games, books	17,862	0	0	0	0

beauty products might look legitimate, the sale of a certain class of products (like private organs' enlargement pills) and the scale at which content is generated raises suspicion.

Table 4.3: Top cross-platform spam campaigns. All three have good coverage across multiple OSNs.

Campaign	TW	FB	G+	YT	FL
Tech Support	28,984	1,974	12,303	2,850	1,737
LoveGuru	6,934	1,418	4,257	101	63
Indonesia Herbal Product	1,443,619	9,238	21	46	336

In this section, we focus on studying in detail the largest spam campaign - the Tech Support campaign. Over the course of six months of data collection, we got a total of 43,552 posts (43K) spread across all five OSNs propagating to the extent of 41 phone numbers. The complete dataset description for tech support campaigns is shown in Table 4.4.

As phone numbers are one of the primary tokens used by spammers, we examined carrier information tied to each number to identify patterns in how spammers source numbers. We derived this

Table 4.4: Statistics for Tech Support campaign.

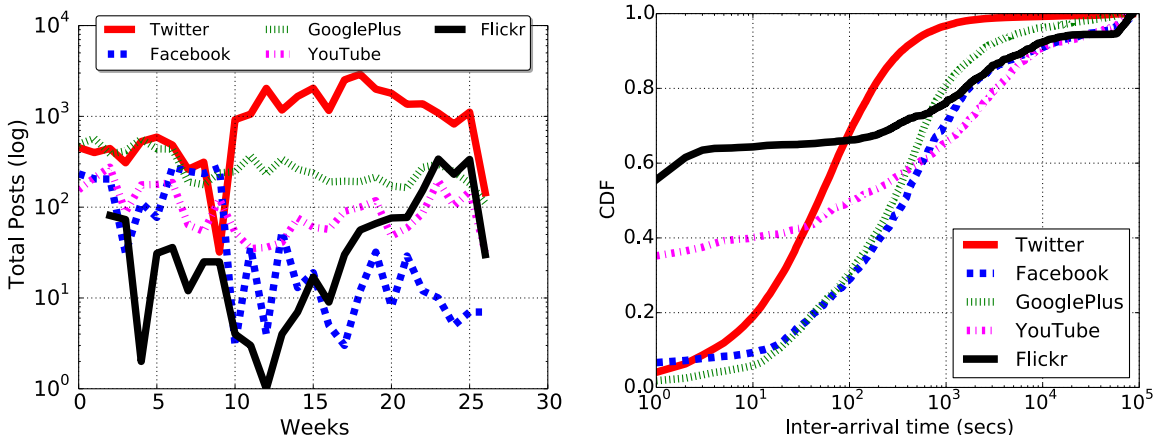
	TW	FB	G+	YT	FL
Total Posts	28,984	2,151	7,830	2,850	1,737
Posts w/ URL	25,245	1,391	5,714	227	1,503
Distinct Phone nos	41	33	37	39	20
Distinct User IDs	748	289	360	433	79
Distinct Posts	16,142	1,797	6,570	2,050	1,449
Distinct URLs	68	951	3,189	80	293

information from several online services like Twilio (mobile carrier information) [8], Truecaller (spam score assigned to the phone number) [9], and HLR lookups (current active location of the phone number). [10] We found that all the phone numbers used in the Tech Support spam were toll-free numbers. Using a toll-free number offers several advantages to a spammer: 1) increased credibility: it does not incur a cost to the person calling, hence people perceive it to be legitimate, 2) it provides international presence: spammers can be reached from any part of the world. Besides, we found that spammers used services like ATL, Bandwidth, and, Wiltel Communications to obtain these toll-free numbers and that a majority of them were registered between 2014 and 2016.

4.4.1 How does content cross-pollinate?

Now, we answer the following question: *Is a particular OSN preferred to start the spread of a campaign?* and *Is there a specific pattern in the way spam propagates on different OSNs?*

Figure 4.6(a) shows the temporal pattern of content across OSNs. We point out a limitation of our dataset - our collection runs for six months while a campaign may have existed before and / or after this period. While the longest detected active time for a campaign in our dataset is 186 days, the actual time may be greater.



(a) Posts across OSNs.

(b) Inter-arrival time of posts appearing on OSNs.

Figure 4.6: Temporal properties of Tech Support Campaign across OSNs - all OSNs are abused to spread the campaign but volume is maximum on Twitter.

A majority of these posts are densely packed into a small number of short time bursts, while the entire campaign spans a much longer period. Though the volume of content is significantly higher

[8] https://www.twilio.com/
[9] http://truecaller.com/
[10] https://www.hlr-lookups.com/

Table 4.5: Distribution of phone numbers according to their first appearance amongst OSNs. Flickr is never chosen as a starting point and there is no particular sequence in which spam propagates across OSNs.

Starting OSN	#Cases	Most common sequence
Twitter	12	TW ->G+ ->YT
GooglePlus	10	G+ ->TW ->YT ->FB ->FL
Facebook	6	FB ->G+ ->TW ->YT
YouTube	13	YT ->G+ ->TW ->FB

on Twitter, all OSNs are consistently being abused for propagation. Inter-arrival time, i.e., the time between two successive posts is observed to be least on Twitter (308s), as shown in Figure 4.6(b). It is interesting to note that a few campaigns on Flickr have an inter-arrival time between two posts close to 1s, even though the average inter-arrival time is highest on Flickr. As Figure 4.6(a) shows, the volume on Flickr increased during the last few weeks of our data collection period. We divided the inter-arrival time into two time windows; first 15 weeks, and last 11 weeks. We observed that the average inter-arrival time in latter time window dropped from 9786s to 2543s which means spammers are had started heavily abusing Flickr to spread the Tech Support campaign. It is hard to ascertain the motivation of the spammers in sending high volume content on Twitter, but, we speculate one of the reasons could be the public nature of the Twitter platform, as compared to closed OSNs like Facebook.

For all the phone numbers, we analyzed the appearance of phone numbers on different OSNs, and the order in which they appear, as reported in Table 4.5. For each network that is picked as the starting point, we identified the most common sequence in which phone numbers appeared subsequently on other OSNs. We found that Flickr was *never* chosen as the starting OSN to initiate the spread of a phone number. There was no specific sequence of OSNs that was chosen to propagate the campaign. Besides, we noticed that the posts originating from YouTube took the maximum time to reach a different OSN with an average inter-OSN time of 5 hours. The human effort required to upload content to YouTube could be a plausible reason.

To summarize, we observed that all OSNs were abused to spread the Tech Support campaign, and no particular OSN was preferred to drive the campaign. In addition, there was no particular sequence in which spam propagated across OSNs.

4.4.2 How do spammers maximize visibility?

We observed various strategies adopted by spammers to increase the dissemination of their posts. In this section, we discuss those strategies and their effectiveness.

The *Visibility* of a post is defined as the action performed by the user (consumer of the post) in

terms of liking or sharing the post, which accounts for traction a particular post received. For each network, we define the value of visibility as the number of likes and reshares on Facebook, +1s and reshares on GooglePlus, the number of retweets on Twitter, and video like count on YouTube. We did not consider Flickr in our analysis since it gives only the view count of the image posted on the platform. A user only viewing an image cannot be assumed to be a victim of the campaign, and is hence ignored in our analysis for visibility. To calculate visibility in all scenarios, we collected the *retweets, plusoners / reshares,* and *likes* from Twitter, GooglePlus, and Facebook using their APIs. We collected this data six months after our data collection period, as posts take time to reach their audience. Apart from calculating values for each visibility attribute, we also collected properties of the user accounts involved, i.e., the IDs of user accounts involved in retweeting / liking / resharing the content.

To increase the visibility of content, we observed that 67% of posts contained Hashtags (for marketing [60], gaining followers [133]), 82.7% of posts contained URLs (for increased engagement with potential victims), 12.1% of posts contained Short URLs (for obfuscating the destination of a URL and getting user engagement analytics), and 72% of posts contained photos (as visual content gathers more attention). We also noticed collusion between accounts and cross-referenced posts to increase the visibility of the campaign on the users.

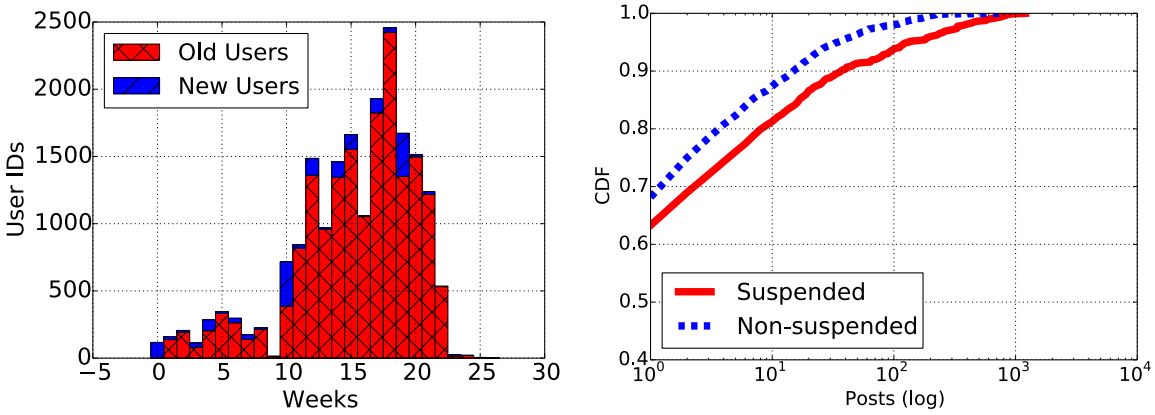
Cross-referenced posts: We call a post cross-referenced if it was posted to OSN X, but contains a URL redirecting to OSN Y. For instance, a Twitter post containing a link 'fb.me/xxxx' which would redirect to a different OSN, Facebook. Spammers either direct victims to existing posts or to another profile which is propagating the same campaign on a different OSN. In the Tech Support campaign, we observed that 3.2% of Facebook posts redirected to YouTube, and 1.78% of posts redirected from GooglePlus to YouTube.

Collusion between accounts: In the Tech Support campaign, we observed traces of collusion, i.e., spammers involved in a particular campaign, *like / share* each other's post on OSNs or like their content to increase reachability. Collusion helps in cascading information to other followers in the network.

We calculated the visibility received by all the posts after removing likes / reshares / retweets by the colluders (i.e., accounts spreading the campaign already present in the dataset). We noticed that the posts containing the above-mentioned attributes (hashtags, URLs, short URLs, photos, cross-referencing, and collusion) garnered around ten times more visibility than posts not containing them. Around 10% of the posts saw traces of collusion, contributing to 20% of the total visibility. Maximum visibility (22.1% of total visibility) was collectively calculated for posts containing hashtags. In addition, we observed that a major chunk of visibility came from GooglePlus, followed by Facebook. This shows that the audience targeted influences the visibility garnered by a particular campaign, as GooglePlus is known to be consumed mostly by IT professionals [30].

4.4.3 Are OSNs able to suspended user accounts?

To aid in the propagation of a campaign, spammers manage multiple accounts to, garner a wider audience, withstand account suspension, and in general increase the visibility. Individual spammer accounts can either use automated techniques to aggressively post about a campaign or use hand-crafted messages. In this section, we examine the behavior of user accounts behind the Tech Support campaign. Spammers want to operate accounts in a stealth mode, which requires individual accounts to post few posts. In reality, it costs effort to get followers to a spam account, and the number of 'influential' accounts owned by a spammer can be limited. Thus, the spammer tends to repeatedly use accounts to post content keeping low volume per account (Figure 4.7(b)), while creating new accounts once in a while (Figure 4.7(a)).



(a) New users created from time to time for campaign sustainability.

(b) Volume per user kept low to evade suspension.

Figure 4.7: New user accounts created from time to time and volume per ID kept low, to avoid suspension in the Tech Support Campaign.

Long-lived user accounts: During our data collection, we found that 68.7% (1,305) of the accounts were *never* suspended or taken down on any of the five OSNs. This is in strike contrast to the URL based campaigns [179], where the authors observed that 92% of the user accounts were suspended within three days, on an average, of their first tweet. To take into account delays in the OSNs' account suspension algorithm, we queried all the accounts six months after the data collection to determine which accounts were deleted / suspended. This process consists of a bulk query to each OSN's API with the profile ID of the account. [11] For each of these accounts, we looked at the time stamp of the first and last post within our dataset, after which we assumed that the account was suspended immediately. Out of the accounts which were suspended, around 35%

[11] If the account is deleted / suspended, a) Twitter redirects to `http://twitter.com/suspended`, and returns error 404, b) Youtube returns 'user not found', c) Facebook returns error 403 in case the account is suspended, d) GooglePlus throws a 'not found' error, e) Flickr responds with a 'user not found' error.

of the accounts were suspended within a day of their first post; the longest lasting account was active for 158 days, before finally getting suspended. On an average, accounts got suspended after being active for an average of 33 days. This is in clear contrast to users getting suspended within three days (on average) for URL based spam campaigns, and thus, focused efforts are needed to strengthen defense from evolving phone-based spam campaigns.

4.4.4 Is existing intelligence based on URLs useful?

Apart from creating accounts to propagate content, and using phone numbers to interact with victims, spammers also need a distinct set of URLs to advertise. In this section, we look at the domains, subdomains and URL shorteners used by spammers. As compared to phone numbers which incur more cost, we found the presence of more URLs, which are relatively cheaper.

Given the prevalence of spam on OSNs, we examined the effectiveness of existing blacklists to detect malicious domains. Specifically, we used Google safe browsing [12] and Web of Trust (WOT) [13] to see if they were effective in flagging domains as malicious. Web of Trust categorizes the domains into several buckets along with the confidence to assign a category. We marked a domain as malicious if the domain appeared in any of the following categories – negative (malware, phishing, scam, potentially illegal), questionable (adult content). In addition, for better accuracy, we only considered cases where confidence was more than 10. We checked the URLs and domains even after six months of data collection since blacklists may be slow in updating response to new spam sites. We marked a URL malicious if it was listed as malicious either by Google safe browsing or WOT. Of all the posts, we had 4,581 unique URLs and 594 distinct domains. We checked these domains against the blacklists, finding that 10% of the domains were blacklisted by WOT, none by Google safe browsing. Table 4.6 shows the different categories in which the domains were listed. Please note one domain can be listed in multiple categories. Of all the URLs, 12.1% were shortened using bit.ly; 3% of them received over 69,917 clicks (data collected from bit.ly API), showing that the campaign was fairly successful. Overall, we found that existing URL infrastructure was ineffective to blacklist URLs used in phone-based spam campaigns.

4.4.5 Can cross-platform intelligence be used?

Given that existing URL infrastructure is ineffective, we study if cross-platform intelligence across OSNs can be used. To this end, we look at the spam user profiles across OSNs to figure out which OSN is most effective in building the intelligence.

Homogeneous identity across OSNs: Simply analyzing users' previous posts might not be

[12] https://developers.google.com/safe-browsing/v4/lookup-api
[13] https://www.myWOT.com/wiki/API

Table 4.6: Web of Trust categories for all URLs in Tech Support Campaign.

Category	Count	Example
Scam	24	nortonhelp.support
Phishing	2	technicalsupporthelpline.page.tl
Tracking	32	www4.zippyshare.com
Malware	2	www.it-servicenumber.com
Illegal	3	newlondon.backpage.com
Suspicious	2	microsoft-windows-support.com
Spam	4	easternshore.backpage.com

sufficient, as users can switch between multiple identities, making it hard for OSN service providers to detect and block them. Moreover, spammers may appear legitimate based on the small number of posts made by a single identity. The challenge remains in analyzing the aggregate behavior of multiple identities. To understand how user activity is correlated across OSNs, we pose the question: *do users have a unique identity on a particular OSN or do they share identities across OSNs? Within the same network, can we find the same users sharing multiple identities?*

To answer this, we looked at user identities across different OSNs in *aggregate* (multiple identities of the same user across different OSNs) and *individual* (multiple identities of the same user on a single OSN) forms. If the *same* user has multiple identities, sharing similar name or username, it is said to exhibit a homogeneous identity. To define user identity in a particular campaign, we used two textual features: *name* and *username* [106,148]. Since networks like YouTube and Google Plus do not provide the username, we restrict matching to identities sharing the same name. We used Levenshtein distance to find similarity in usernames. $\mathrm{LD}(s_i, s_j)$ is the Levenshtein edit distance between usernames s_i and s_j. Here, $\mathrm{LD}(s_i, s_j) = 1$ means the strings are identical, while $\mathrm{LD}(s_i, s_j) = 0$ means they are completely different. After manual verification by comparing profile images across OSNs, we found users having LD $>= 0.7$ are homogeneous identities. We found four cases where multiple user identities were found for the same user within the same network, and in 65 instances, multiple user identities were present for the same user in more than two networks. Specifically, we found 51 users sharing multiple identities across two different OSNs, and 10 users sharing multiple identities across 3 OSNs. We noticed that these accounts shared same phone numbers across OSNs; some accounts post more phone numbers that are part of tech support campaign.

Overall, we found that the total number of posts made by these accounts was highest on GooglePlus (2696), followed by Twitter (1776), Facebook (577), Flickr (387), and YouTube (323). Out of all the homogeneous identities, the following are the percentages of accounts suspended on each OSN – Twitter (60%), YouTube (48%), GooglePlus (32%) Flickr (33%), and Facebook (4%). Our data is insufficient to determine whether account suspension is due to dissemination of content across

OSNs or other unobserved spammers' properties like language used, blocked / reported by OSN users and like. Notwithstanding, the association that we observe, strengthens the fact that sharing information about spammer accounts across OSNs could help OSNs to detect spammers accurately.

Reducing financial loss and victimization The actual number of users that are impacted depends on how many victims called spammers and bought the products advertised by campaigns. Since it is hard to get this data, we provide a rough estimate of the number of victims falling for campaigns identified in our dataset. We find reputation of spammers in terms of their followers count on Twitter, friends / page likes on Facebook, circle count on GooglePlus, and subscriber count on Youtube. As these users have subscribed to spammers to get more content, they are likely to fall for the spam. Some of the users would be the ones who aren't aware of the campaign being spam, while some followers / friends could be spammers themselves who have followed other spammers' accounts. We again collected this data after 6 months of our data collection and recorded 637,573 followers on Twitter, 21,053 friends on Facebook, 11,538 followers on GooglePlus, and 2,816 likes on YouTube amounting to a total of 670,164 users. Please note that this number is a lower bound, as we were not able to retrieve statistics for suspended / deleted accounts. Assume that we transfer knowledge from Twitter to other OSNs and prevent the onset of campaigns on other OSNs, we analyzed how much money and victims could be saved. Looking only at the friends, followers, and likers on Facebook, GooglePlus, and YouTube respectively, we could save 35,407 (21,053 + 11,538 + 2,816) unique victims and $8.8M (35,407 * $290.9) by transferring intelligence across OSNs. We used the average cost of the Tech Support Spam to be $290.9 per victim, as reported by Miramirkhani et al. [139].

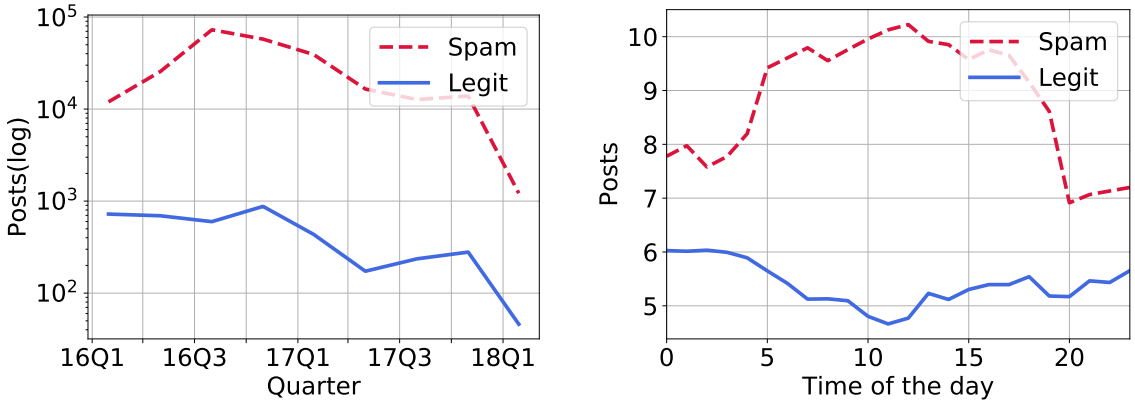
4.5 Legitimate vs. Spam Tech Support

In this section, we compare the characteristic difference between the accounts involved in propagating spam and legitimate Tech support campaigns.

Within the Tech support campaign, we curated the list of brands / organizations that were being targeted by the spammers to coerce victims. We found 16 such brands viz Microsoft, Gmail, Facebook, Yahoo, McAfee, etc. To find the legitimate dataset, for each brand, we searched the official verified website on Google and took the official phone numbers that were being used for handling respective technical support. Further, we took all the phone numbers used by legitimate handles and searched for tweets containing those phone numbers using Twitter Streaming API. Table 4.7 presents basic statistics for the two campaigns. The relative sizes of these posts illustrate the scale of the problem: spam is approximately 47 times larger than the legitimate posts received.

Table 4.7: Characteristics of attributes for spam and legitimate Tech support campaigns.

Category	Spam	Legitimate
#Posts	269,652	5,712
#Unique Phone Numbers	1,164	279
#Unique IDs	6,077	794
#Suspended IDs	67,757	47



(a) Volume per quarter

(b) Volume per hour

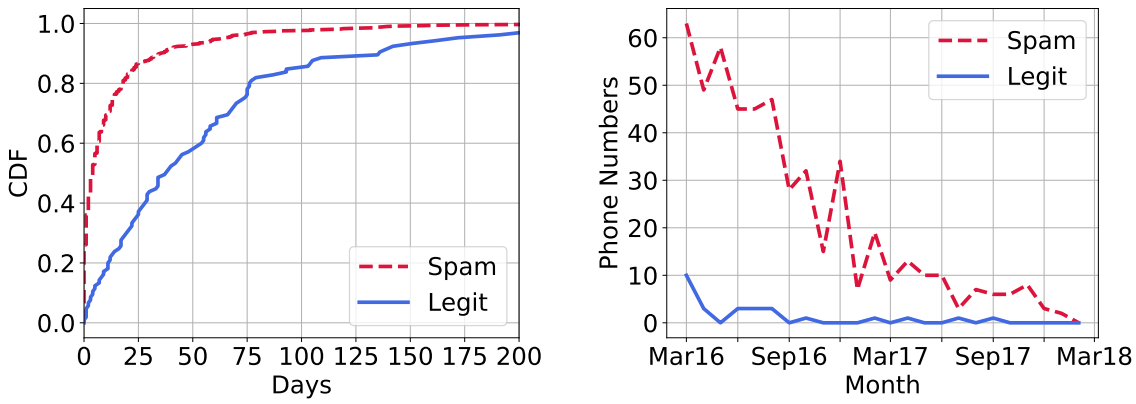
Figure 4.8: (a) Volume generated in spam campaigns is higher than that generated in legitimate campaigns to maximize reach. (b) Hours of operation in both the campaigns is complementary.

4.5.1 General Characteristics

As seen in Figure 4.8(a), proportion of the data collected for both legitimate and spam campaigns remain the same through out the period of collection; volume generated in spam campaigns is higher than the volume generated in legitimate campaign. This is because spammers need to maximise the reach and target as many victims as possible, hence the large number of tweets. Figure 4.8(b) shows the difference between the two classes in terms of time of day when tweets were posted from these accounts. Hours of operation are almost complementary in both the campaigns. In addition, we observed that spam campaigns pick up early and decay during the day in terms of volume, but legitimate campaigns consistently post during the day without long spikes.

4.5.2 Phone Number Reusability

For each phone number, we calculated the number of days between consecutive occurrences for a phone number, defined as *reusability*. We observed that around 50% phone numbers appeared again



(a) Phone number reusability

(b) New numbers every month

Figure 4.9: (a) Spam phone numbers are reused more; one phone number is not used at a stretch. (b) Spam phone number pool is replenishd with new phone numbers every month to avoid pattern detection.

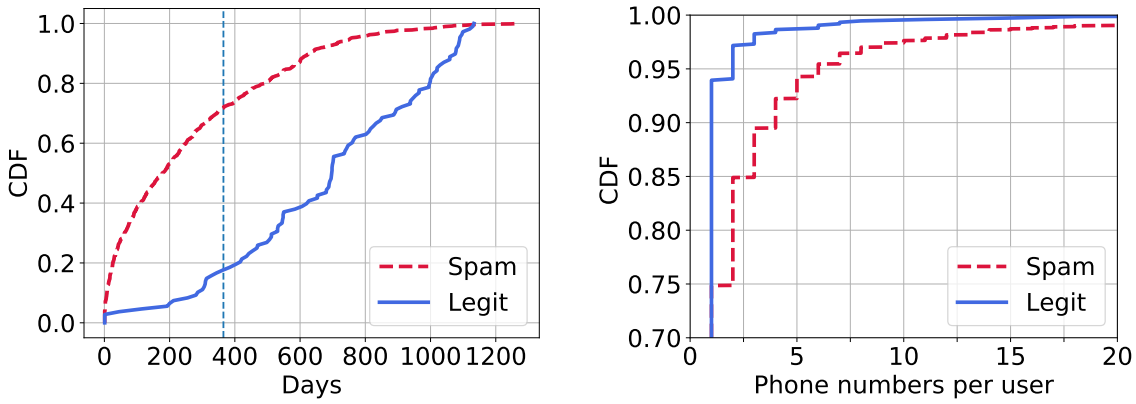
in less than 5 days. In addition, 70% phone numbers used in spam campaigns were being reused within 10 days of the first appearance. This shows that the pool of phone numbers is not kept for long and is replenished in sometime (see Figure 4.9(a)). Figure 4.9(b) shows that new phone numbers appear every month; spammers keeping switching between phone numbers and not use a particular phone number for a very long time.

Figure 4.10(a) shows that the same spam phone number is not used for a very long time whereas legitimate phone numbers have a comparatively higher lifetime (difference between first and last post made using that phone number). Previous studies have also indicated that most of the spam comes from IP addresses that are extremely short-lived to avoid detecting behavioral patterns from historical data [182]. In addition, the phone numbers used per spam account is more than a legitimate user, as shown in Figure 4.10(b). Spammers could imploy such tactic to regulate the volume per phone number to avoid detection by OSNs. Further, since there are physical entities handling the phone call requests, number of phone numbers per spam account is not very high.

In conclusion, spam phone numbers have shorter lifetime and are more reused, i.e., a single phone number is not used for a very long time at a stretch, but reappears in sometime.

4.5.3 Brand Propagation

Brands are the top companies / organizations a user tweets about. We first selected the top 15 brands that represented majority of our data. Products belonging to the same companies were grouped together for e.g. Instagram and WhatsApp were clubbed along with Facebook as the parent brand (company) is the same. Any tweet that does not mention any of these brands was



(a) Lifetime of a phone number

(b) Phone numbers per user

Figure 4.10: (a) Lifetime of a spam phone number is lesser than legitimate phone number since new phone numbers are added in the pool. (b) Nearly 95% of the legit users have only one phone number whereas a lot of spam users employee multiple phone numbers to maximise reach and regulate volume per phone number to avoid detection.

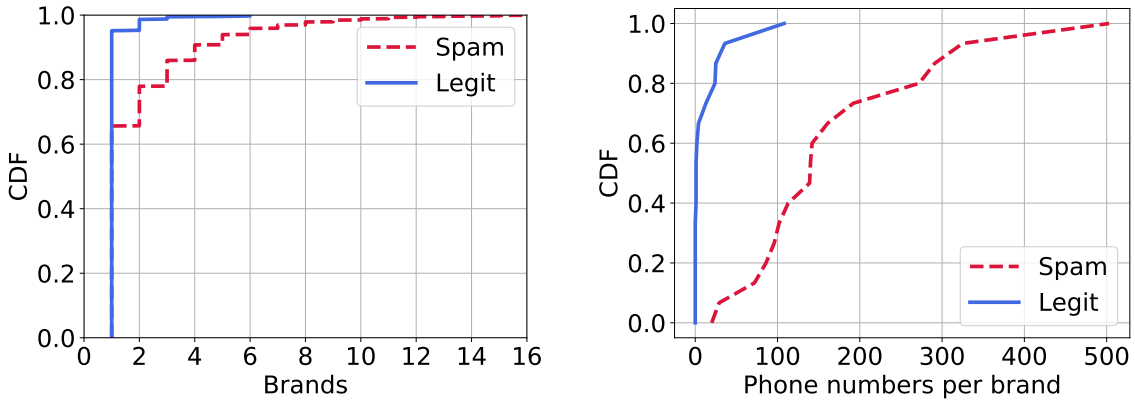
taken as *Others*. Figure 4.11(a) shows that majority of the legitimate users only talk about one brand but nearly 35% of the spam users mention more than one brand. This could be because it increases spammers' probability of receiving a tech support related victim phone call. On the other hand, the legitimate tech support only talks of the brand they are serving.

In addition, the number of phone numbers used to propagate Tech Support about any one brand in spam data was much greater than legitimate. More than 90% legitimate brands had a few phone numbers while spammers used several hundred phone numbers in operation, using 3 or 4 phone numbers in the same tweet. Using more phone numbers per brand helps spammers to handle more requests for a particular brand, thereby maximizing the reach.

Even though phone numbers per spammer were less (see Figure 4.10(b)), overall, the number of unique phone numbers used in propagating spam campaigns were higher than the phone count used in legitimate campaigns.

4.5.4 Lifetime of Spammers

Lifetime of a spammer and legitimate account was calculated by taking March 1, 2016 as the starting date (beginning of data collection) and represented as the number of days user between the date user created an account on Twitter and starting date. Negative lifetime means the user account was created before the start date. As Figure 4.12(a) shows, legitimate accounts have been on Twitter much longer than spam accounts. Since the volume posted by legitimate users is lesser



(a) Brands per user

(b) Phone numbers per brand

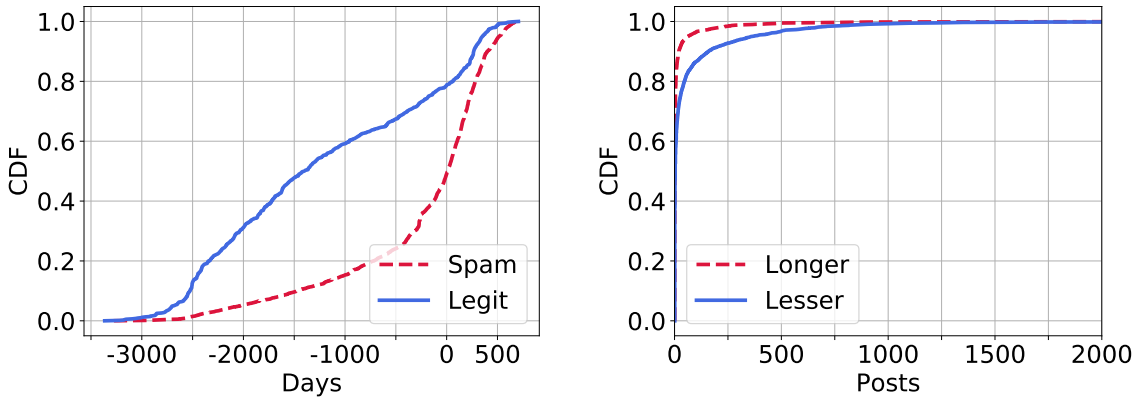
Figure 4.11: (a) Spammers tweet about multiple brands, use multiple phone numbers for a single brand. (b) On the other hand, legitimate users tweet about a single brand and in more than 90% cases, use one phone number per brand.

than spammers, they did not get suspended by Twitter. In addition, we observed that the spam accounts which were not suspended by Twitter posted fewer tweets, as shown in Figure 4.12(b).

4.5.5 Network Characteristics

We analyzed the frequency at which spam and legitimate users interact within their own group. Spammers operate in cohorts. They post similar content, target the same brands (major tech companies like Apple, Google, Microsoft), retweet each other and share phone numbers. Group of spammers collude so that their malicious content spreads out effectively. The high density of connections among spam users suggest a strong modularity, as shown in Figure 4.13(b). In this network graph, each node represents one user and nodes are connected if one user mentions the other and a user can be tech support provider (spammer / legitimate user) or someone looking for tech support information online. The network graph generated using Louvain Algorithm [14] showed high modularity for spammers. In contrast, the legitimate clustering graph depicted well defined boundaries. Legitimate tech support providers have little incentive to collaborate since they only provide tech support for their own products.

[14] `https://perso.uclouvain.be/vincent.blondel/research/louvain.html`



(a) CDF of lifetime of a user.

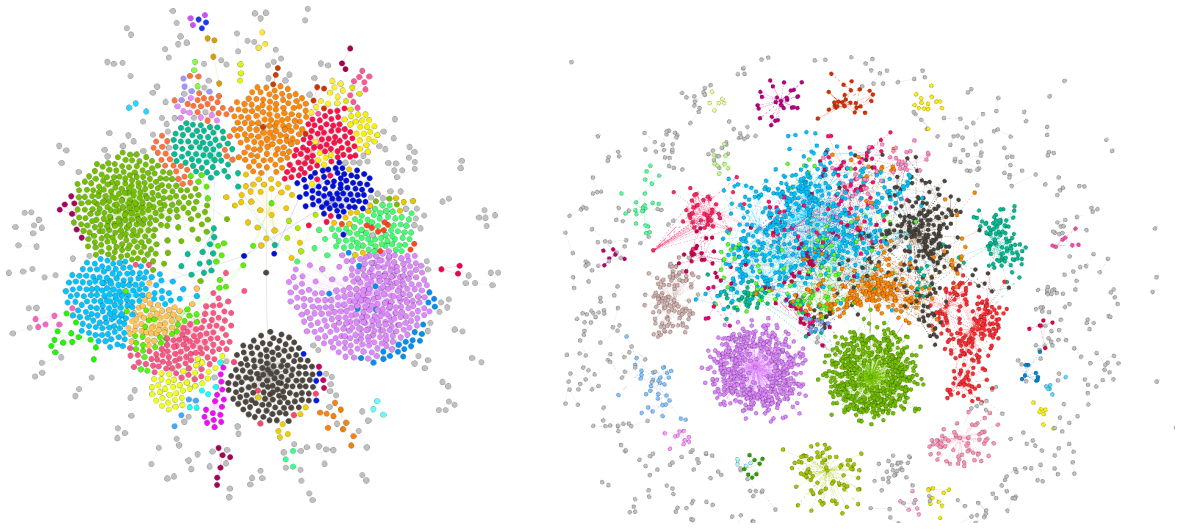
(b) Volume of users who have stayed lesser vs. longer time.

Figure 4.12: (a) Lifetime of a spam account tends be much smaller than legitimate account because Twitter suspends spam accounts due to high volume of tweets. (b) Accounts that have posted more tweets were suspended by Twitter sooner.

4.6 Discussion

Getting a fair idea about spammers' motivation is hard, however, we believe that our first-of-its-kind analysis of these phenomena still provides great value and opens new doors to understand the phone-based spammer ecosystem across OSNs better. In this section, we provide a synthesis of our evaluations and propose some recommendations to OSN service providers.

For the 238 clusters (including the ones for which topics were not inferred) identified in the dataset, we found 5,525,772 posts that propagated 2,347 unique phone numbers. Out of 157,494 user accounts, only 9,556 were suspended showing that OSN service providers are unable to bring down spammers accounts due to low volume of phone-based spam campaigns, as compared to URL-based campaigns. To verify the difference in volume of OSN content containing URLs and phone numbers, we gathered a random sample of one million tweets from Twitter; found URLs in 30% of posts and phone numbers in only 0.5% of posts. Out of 5,525,772 posts in 238 clusters, we observed that these posts contained 426,775 distinct URLs (and 2,133 unique domains), out of which only 2.1% were blacklisted by existing URL blacklists: Google Safe browsing and Web Of Trust. Since URLs are absent in majority of the phone-based spam posts, existing research solutions based on detect- ing and preventing URL-based spam are inappropriate. In addition, malicious sites are ephemeral; many of the URLs can point to destinations that no longer exist, rendering URL blacklists ineffective. Phone numbers, on the other hand, are a more stable and reliable re- source since spammers need to provide their real phone numbers so that victims can reach out to them. A solution appropriating the phone number, therefore, would be more reliable in bringing down spammer



(a) Legitimate users (b) Spammers

Figure 4.13: Network Graph using *user mentions* shows high modularity for spammers, compared to legitimate users. Each color represents different communities. (a) Legitimate users have loose community, modularity coefficient < 0.5, while (b) spammers have a dense structure where they mention each other in their tweets achieving a high modularity coefficient (0.85).

accounts.

Providing feedback via Twitter is seen as one of the powerful tool for prompt grievance redressal, where anybody with a grievance against a company can be heard by fellow customers. The company can instantly find information on the consumer before deciding whether and how to respond. A recent study showed that 99% of brands are on Twitter, and 30% of them have a dedicated customer service handle. [15] The average response time was 5.1 hours with 10% of companies answering within an hour, and 93% of companies answering within 48 hours. While the social media interaction helps in strengthening the customer-brand relationship, our work sheds light on it's exploitation by spammers. Figure 4.14 depicts a Twitter user tagging a spam Tech Support handle to get their issue resolved.

How spammers can be choked? Phone numbers are a stable resource for spam since spammers need to provide their real phone numbers so that victims can reach out to them. A solution built around phone numbers, therefore, would be more reliable in bringing down spammers. As a countermeasure, there are two potential mechanisms – a) phone blacklist and b) suspension of OSN accounts. A *phone blacklist* should be created, similar to URL blacklists, to check if a phone number is involved in a spam / scam campaign. Blacklisting a phone number would break the

[15] `http://get.simplymeasured.com/rs/simplymeasured2/images/CompleteGuidetoTwitterAnalyticsSimplyMeasured.pdf`

Figure 4.14: People tagging wrong handles for complaint redressal.

connecting link between victims and spammers, thus bringing down the spammers' monetization infrastructure. However, it is difficult to create one, because there are little identifiable features associated with a phone number as there are with OSNs like landing page, some special characters, domain typo-squatting, etc. Therefore, user suspension can come to rescue. From this research we established that the link between a phone number and the spammer account is crucial. Thus, one can focus on removing malicious users from user communities sharing the same phone number. In this network of user accounts, some users would already be suspended by OSNs. The labels can be recursively propagated to other unknown nodes from the known suspended nodes using several graph-based algorithms like Page Rank. Bringing down the spammers propagating phone numbers would disintegrate the entire campaign.

There exist some services, like Truecaller [21] and FTC's do-not-call complaint dataset [10], which collect information about phone numbers that spammers use to call victims (*incoming spam communication*). In this work, however, we demonstrated that spammers advertise their phone numbers across OSNs, so that victims would call them instead (*outgoing spam communication*). We found the overlap between our collected phone numbers with FTC (0.001%) and Truecaller (0.4%) databases to be minimal. It is, therefore, imperative that solutions also be built on outgoing spam communication.

Measuring Impact using Honeypots In this work, we focused on using friends and followers of the user as a metric to measure the impact; it might not capture the actual victims who fell for those campaigns. As an alternative approach, we ran two campaigns from the dataset on Facebook, using phone numbers under our control to find out how many people call these phone numbers. The text for campaigns was taken from the campaign text, and a number was picked from Twilio [16], a service that allows calls to be made over the Internet. We used it to create a framework where calls made to our Twilio number were recorded. Since we did not have an extensive network which a spammer would possess, we instead relied on Facebook Ads to propagate our campaign. We made automated posts every day where the content was changed while using the same phone number. We believe this is a potential way to measure the impact of campaigns, already have a framework

[16]https://www.twilio.com/

in place, and plan to explore in future.

In this chapter, we uncovered the threat landscape through a large-scale study of cross-platform spam campaigns that abuse phone numbers across online social networks. By examining campaigns running across OSNs, we showed that Twitter could suspend ∼93% more accounts spreading spam as compared to Facebook. Therefore, sharing intelligence about spam user accounts across OSNs can aid in spam detection; ∼35K victims and 8.8M USD could be saved based on exploratory analysis of our 6 months data.

Chapter 5

Guess Who's Calling: Questioning the Trustworthiness of Caller ID Applications

In this chapter, we focus on exploiting vulnerabilities found in caller ID applications that is being seen as a trusted source of information for flagging fraudulent / spam calls. Through an online survey study of 426 Amazon MTurk participants, we show that there has been a decline in the trust in the telephony channel, as 56.3% people no longer trust it (Studies in 2006 revealed trust in telephony channel over other digital channels [181]). Because of the rise in fraudulent calls, users tend not to pick up calls from unknown phone numbers (callers without additional information). For this reason, many users are moving to crowdsourced caller ID applications like Truecaller, Facebook's Hello, etc. We also found that users trust the information provided by these caller ID applications. However, such applications cannot be trusted; we demonstrate the ease with which the integrity of data provided by such applications can be compromised by inserting fake information during user registration. As the information is not verified, it is fairly easy to add accounts with false information. The success rate of vishing attacks can be increased by making it more personalized and targeted using the information from social networking sites. We also suggest some plausible defense mechanisms against attacks that exploit caller ID applications.

5.1 Introduction

With the growing communication via social media, texting and e-mail, people might find telephone communication less attractive than its new-media counterparts, however, it is still viewed as a trusted means for communication by both, people and organizations. People tend to give out

personal information like bank account number, address or date of birth over voice calls to financial organizations' representative (e.g., call center of a bank) even when the caller is unknown.

With the advent of cheap Voice over IP (VoIP) calls, phishing attacks have evolved from web-based attacks to targeted telephone-based attacks, called as voice phishing (vishing) [26] conducted by exploiting the trust in voice communication. Since scammers can't ensure whether their e-mail is being read by the victim, easily accessible phone calls helps them to guarantee it. Moving from spam e-mail messages, scammers are making phone calls, which are more accessible. They seek new methods to explore the vulnerabilities in online system, like smartphone applications, bank applications, and other financial transaction systems to breach privacy. In vishing, scammers either call or lure people to call phone numbers controlled by them, and try to collect personal information about the owner of a phone number for the purpose of committing fraud [7, 24]. In recent vishing attacks on Holiday Inn hotels, scammers hacked their phone lines and targeted Bank of America customers by calling them [4]. Attackers, posing as bank officials asked people to enter their credit card details. In an another phone scam, by spoofing the IRS (Internal Revenue Service), scammers asked people to make immediate payments for their taxes, costing consumers $15.5 million [13]. In Microsoft tech support scam, caller lured consumers in believing that their computer is infected with a malware [14], with the intent to defraud them by offering fake repair services. These calls are sometimes scam calls that try to obtain individuals' credentials by exploiting the trust associated with voice communication.

One of the reason why such vishing attacks are successful is due to the lack of intelligence about the calling phone number itself. To avoid falling a victim to such vishing attacks and to know more about the incoming phone number, cloud-based caller identification services are emerging to help in getting additional information about the caller. Millions of people are using such applications, namely Whoscall Caller ID and Block [31], Truecaller [21], Contactive [5], Facebook's Hello [9], and Whitepages Caller ID and Block [29]. In general, these applications allow an individual to register using his / her phone number and help in identifying the caller by showing the information (like name) from their respective databases. Caller ID applications also gather information from social networking sites to collect more information about the caller. These applications increase their visibility and expand their databases via "address book sync" feature, where all contacts in the address book are uploaded to the respective applications' databases.

In this chapter, we explore the integrity of the information provided by such crowd-sourced caller ID applications and show how scammers can exploit them to target their victims by conducting targeted social engineering / vishing attacks. Specifically, a) scammers can register a phone number (controlled by him / her) as a trusted bank / company / organization in which a user is interested in or is dealing with. b) Spoof one of the already registered phone numbers with the caller ID applications and call victims such that the call appears to come from a real entity. Abuse of such applications can lead to several negative implications; ranging from misleading political campaigns

to financial frauds and telemarketing, since people put a lot of trust on the information provided by these applications. The attacks that abuse such caller ID applications can be classified into three major categories:

- **High-impact attacks:** This includes the plausibility of influencing mass movements, e.g., political campaigns by registering the phone number as a particular party and calling millions of people to convince them to vote in their favor. On the other hand, reputation of a party can be undermined by misusing it's name; creating a fake profile with their name on caller ID applications, making calls in their name, and spreading false information to discredit the opposition candidate. Fake election calls have been observed in the past [3].
- **Medium-impact attacks:** Scammers can imitate bank officials and deceive people in giving out their personal information like SSN, bank account number, credit card number etc. Moreover, scammers can impersonate as a software company representative and try to install malware in the name of installing software updates e.g., Microsoft tech support scam [14].
- **Low-impact attacks:** Personalized calls can increase the success of these attacks. Scammers can extract money by making calls for charitable reasons. By exploiting the sentiments of people, they can lure them to pay handsome sums of money in the name of charity. Telemarketing fraud calls can be made to obtain personal information in the name of selling gift items or fake lottery prizes.

The above mentioned attacks can be made more targeted and personalized by aggregating information about the victim from social networking sites and using them to craft an attack. We make the following contributions:

- With a survey conducted with 426 Amazon MTurk participants, we show the users' high level of trust in caller ID applications. A large fraction of participants (315 out of 426) relied on the information provided by caller ID applications while receiving calls. The primary reason for majority of the participants (313 out of 426) to install caller ID applications was it's functionality to provide information about the caller. This shows the inherent trust of people in caller ID applications.
- We explore the feasibility of targeted vishing attacks exploiting caller ID applications and discuss some of the possible attack scenarios. We demonstrate the plausibility of adding noise in these applications, compromising the integrity of the information stored in their databases. We focused on five applications viz., Truecaller, Facebook's Hello, Contactive, Whitepages Caller ID and Block, and Whoscall.

- We demonstrate the feasibility of our targeted vishing attacks by collecting information from one of the caller ID applications, Truecaller, and leveraging more personal information from Facebook. Using Truecaller, we extracted information for 1,779,764 Indian phone numbers. Further, we collected attributes like gender, date of birth, work / school / employer details about the owner of the phone number publicly available from Facebook. This can be used to increase the success rate of attacks by making them more targeted.

5.2 Trust in Caller ID Applications

In this section, we first explore how much trust users have in today's telephony channel and caller ID applications. Caller ID applications are designed to aid users in identifying callers by collating information from their crowd-sourced databases and other social networking sites. During registration, users enter their details and the information is stored in caller ID applications' databases. Using address book sync feature, friend's (contacts) information is also collected and stored in their databases. Table 5.1 shows details of top 5 free caller ID applications used world-wide. It shows the success of these applications in terms of their adoption world-wide.

We answer the following three research questions.

1. RQ1: How trustworthy is the telephony channel?
2. RQ2: How trustworthy is the information provided by caller ID applications?
3. RQ3: What is the reason behind the huge success of these caller ID applications?

To answer these questions, we conducted an online survey by recruiting participants through Amazon's Mechanical Turk (MTurk), a crowd-sourcing platform used to conduct human intelligence tasks [2]. Participation in this study was restricted to only those who were above 18 years of age and using or have used caller ID applications. Participants who completed the survey were paid \$0.15. All the conducted experiments were approved by the Institutional Review Board (IRB). Data collected from the participants was anonymized and protected according to the procedures described in the corresponding IRB submission documents.

Each participant was shown two screenshots; a) call appearing from an unknown phone number with no name or any kind of information from caller ID applications, and b) call coming from HDFC bank as shown by a caller ID application (see Figure 5.3). To understand the trust of participants in both telephony channel and caller ID applications, we asked if they would pick the call in each scenario. We believe that the decision is based on trust in the particular medium.

Table 5.1: Popular caller ID applications adopted world-wide, available free of cost.

Application	**User base**	**Social networks integration**	**Mobile Platform**	**Address book sync feature**	**Countries**	**Start (Year)**
Truecaller	100 million	Facebook, Twitter, LinkedIn	All	Yes	World-wide, popular in India	2013
Whitepages	50 million	Facebook, LinkedIn	Android, iOS	Yes	World-wide	2013
CIA	45 million	Facebook, LinkedIn	All	Yes	Worldwide, popular in US, Austria, India, Nigeria, Morocco and Turkey	2009
Contactive	Unknown (600 million numbers)	Facebook, Skype, Twitter, Google-Plus, LinkedIn, WhatsApp	Android	Yes	World-wide	2013
Facebook Hello	Unknown	Facebook	Android	Yes	USA, Brazil, and Nigeria	2015
Whoscall	Unknown (600 million numbers)	None	Android, Windows, iOS	Yes	World-wide, popular in Taiwan, Japan, HongKong, Korea, USA, Brazil, and Malaysia	2010

5.2.1 Survey Results

There were a total of 510 participants who completed the survey, however, to ensure the responses of the participants are consistent, we repeated a set of questions. We discarded the participants with inconsistent responses to the same question and were left with 426 participants. Demographics of participants are shown in Table 5.2.

Table 5.2: Demographics of survey participants.

Variable	Count (N = 426)
Age	18 - 30 years (238) 31 - 50 years (161) 51 - 65 years (25) Above 65 years (2)
Gender	Male (242) Female (184)

We found a significant difference in participants' responses when they were asked "do you feel

comfortable in picking up the call even if it is coming from an unknown phone number?" as only 141 participants agreed (agree and strongly agree) to pick the call (see Figure 5.1), whereas 240 participants disagreed (disagree and strongly disagree) to pick the call (see Figure 5.2). This means that people are losing trust in telephony channel and do not prefer answering calls unless some information about the caller is known beforehand.

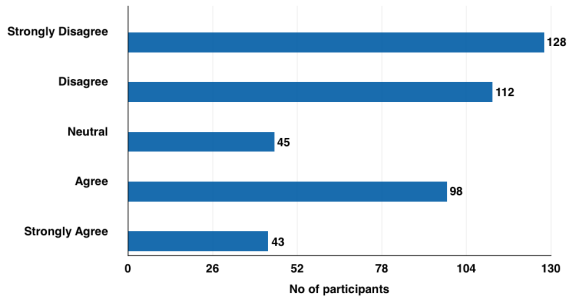


Figure 5.1: Survey response showing a decline in the trust in telephony channel as majority of the participants disagreed to pick the call coming from an unknown phone number.

However, we saw that majority of the participants (315 out of 426) trusted the call showing HDFC bank as the caller (see Figure 5.2), since they believed the information caller ID application provided about the caller to be correct.

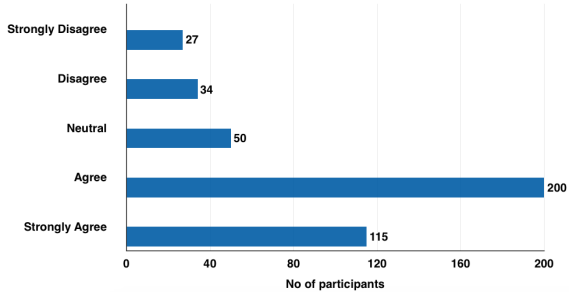


Figure 5.2: Survey response showing increasing trust in caller ID applications as majority of the participants agreed to pick the call from HDFC bank, as they believed the information provided by caller ID applications to be correct.

We use paired t-test to test the significance level (subject to 95% confidence level) of the difference in picking up a call in case of an unknown phone number and when some information is associated with a phone number. Our null hypothesis is that participants would react in the same way (either pick or not pick the call) in both the scenario. The alternative hypothesis is that participants would behave differently. We found that there is a higher likelihood of picking up the calls when there is a name associated with it or information is provided by caller ID applications (paired T-test, p-value < 0.001). This explains the trust of participants in caller ID applications than telephony channel.

Based on the survey results, we also found that majority of the participants installed the caller ID applications to identify the caller, followed by the ability given by these caller ID applications to report phone numbers as spam. This shows that caller ID applications are known to be a credible source of information to identify an incoming caller.

5.3 Launching Vishing Attacks Undermining the Integrity of Caller ID applications

This section briefly outlines how an attacker can compromise the integrity of the information provided by various caller ID applications.

5.3.1 Fake Registration

An attacker can introduce noise into caller ID applications fairly easily, thus compromising the integrity of the information provided by them to launch vishing attacks. Associating an identity with a phone number increases the trust of an individual and likelihood to pick a call. Since caller ID applications do not have a mechanism for verification of the users' details, and rely on the information provided by the user while registering, it is easy for an attacker to abuse this trust. For example, an attacker can register as multiple fake banks on caller ID applications. For registration, he needs a smartphone device with working phone connection. It is a manual process where a short SMS code is sent from caller ID applications to verify the phone number. Since the number of banks are limited, it is not difficult for the attacker to do this manually. Attacker does not know the the bank of all these victims, therefore, the attacker can target a large user population to achieve a good success rate. He can generate millions of automated VoIP calls at low cost, as shown in the past [88].

To make the attack look more authentic, fake social media profiles can be created and linked to caller ID applications while registering an account on it. Further, fetching details about the victim (person to be called) from social networking sites can increase the success of such attacks. Since vishing attacks are already known to be successful [7,13], we believe success rate of targeted attacks would be higher. Figure 5.3 shows fake registration as HDFC bank by exploiting one of caller ID applications, Truecaller. We only use HDFC Bank as an example, but any name / entity can be used.

We performed similar exploits on various caller ID applications and Figure 5.4 shows incoming call showing fake HDFC bank on various caller ID applications; Truecaller, Whitepages Caller ID and Block, Line's Whoscall Caller ID and Block, and Contactive.

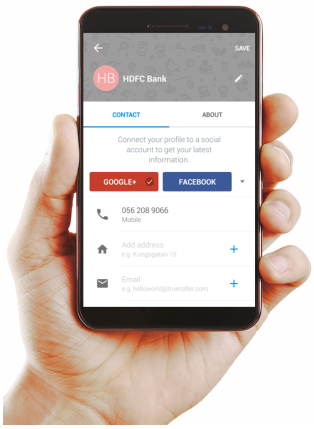


Figure 5.3: Fake registration on Truecaller as HDFC bank to trick victims.

5.3.2 Caller ID Spoofing

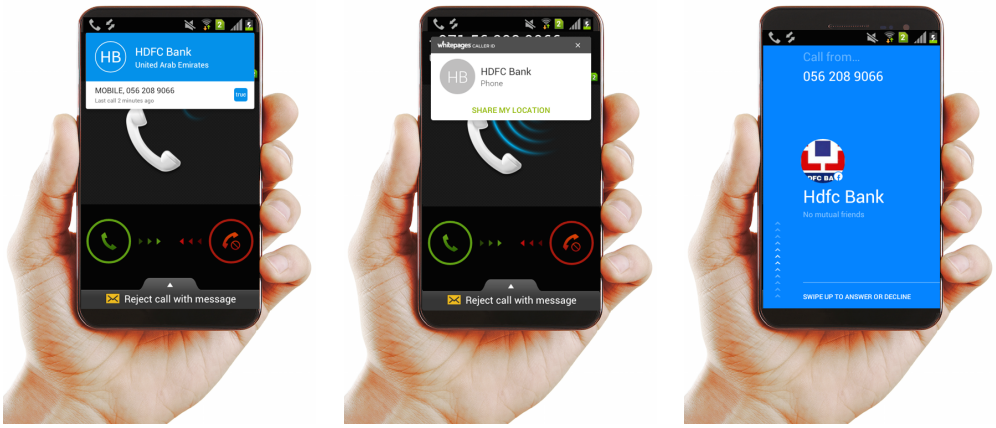
Another form of attack uses caller ID spoofing which can be carried out by imitating already registered phone numbers on a caller ID application or other phone numbers whose details were uploaded by the caller ID application using address book sync feature. As the user must have entered some details about him / her while registering, it makes him / her a more likely target, than an unknown phone number flashing on the screen. To identify all potential victims whose numbers can be spoofed, in Section 5.3.3, we present our results by aggregating all possible victims using address book sync feature of caller ID applications.

5.3.3 Case Study - Truecaller

In this section, we present a case study using one of the caller ID applications, Truecaller, to demonstrate the feasibility and scalability of proposed vishing attacks. We used Truecaller as an example, but any other application can be used to determine this information. The data collection procedure is divided into two phases: a) collecting information from Truecaller about random phone numbers, and b) aggregating more information from social networking application to make the attack targeted and personalized. Note that we neither perform actual attacks nor intend to abuse the caller ID application. Our goal is to provide a proof of concept demonstration for our proposed attacks.

Exploiting Caller ID applications

We first generate a large pool of random phone numbers. To obtain information about potential victims, we use Truecaller, an application that enables searching contact information based on a



(a) Truecaller (b) Whitepages Caller ID & Block (c) Facebook's Hello

(d) Whoscall-Caller ID and Block (e) Contactive

Figure 5.4: Incoming call showing fake HDFC bank (example in our case) on various caller ID applications.

phone number [21]. It is used to identify incoming callers and block unwanted calls. It is a global crowd-sourced phone directory that keeps data of millions of people around the globe. Truecaller also maintains data from social networking sites and correlates this information to create a large dataset for people who register on it. When users registers himself on Truecaller, it syncs all the contacts (friends) of the owner of the phone number using the address book sync feature. The 'search' endpoint of Truecaller application provides details like: name, address, phone number, country, Twitter ID, e-mail, Facebook ID, Twitter photo URL, and photo URL of an individual. However, the private information for each user is obtained according to the privacy settings of users. First and the last name was obtained for all the users on Truecaller. We automated the whole process of fetching information about phone numbers from Truecaller. We extracted the registration ID from the network packet sent when searching a random phone number on Truecaller's iPhone application.

Multiple instances of the process were initiated, on a 2.5 GHz Intel i5 processor, 4GB RAM at the rate of 3000 requests / min. We enumerated through a list of 35,595,280 random Indian phone numbers and found information for 1,779,764 phone numbers on Truecaller.

Aggregating Social Attributes

After collecting basic information from Truecaller, attackers can extract additional information about the owners of these phone numbers from a social networking website to launch targeted vishing attacks. We used Facebook as the social networking platform to obtain information about a user, as it is the largest social networking site used by people world-wide for day-to-day interactions [1]. Truecaller aggregates data from various social networking websites and provides a link of the public profile image of the victim on Facebook. We extracted Facebook ID from these links to collect more information about the victim on Facebook. We use Facebook's Graph API to extract public details about the person; gender, relationship status, birthday, work details, school details, hometown, and public feed [8]. Even though normal access token from Facebook does not provide these details, we were able to fetch the information using a never-expiring mobile OAuth token obtained from iPhone's Facebook application. [1] The information can be used to make the attack more targeted and personalized. For example, likes / interests can be extracted from public feed using text clustering algorithms and targeted vishing calls can be initiated selling particular brands.

To obtain social attributes of the user, we could extract Facebook ID's for 133,983 users. Using Facebook Graph API, we obtained following details for these users: gender (121,924), relationship status (57,965), work details (99,081), school information (129,962), employer details (116,564), birthday (10,455), and hometown (101,027). The collated information can be used to increase the success rate of spear vishing attacks.

5.4 Discussion

In this chapter, we showed a novel targeted vishing attack that can be carried out by compromising the integrity of caller ID applications. Although the phone number itself is verified, several caller ID applications do not check validity of other information provided to these applications during registration, which is what the user actually rely on. For example, a malicious user can associate the name of a legitimate bank to impersonate bank officials and trick people into giving out their personal information like bank account number, credit card number etc. The success rate of vishing attacks can be increased by making them more personalized and targeted using information collected from other sources. Even though people trust caller ID applications as the source of identifying spam calls, such applications are not effective and a better solution to identify phone spam should be in

[1] Recently, Facebook patched this bug and mobile tokens do not give away all the public information.

place. A solution which is resistant to attackers' manipulation can be serviced as a good phone reputation service, which is presented in the next chapter.

As mentioned, there are some challenges associated with caller ID applications. For instance, we observed that Whoscall caller ID application took 24 hours to update their database. Another application Contactive, did not automatically update the database for a new number. We first had to make a call with that phone number to another phone and consequently (manual) add the fake information. Once this is done, the information was synced with it's database and further calls with the registered phone number showed the fake details. However, we are not sure at this point of time, whether this is a feature by design or is a bug in the application. In any case, this can be easily bypassed by an attacker. Given that caller ID applications are well trusted by users, it is imperative that the information is verified and credible. We highlight some recommendations to caller ID applications that can inherit to maintain information integrity.

5.4.1 Recommendations to Caller ID Applications

There is also a necessity to ensure the integrity of the information provided by caller ID applications, as people rely heavily on them to know about the incoming call and trust the information provided by these services.

- **Verification:** One of the biggest challenge that caller ID applications have to face is to implement verification of the information provided by users. Currently, at the time of registration, only phone numbers are verified, and neither the entity behind these phone numbers nor the details of owners of these phone numbers is verified. Caller ID applications can check the integrity of specific business organizations with appropriate authority, listing them as verified users, and routinely scan for any malicious activity in these accounts. Similar techniques can be implemented by caller ID applications to maintain the integrity of the information stored in their databases.
- **Expanding user base:** Additional information can be provided about the caller / the owner of the phone number. For instance, applications can record the timestamp when the account was registered and call frequency patterns. These details can be provided to the user so that he / she can make an informed decision about the caller. In addition, social information about the caller can be displayed, like number of mutual friends, presence on social networks etc. Caller ID applications can design several metrics like social rank based on the information aggregated across social networks. If the same name appears across multiple networks and the user is found to be active, he / she can be assigned a higher score than a passive user.
- **Delinking information:** Caller ID applications like Truecaller serve as a reservoir of information, by collating information from multiple sources. The data from different sources

should not be aggregated and stored at the same place, it serves as a goldmine reservoir. Some parts of the information can be even encrypted to ensure unnecessary information leakage.

Apart from this, an effective defense technique to educate users about privacy implications of using such platforms. To combat the abuse, there have been services in place, for instance, CNAM lookup databases [2] for landlines numbers and Secure Telephone Identity Revisited (STIR) working group [3] that aim to authorize the calling party to use a particular phone number. Some services have initiated defense in this direction; WhatsApp incorporated spam blocker feature as a first step in this direction [28], though their effectiveness need to be studied.

[2] http://www.voip-info.org/wiki/view/CNAM
[3] https://datatracker.ietf.org/wg/stir/charter/

Chapter 6

Collective Classification of Spam Campaigners on Twitter: A Hierarchical Meta-Path Based Approach

This chapter presents our phone reputation service, a phone blacklist that flags spam phone numbers deriving intelligence from Online Social Networks. To this end, we collected information (tweets, user meta-data, etc.) about 3,370 campaigns spread by 670,251 users. We model the Twitter dataset as a heterogeneous network by leveraging various interconnections between different types of nodes present in the dataset. Once spammers are detected, this intelligence is extrapolated to flag spam phone numbers. As discussed in Chapter 5, a robust solution is the one which is resistant to attackers' manipulation; in this chapter we compare how our algorithm performs well in comparison to OSN based state-of-the-art solutions that can be tampered by malicious entities.

6.1 Introduction

Online Social Networks (OSNs) are becoming more and more popular in the recent years, used by millions of users. As a result, OSNs are being abused by spam campaigners to carry out phishing and spam attacks [87]. While attacks carried using URLs [67, 85, 87, 179, 195] has been extensively explored in the literature, attacks via a new action token, i.e., a *phone number* is mostly unexplored. Traditionally, spammers have been exploiting telephony system in carrying out social engineering attacks either by calling victims or sending SMS [193]. Recently, spammers have started abusing OSNs where they float phone numbers controlled by them. Besides exploiting trust associated with a phone number, spammers save efforts in reaching out their victims themselves.

We aim to detect spam campaigners (*aka*, spammers) spreading spam campaigns using phone num-

bers on Twitter. We here define *spammers* as user accounts that use phone numbers to aggressively promote products, disseminate pornography, entice victims for lotteries and discounts, or simply mislead victims by making false promises. Discovering the correspondence between the spammer accounts and the resources (such as URL or phone number) used for spam activities is a crucial task. As the phone numbers are being propagated by spammers, and their monetization revenue starts once people call them, it is fair to assume that these phone numbers would be under their control. As an added advantage of this, if we can identify the spammer accounts in Twitter and bring them down, the entire campaign would get disintegrated. To identify spammers, we model the Twitter dataset as a heterogeneous graph where there are different connections between heterogeneous type of entities: users, campaigns, and the action tokens (phone number or URL) as shown in Figure 6.1.

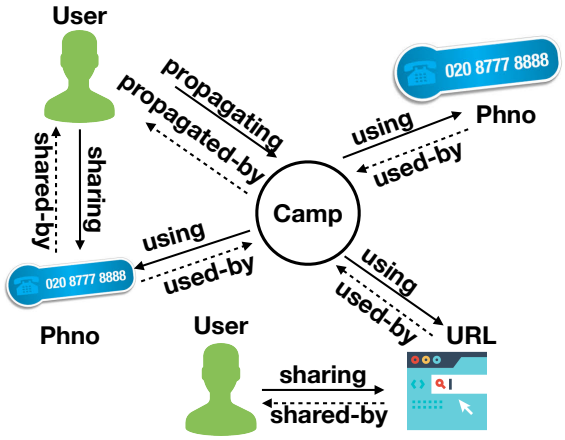


Figure 6.1: Twitter modeled as a heterogeneous network.

Heterogeneous networks have been proposed for data representation in a variety of datasets like path similarity in scholar data [176], link prediction in social network data [114], etc. Objects of different types and links carry different semantics. For instance, a phone number being a more stable resource would help in connecting user accounts over an extended period. Physical identity verification is required to purchase phone numbers, while only e-mail verification is sufficient to purchase domains. Studying similarity between a pair of nodes keeping the heterogeneous nature of the network helps in distinguishing the semantics of different types of paths connecting these two nodes. To distinguish the semantics among paths connecting two nodes, we introduce a *meta-path* based similarity framework for nodes of the same type in the heterogeneous network. A meta-path is a sequence of relations between node types, which defines a new composite relation between its starting type and ending type. It provides a powerful mechanism to classify objects sharing similar semantics appropriately.

The problem of identifying spammers on Twitter that use phone numbers is useful in many aspects. Attacks using phone numbers and URLs are different in some aspects: in URL based spam, the

campaign propagates and spreads on the same medium, i.e., OSNs, while in case of phone number based spam, the attacking medium is a telephone and the propagating medium is OSNs. As a result, it is challenging for OSN service providers to track down the accounts spreading these spam campaigns. In addition, there is no meta-data available for phone numbers, unlike URLs where landing page information, length of URLs, obfuscation, etc. can be checked. Perhaps due to the challenges associated with finding spam phone numbers, there have been several attacks and financial losses caused by the phone-based attacks [139]. Using the collective classification approach proposed in the chapter, Twitter will be able to find potential spammers and suspend the accounts, thereby restricting phone number based spam campaigns.

In this work, we use the *collective classification* approach that exploits the dependencies of a group of linked nodes where some class labels are known and labels diffuse to other unknown nodes in the network. In our case, the known nodes are the already suspended Twitter users that were propagating campaigns with phone numbers. Here, we propose *Hierarchical Meta-Path Score* (*HMPS*), a simple yet effective similarity measure between a pair of nodes in the heterogeneous network. We first build campaign-specific hierarchical trees from the large heterogeneous network. We then systematically extract most relevant meta-paths from a pool of meta-paths linking various heterogeneous nodes.

We collected tweets and other meta-data information of users from April-October, 2016, and identified $3,370$ campaigns, containing $670,251$ users (Section 6.2). Each tweet carries a phone number. We consider user accounts suspended by Twitter as ground-truth spammers. However, due to the lack of enough training samples per campaign, we introduce a novel *feedback-based active learning mechanism* that uses a SVM-based one-class classifier for each campaign. Over multiple iterations, it keeps accumulating evidences from different campaigns to enrich the training set for each campaign. This, in turn, enhances the prediction performance of individual classifiers. The process terminates when there is no chance of finding the label of the unknown users across iterations.

We compare our model with three state-of-the-art baselines used for spam detection (Section 6.5.3). We design various experimental setup to perform a thorough evaluation of our proposed method. We observe that our model outperforms the best baseline method by achieving 44.8%, 16.7%, 6.9%, 67.3% higher performance in terms of accuracy, precision, F1-score and AUC (Section 6.5.3). We further demonstrate how / why one-class classifier (Section 6.5.5), active learning (Section 6.5.6) and feedback-based learning (Section 6.5.7) are better than 2-class classifier, general learning and other oversampling method, respectively. Moreover, we conduct a case study and present an intuitive justification why our method is superior to the other methods (Section 6.5.4).

6.2 Dataset

We collected tweets containing phone numbers from Twitter based on an exhaustive list of 400 keywords via Twitter streaming API. We chose Twitter due to easy availability of data. The data was collected from April - October, 2016. Since we intended to detect campaigns around phone numbers, the keywords we chose were specific to phone number such as 'call', 'SMS', 'WA', 'ring' etc. We accumulated $\sim$ 22 million tweets, each of which containing at least one phone number. The reason behind collecting only tweets containing phone numbers is that they are found to be a stable resource, i.e., spammers use them for a long period due to attached cost. Moreover, the phone numbers are known to help in forming better user communities [68], which is the basis of the approach adopted in this work.

Campaign identification: We define a *campaign* as a group of similar posts shared by a set of users propagating multiple phone numbers. A phone number could be a part of multiple campaigns; however, in this work, we restrict the phone number to be part of a single campaign (since our campaign detection approach is text-based, we want the campaigns to be coherent). Note that, multiple phone numbers could be a part of a single campaign. The detailed approach for campaign identification is shown in Figure 6.2 using a toy example for three phone numbers as described below:

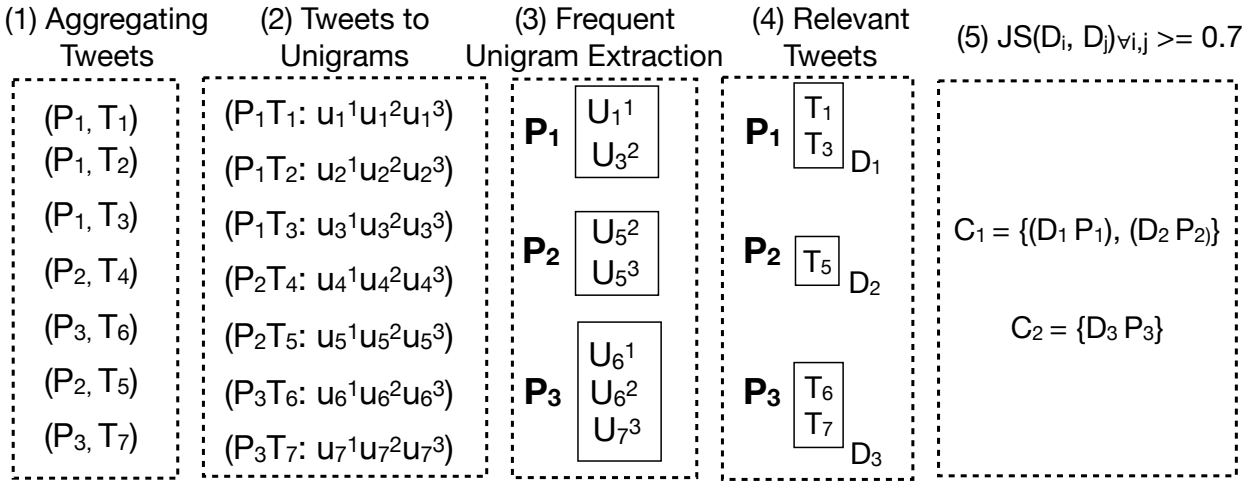


Figure 6.2: A schematic diagram of the framework for campaign identification (notation: P: a phone number, U: a unigram, T: a tweet represented by a set of unigrams, D: a document consisting of a set of similar tweets, and C: a campaign containing a document and its associated phone number).

Step 1: Aggregating tweets. For every phone number, we aggregate all the tweets containing that phone number in a set. We do not find a single tweet containing two phone numbers in our dataset. This implies that every phone number P_i has a set of unique tweets represented as

$T_1, T_2, T_3, \cdots$. In Figure 6.2, P_1 is associated with $\{T_1, T_2, T_3\}$.

Step 2: Tweets to unigrams. We extract unigrams from tweets. Each tweet T_i is now represented as $\{U_i^1, U_i^2, U_i^3, \cdots\}$. In Figure 6.2, T_1 is represented as $\{U_1^1, U_1^2, U_1^3\}$. While calculating unigrams, the author removed stopwords, URLs, and other special characters like '-', '/', ',', '.', '+', '", '=' that was present in the text. In addition, phone numbers or any other digit was not considered as a unigram.

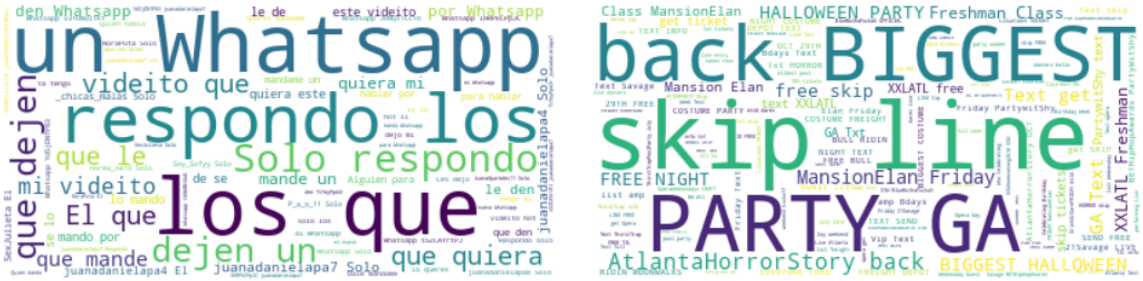
Step 3: Extracting frequent unigram. We aggregate all tweets containing a certain phone number and extract top 30 unigrams that frequently appear in these tweets. This set of unigrams characterizes the document associated with the phone number. In Figure 6.2, the set $\{U_1^1, U_3^2\}$ represents the document associated with P_1.

Step 4: Selecting relevant tweets. From the set of tweets associated with a certain phone number, we choose those which have at least 5 unigrams common with the set of 30 unigrams representing the phone number. In Figure 6.2, we only choose T_1 and T_3 to form document D_1 for P_1 (note, in this example we only match at least one unigram in each tweet instead of 5 to be qualify as a part of the document).

Step 5: Jaccard similarity to find campaigns. Once we form the document corresponding to a phone number, we use Jaccard coefficient to find similarity between two documents and combine them as part of the same campaign. If the Jaccard coefficient is greater than 0.7 (experimentally calculated, as corresponding Silhouette score is 0.8), the documents are merged and thus the corresponding phone numbers become part of a single campaign. In Figure 6.2, D_1 and D_2 are merged together and form campaign C_1.

Using this approach, we identify $22,390$ campaigns in our dataset. These account for $10,962,350$ tweets posted by $670,257$ users, containing $26,610$ unique phone numbers, and $893,808$ URLs. For collective classification to identify campaign-specific spammers, we need to have a set of labeled users. Therefore, we check the user accounts already suspended by Twitter. This process consists of a bulk query to Twitter's API with the profile ID of the account. Twitter redirects to `http://twitter.com/suspended`, and returns 'error 404' in case the user account is suspended. Since Twitter suspension algorithm can have a delay in suspension, we made this query 6 months after the data collection. We find $5,593$ user accounts to be already suspended by Twitter. These accounts are taken later as the training set to perform spam classification (see Section 6.4). Note that for further analysis, we take campaigns that have at least one suspended user – $3,370$ out of $22,390$ campaigns ($670,251$ user accounts) are seen to observe such behavior. We also observe 21% users to be part of multiple campaigns (see Figure 6.7(b)). Figure 6.3 shows the word cloud of top 2 campaigns containing the highest number of suspended users – first one is a Spanish campaign (Figure 6.3(a)) where people requested others to send WhatsApp invitation for receiving adult and pornographic videos. The second campaign (Figure 6.3(b)) offers people reservations for parties and clubs at

discounted rates.



(a) Spanish campaign sending porn videos on WhatsApp.

(b) Party reservation campaign offering discounted rate tickets.

Figure 6.3: Word cloud of top two campaigns containing maximum suspended users.

6.3 Heterogeneous Information Networks (HIN)

We model the entire Twitter data as a heterogeneous information network (HIN). A HIN is a special type of information network which contains multiple types of users or links between users [176]. Mathematically, it is defined as follows:

Definition 6.3.1 Heterogeneous network. *A large number of OSNs are heterogeneous in nature, involving diverse relationship between the nodes. For example, local area networks (LANs) that connect Microsoft Windows and Linux based personal computers with Apple Macintosh computers are heterogeneous. Heterogeneous network is represented as a graph, $G = \{V, E, T\}$ in which each node $v \in V$ and each link $e \in E$ are associated with their mapping functions:*

$$\phi(v) : V \rightarrow T_V$$

$$\phi(e) : E \rightarrow T_E$$

respectively. $T_V \in T$ and $T_E \in T$ denote the sets of users and edge types.

Our heterogeneous network contains different types of nodes such as users, campaign, URLs, and phone numbers; edges connecting two nodes represent different types of relationships (see Figure 6.1). A user is linked to a campaign by `promoting` or `promoted-by` relation; a user is linked to a phone-number by `sharing` or `shared-by` relation; a campaign is linked to a phone-number by `using` or `used-by` relations. Two users can be connected via different paths viz. `user-phone-user`, `user-url-user`, `user-phone-url-user`, and so on. Formally, these paths are called *meta-paths*, as defined below.

Definition 6.3.2 Meta-path. *A meta-path $\Pi_{1.....k}$ is a path defined on the graph of network schema $T_G = (U, R)$, and is denoted in the form of $U_1 \xrightarrow{R_1} U_2 \xrightarrow{R_2} U_3..... \xrightarrow{R_k} U_{k+1}$ which defines a composite relation $R = R_1 \circ R_2 \circ .... \circ R_l$ between type U_1 and U_{k+1}, where $\circ$ denotes the composition operator on relations. In our context, $U \in$ {user, campaign, phone number, URL} and $R \in$ {sharing, promoting, using}.*

The length of a meta-path Π is the number of relations that exist in Π – e.g., `user-phone-user` is a 2-length meta-path between a pair of users, while a 3-length meta-path instance between two users is `user-phone-URL-user`. Figure 6.4 depicts some example meta-paths in our heterogeneous network. For instance, a user participating in a campaign and sharing a phone number can be represented by a 2-length meta-path `User-Camp-Phno`.

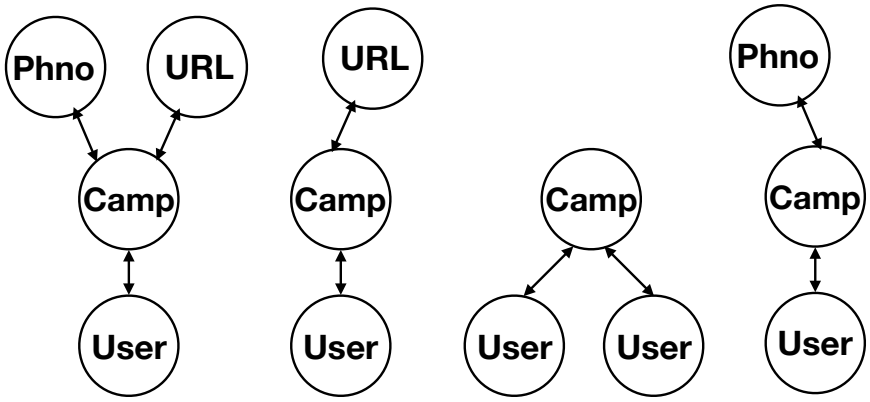


Figure 6.4: Examples of different meta-paths used in the thesis to find HMPS.

Given a user-specific meta-path, $\Pi = U_1, U_2, ...U_t$, similarity measures can be defined for a pair of users $x \in U_1$ and $y \in U_2$ according to the path instances between them following the meta-path. Previous research has shown that including redundant meta-paths (i.e., a smaller meta-path that can be a part of a longer meta-path) in the collective classification may inject noise in the feature space, which can lead to over-fitting [113]. To minimize the risk, it is advisable to extract meta-paths that cannot be further disintegrated to shorter meta-paths. The major challenge in dealing with meta-paths is to find *all and only relevant* meta-paths. Sun et al. [176] showed that finding all possible meta-paths and picking the most relevant out of them is an NP-hard problem, and therefore many greedy approaches have been proposed to find relevant meta-paths [137]. To the best of our knowledge, *this is the first work towards modeling Twitter as a heterogeneous network for spam campaigner detection.* Therefore, there is no prior work suggesting possible and relevant meta-paths for our heterogeneous network. To deal with these challenges, we propose a simple yet efficient concept, called ***Hierarchical Meta-Path Scores*** (***HMPS***) to find similarity between a pair of users by picking shortest and relevant meta-paths (restricted to length 4) which can be used

to calculate similarity between nodes. [1] We also impose an additional constraint on the meta-path selection - we consider meta-paths between two users that *only contain* a campaign, a phone number, or a URL as the intermediate node (i.e., no other intermediate user nodes are allowed between a given pair of users).

6.4 Proposed Methodology

In this section, we describe the overall proposed methodology for collectively classifying users as spammers on Twitter (see Figure 6.5).

Why collective classification? Collective classification refers to the combined classification of nodes based on correlations between unknown and known labels [166]. Given the labels of the instances in training set $Tr \subset All$, the task of collective classification in HIN is to infer the labels of the testing set ($Te = All - Tr$). We address collective classification problem using HMPS to find users (unknown labels) that are *similar* to spammers (known labels). In individual classification, nodes are classified individually without taking into account their interdependencies via the underlying network structure. However, in our heterogeneous networks, nodes are connected by same phone number or URL. Therefore, we employ collective classification approach. It has been shown to achieve better accuracy compared to independent classification [166].

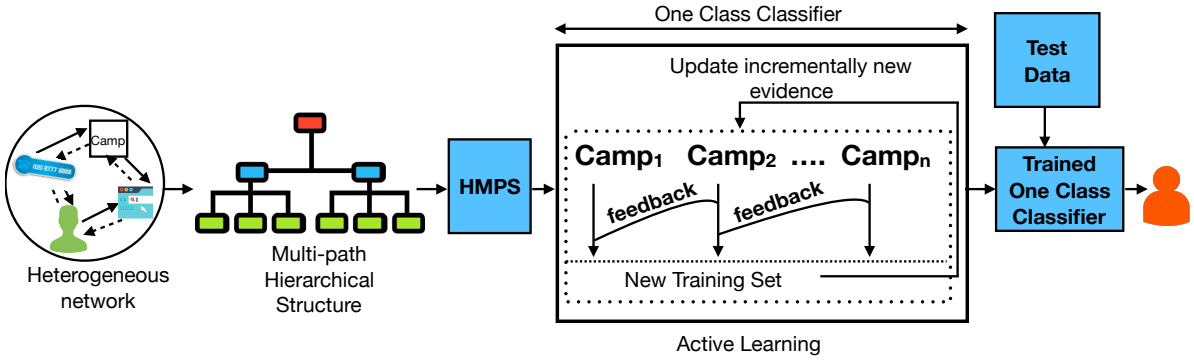


Figure 6.5: Proposed collective classification framework to detect spammers on Twitter.

6.4.1 Hierarchical Meta-Path Scores (HMPS)

After identifying campaigns (see Section 6.2), the next step is to measure HMPS for a user (Algorithm 1) to find the similarity of the user with other known spammers (suspended users). To this end, we propose an additive similarity score for a user with respect to all the spammers in

[1] We experimented with meta-paths of length more than 4. The results were not that encouraging compared to the time it takes to extract long-length meta-paths.

that particular campaign. Although there are several other similarity measures available, they are biased towards underlying network structure and prior information about relevant meta-paths. For instance, PathSim [176] only works for symmetric relations, HeteSim [168] relies on the relevance of a single meta-path. Forward Stagewise Path Generation (FSPG) [137] generates the set of most relevant meta-paths under a given regression model, which is validated by a human expert. However, in the context of Twitter being modeled as a HIN, the relevant meta-paths are not known. Therefore, it is computationally intractable to find the relevance of a meta-path.

This motivates us to propose a novel meta-path based similarity measure, called *Hierarchical Meta-Path Scores* (*HMPS*) that captures the similarity between two users based on the function of distance through which they can be reached.

HIN to hierarchical structure: To measure HMPS, we model the Twitter heterogeneous network in the form of a multi-path hierarchical structure as shown in Figure 6.6. In this structure, nodes on a meta-path are connected with their Least Common Ancestor (LCA) node. LCA node for users is taken as a phone number or URL, and subsequently, campaign node is taken as the LCA node for a phone number / URL. The purpose of LCA node is to limit the range of operations that can be applied across two related nodes. We choose such a structure because if two users share the same phone number or URL for promoting campaigns, they should be more similar rather than two users who do not share any common phone number or URL but are still part of a single campaign. The intuition behind HMPS is that if two users are strongly connected to each other, the distance between them in the hierarchical structure would be less.

The *similarity score* between two entities x and y is a real number, computed by a function F of the similarity scores for each meta-path of a set $\Theta : \Psi(x, y \mid \Theta) = F(max\{\phi(x, y \mid \Pi_i) \mid 1 \leqslant i \leqslant p)\})$, where $\phi(x, y \mid \Pi_i)$ is a similarity score between x and y given meta-path $\Pi_i, \Theta = \Pi_1, ..., \Pi_p$, and F is the maximum similarity score over the 'p' meta-paths. Then the HMPS of an entity x is defined as: $HMPS(x) = \sum_{y \in S} \Psi(x, y)$, where S is the set of spammers in the campaign where x belongs to.

For every user, HMPS is calculated with respect to each spammer (suspended user) in the campaign, and the scores are finally added, as shown in Algorithm 1. Following are the weights used for each edge in the hierarchical structure.

- **W($User_i$, $Phone_j$):** This is the weight of the edge connecting a user and a phone number, and is measured as the ratio of tweets propagated by $User_i$ containing $Phone_j$ over all the tweets propagated by $User_i$.
- **W($User_i, URL_j$):** This is the weight of the edge connecting a user and a URL, and is measured as the ratio of tweets propagated by $User_i$ containing URL_j over all the tweets propagated by $User_i$.

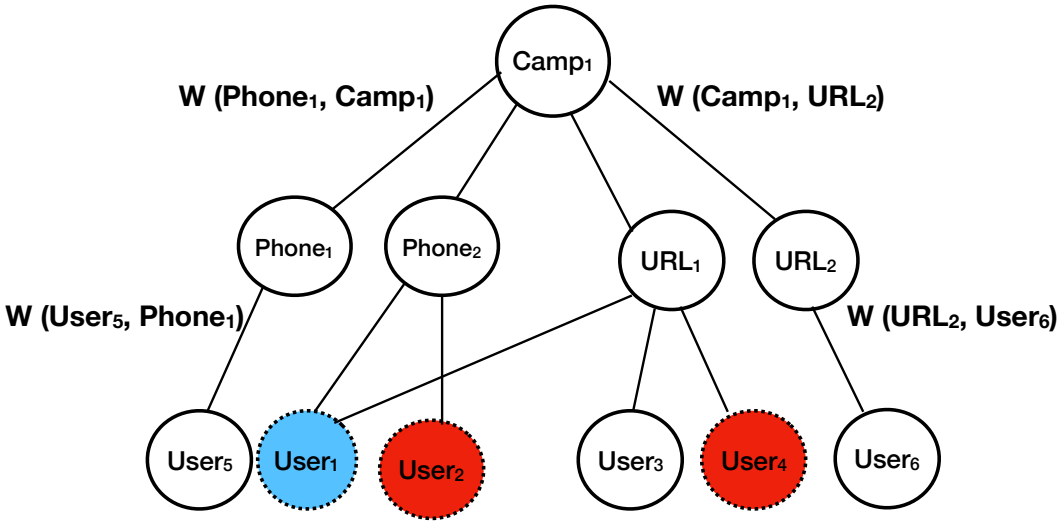


Figure 6.6: A hierarchal structure to measure HMPS of users. Users with red color are known spammers.

- **W**$(Camp_i, Phone_j)$**:** This is the weight of the edge connecting a campaign and a phone number, and is measured as the ratio of tweets containing $Phone_j$ in $Camp_i$ over all the tweets containing phone numbers in $Camp_i$.
- **W**$(Camp_i, URL_j)$**:** This is the weight of the edge connecting a campaign and a URL, and is measured as the ratio of tweets containing URL_j in $Camp_i$ over all the tweets containing URLs in $Camp_i$.

Let us assume that we want to calculate the HMPS for $User_1$ (unknown) shown in Figure 6.6. The campaign contains two suspended users, $User_2$ and $User_4$. So the HMPS score of $User_1$ is calculated w.r.t. $User_2$ and $User_4$ as follows:

- Weight between $User_1$ and $User_2$, W_1: $W\ (User_1, Phone_2)\ .\ W\ (User_2, Phone_2)$
- Weight between $User_1$ and $User_4$, W_2: maximum score calculated for 2 possible meta-paths, i.e., $\texttt{User}_1\texttt{-URL}_1\texttt{-User}_4$ and $\texttt{User}_1\texttt{-Phno}_2\texttt{-Camp}_1\texttt{-URL}_1\texttt{-User}_4$; W_2 = max (W $(User_1, URL_1)$ $.W\ (User_4, URL_1)$, $W\ (User_1, Phone_2)\ .\ W\ (Camp_1, Phone_2)\ .\ W\ (Camp_1, URL_1)\ .\ W\ (User_4, URL_1)$)
- The final HMPS of $User_1$, HMPS $(User_1)$= W_1 + W_2.

Note that in order to measure the HMPS for each user from the hierarchical structure, we build the hierarchical structure for individual campaigns separately instead of combining all the campaigns due to the following two reasons: (i) it is computationally expensive to find meta-paths for all

Algorithm 1 HMPS for Collective Classification

for $Camp_i \in Campaigns$ **do**
  S = Set of known spammers in $Camp_i$ ($m = |S|$); U = Set of unknown users in $Camp_i$; n = Total number of users in $Camp_i$
  $score_i \leftarrow \sum_{j=1}^{m} \text{HMPS}(U_i, S_j, Camp_i) \forall i \in [1, n]$
procedure HMPS(u, s, $camp$)
  $res = 0$
  for $i \in Parent(u)$ **do** ▷ $Parent(u)$ = Immediate antecedent of u
    for $j \in Parent(u)$ **do**
      if $i == j$ **then** ▷ $W(s, j)$=weight of the edge $\langle s, j \rangle$ in the hierarchical structure
        if $W(u, i).W(s, j) > res$ **then**
          $res \leftarrow W(u, i).W(s, j)$
      else
        if $W(u, i).W(s, j).W(i, camp).W(j, camp) > res$ **then**
          $res \leftarrow W(u, i).W(s, j).W(i, camp).W(j, camp)$
  return res

the connections of users across campaigns from a large hierarchical structure, and (ii) HMPS is an absolute value; global HMPS can result in wrong labeling. Specifically, if a spammer (S) has HMPS value X in campaign C_1 and other unknown user (U) has same value X in another campaign C_2, then U will be wrongly labeled as a spammer. It might not be a spammer based on HMPS calculated within that campaign. Researchers have used follower graph network to find new spammers, however, with huge number of unique user nodes (670, 257), aggregating the entire follower graph was challenging.

6.4.2 Active Learning with Feedback

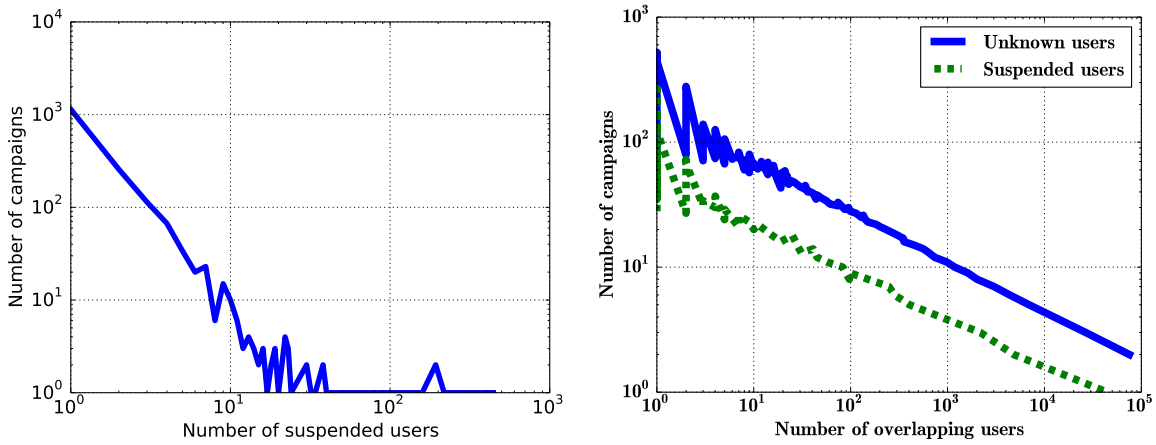
As we consider only those campaigns which contain more than one suspended user (spammer), the classes (spammers and non-spammers) present in our dataset would be highly imbalanced. Exiting research has shown that ***one-class classification* (*OCC*)** achieves much better performance than two-class classification if: (i) there is highly imbalanced dataset [156] and the target class is prevalent in the training set, (ii) the unknown instances do not belong to any known class, or (iii) the unknown instances are difficult to be categorized into a known class due to several reasons such as lack of annotators, lack of enough evidences etc.

OCC is trained only on the target class (which is spam in our case), and its task is to define a classification boundary around the target class, such that it accepts as many instances as possible from the target class, while it minimizes the chance of accepting the outlier instances. In OCC, since only one side of the boundary can be determined, it is hard to decide from just one-class how tightly the boundary should fit in each of the directions around the data. It is also hard to decide

which features should be used to find the best separation of the target and outlier class instances.

Learning with feedback: We would like to reiterate that we picked individual campaigns and not the entire dataset together since the HMPS local to a campaign helps in finding similar users better (see Section 6.4.1). Each campaign is associated with a supervised classifier (one-class classifier in our case). Out of $3,370$ campaigns in the dataset that have at least one suspended user, not all campaigns have sufficient training samples to train the models, as shown in Figure 6.7(a). However, the process of human annotation to enrich the training set can be costly. To reduce the effort of human labeling, one can obtain meaningful shreds of evidence from some external sources and incorporate them into the training set. For instance, in ensemble learning, one can leverage the output class of unknown objects obtained from one classifier and feed them into the other classifiers. This might be related to *active learning*, where given a pool of unlabeled data, one can try to select a set of training examples actively to reach a minimum classification error.

Since individual campaigns may not have significant training instances, we propose an *active learning approach with feedback* to collect cues about unknown users from multiple campaigns to enlarge the individual training set associated with each campaign-specific model. We further notice that campaigns have significant user overlap – 21% users belong to multiple campaigns (see Figure 6.7(b) for the distribution of overlapping users). Presence of user overlap further motivates us to incorporate the feedback-based model as follows.



(a) Number of suspended users per campaign.

(b) Overlapping users in campaings.

Figure 6.7: Distribution of the (a) suspended and (b) overlapping users (users belonging to multiple campaigns) in our dataset. The number of suspended users per campaign is less. Therefore, to increase the training samples, overlapping users are picked for human annotation.

Let us assume that user u is classified as a spammer by a classifier (associated with a campaign say, Cam_i) with high confidence. If u is also a part of some other campaigns (say, Cam_j) where the class of u is unknown, we assign u to the training set of Cam_j along with its class as a spammer.

In this way, we keep increasing the size of the training set of individual classifiers (see Figure 6.8 for a schematic diagram of our proposed feedback-based active learning method). Overall, we perform the following steps:

- An initial set of labeled instances is used for training individual classifiers. Since one-class classifier is used, the training set consists of only the spammers (suspended Twitter accounts). Each campaign-specific classifier is then used to label the unknown users.
- From each set of unknown users labeled by the classifier, we choose a subset of users according to the *selection criterion* (mentioned later). The selected users are then augmented with the training set of other classifiers whose corresponding campaigns also contain these users.
- These steps are iteratively executed for all the campaigns. This constitutes level 1 of the iteration (as shown in Figure 6.8). At the end of this level, we obtain a set of new training set for each classifier.
- In the next level, the new training set is introduced to the classifier and used to predict the class of the rest of the unknown users. This constitutes level 2 of the iteration. The above process convergences once we obtain no more labeled user from the current level to be augmented further with the training set of any classifier in the next level.

Selection criterion: It is important to decide a selection criterion to choose a subset of users from the output of the classifiers; inappropriate criterion might inject noise in the training set that will propagate throughout succeeding levels. We propose the following criterion for selecting users:

> Given (a) a one-class classifier C, represented by the function $f(x)$ which, for an instance x, provides the distance of x from the classification boundary, and (b) X, a set of unlabeled instances, we take the maximum distance among all the training samples from the decision boundary, $T^c_{max} = \max_{x \in X} f(x)$. Now, from the unknown set X_u which are labeled by C, we choose those instances X'_u such that $\forall x \in X'_u : f(x) \geq T^c_{max}$. Note that the threshold T^c_{max} is specific to a campaign.

6.5 Performance of Supervised Machine Learning Algorithms

In this section, we start by presenting the baseline methods used to compare with our method, followed by a detailed comparative evaluation.

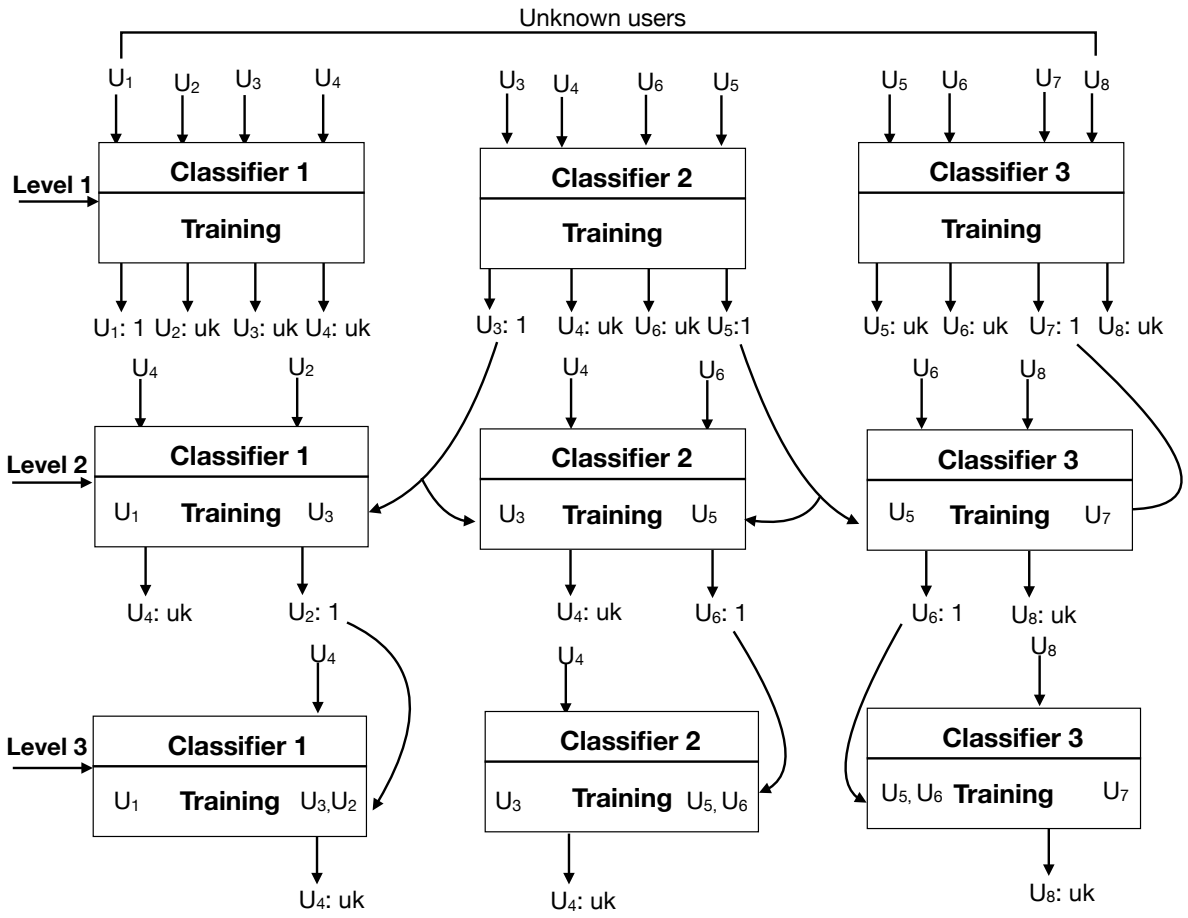


Figure 6.8: A schematic diagram of active learning with feedback amongst campaign-specific classifiers.

6.5.1 Baseline Methods

We compare our method with three state-of-art supervised machine learning methods proposed in the literature for spam detection in general. However, none of them focused on phone number specific spammers whose dynamics are different (as mentioned in Section 6.2), but takes into account several OSN based features that can be accounted for spammer detection. Since we did not obtain the source code, we implemented the methods on our own. Note that all the baselines originally used 2-class classifiers. However, in this chapter, we show the results of the baselines both for one-class and 2-class classifications after suitable hyper-parameter optimization.

Baseline 1: We consider the spam detection method proposed by Benevenuto et al. [51] as our first baseline. They proposed the following OSN-based features (referred as **OSN1**) per user: fraction of tweets with URLs, age of the user account, average number of URLs per tweet, fraction of followers per followee, fraction of tweets the user replied, number of tweets the user replied, number of tweets

the user receives a reply, number of friends and followers, average number of hashtags per tweet. They showed that the SVM-based classifier performs best.

Baseline 2: We consider the method proposed by Khan et al. [107] to segregate spammers from genuine experts on Twitter as our second baseline. They suggested the following features (referred as **OSN2**): authority and hub scores of users in the follower-followee network, fraction of the user's tweets that contain the URLs, average number of URLs in a tweet, average number of URLs per number of words in a tweet of the user, average number of hashtags per number of words in a tweet, and average number of hashtags in a tweet. They showed that Logistic Regression performs best.

Baseline 3: We consider the method proposed by Adewole et al. [39] to detect spam messages and spam user accounts as our third baseline. They proposed the following list of profile and content-based features (referred as **OSN3**): length of the screen name based on characters, the presence or absence of profile location, whether the user includes URL or not in his profile, age of the account in days, number of followers of the user, number of friends / followers of the user, total statuses of the account, number of tweets the user has favorited, indicating presence or absence of profile description, whether the user has not modified the theme of their profile, presence or absence of time zone, whether the account has been verified or not, whether the user has not changed the default profile egg avatar, number of the public lists the user is a member, whether or not the user has enabled the possibility of geo-tagging their tweets, normalized ratio of followers to friends, ratio of the number of follower to friends, ratio of the number of friends to followers, (total, unique, and mean) number of tweets, hashtags, URLs, mentions, favorite count, and retweets, ratio of (hashtags, URLs, mentions, retweets) to total number of tweets, (hashtag, URLs, mention, retweet, tweet-length) deviation, average number of daily tweets, average tweet length, popularity ration, number of duplicate tweets, and maximum value of hashtag frequency. They showed that Random Forest performs best for the classification task.

Note that previous work considered only those campaigns which involve only URLs [39, 51, 107]. In our work, a phone number, being a stable resource, helped in forming campaigns better. Besides, most of the OSN features used in the baselines are easy to evade by spammers, whereas HMPS-based feature is difficult to manipulate.

6.5.2 Experimental Setup

Our proposed classification method is run separately with different features (HMPS, OSN1, OSN2, and OSN3) and their combinations. We use the standard grid-search technique to tune the hyper-parameters. For evaluation, we design two experimental settings:

(i) Setting 1: Our primary goal is to detect user accounts which are suspended by Twitter because they are spam accounts. Therefore, the set of suspended accounts constitutes the ground-truth for

the classifiers. Out of all suspended accounts present in our dataset (mentioned in Section 6.2), we adopt leave-one-out cross-validation technique (due to the very limited number of suspended accounts) and report the average accuracy of the classifiers. Note that in this setting, we use one-class classifier for all the competing methods.

(ii) Setting 2: We believe that our method is capable of detecting those accounts which are spammers, but not suspended by Twitter yet. Therefore, we further invited **human annotators**[2] to annotate some non-suspended accounts as spammers or non-spammers. This will further help us to run the baseline methods which originally used binary classifiers (see Section 6.5.5). Since it is not possible to label all non-suspended users, we adopt a convenient sampling approach. We define user bins according to the number of campaigns the non-suspended users exist (see the distribution in Figure 6.7(b)). Our sampling approach preferentially chooses users who are part of multiple campaigns to maximize the evidence per campaign – the probability of choosing a user belonging to multiple campaigns is higher than that for a user who is a part of a single campaign. Following this approach, we picked 700 users from $3,370$ campaigns. Each user was labeled by three human annotators as spammers or non-spammers, and then the majority vote was considered as the final class. The inter-annotator agreement was 0.82 according to Cohen's kappa measure.

Out of 700 manually annotated accounts, we hold out 20% of the dataset to be used as the test set in Setting 2. We repeat this experiment 50 times and report the average accuracy. Here also, we use one-class classifier for all the competing methods and consider 'spammer' as our target class.

Evaluation metrics: For comparative evaluation, we use the standard information retrieval metrics – Precision, Recall, F1-score, Area under the ROC curve (AUC).

6.5.3 Comparative Evaluation

Table 6.1 shows the performance of the competing methods for both settings. We report the results of our active-learning based one-class classifier with different feature combinations. For setting 1 (leave-one-out), we report the performance w.r.t the *accuracy* (fraction of known spammers identified by the method) and observe that our method performs significantly well with only HMPS feature – it achieves an accuracy of 0.77, outperforming all baseline methods. However, incorporating OSN2 features along with HMPS further enhances 9.1% performance of our classifier, achieving an accuracy of 0.84.

A similar pattern is observed for setting 2 (where human annotators are used). However, here our model with only HMPS turns out to be even stronger classifier, outperforming all others in terms of precision (0.99), F1-score (0.93) and AUC (0.88). Here also, incorporating most of the OSN features with HMPS does not enhance the performance of our method (or sometimes deteriorates

[2]All annotators were security researchers between the age group of 25 - 35 years.

the performance), except ONS2 which seems to be quite competitive. However, baseline 2 seems to be the best method w.r.t recall (0.92); but it significantly sacrifices the performance w.r.t. precision, F1-score, and AUC.

Nevertheless, we consider the following setting as our default method since it outperforms other methods in almost all experimental setup: HMPS + OSN2 + one-class classifier + active learning. Baseline 2 is considered as the best baseline method for reporting purpose.

Table 6.1: Comparative evaluation of the competing methods on two different experimental settings. For all the methods, one-class classifier is used. The colored row shows the performance (P: Precision, R: Recall, F1: F1-score) of our default method. The last row shows the results of our default method *without* active learning (see Section 6.5.6).

Method	**Feature**	**Setting 1**	**Setting 2**			
		Accuracy	**P**	**R**	**F1**	**AUC**
Baseline 1	OSN1	0.62	0.86	0.71	0.77	0.48
Baseline 2	OSN2	0.58	0.84	**0.92**	0.87	0.52
Baseline 3	OSN3	0.62	0.86	0.66	0.74	0.47
Our	HMPS	0.77	**0.99**	0.87	**0.93**	**0.88**
	HMPS + OSN1	0.76	0.89	0.90	0.89	0.72
	HMPS + OSN2	**0.84**	0.98	0.88	**0.93**	0.87
	HMPS + OSN3	0.70	0.88	0.73	0.80	0.59
Our	HMPS + OSN2 - Active Learning	–	0.42	0.98	0.55	0.51

6.5.4 Justification behind superior performance of HMPS

All of the baseline methods rely on the features that can be changed over time. These methods either consider URL attributes (baselines 1 and 3) within the tweets or changes in profile characteristics between a legitimate and spam user account (baselines 2 and 3). Given these specificities, it is easy for a spammer to manipulate these features. In contrast, HMPS relies on the monetization infrastructure (phone numbers) to identify campaigns and spammers. As discussed earlier, we aggregate tweets as part of the same campaign when they use multiple phone numbers wrapped around similar text. As a result, our method is resilient to spammers' manipulation. Furthermore, to understand how HMPS helps in improving the detection of spammers over the baselines, we manually analyze a sample of 'spammers'. Some of the users not identified by baselines 1 and 3 as spammers have a balanced number of friends and followers and a low number of tweets. In addition, users were not using URLs to spread the campaign. Therefore, all URL-based features do not aid in the detection task.

Baseline 2 measures the authority and hub scores based on the tweets with hashtags. As a result, it

wrongly detects some benign users as spammers that were retweeting posts related to (say,) blood donation campaigns. When baseline 2 is combined with HMPS, the false positive rate is reduced since these users are not found in the spammer network.

In addition, HMPS can find spammers that are not suspended by Twitter yet. For instance, Figure 6.9 shows a spammer account that clearly violates the Twitter policy by promoting and posting repeated, pornographic content. [3] Surprisingly, this account has not been suspended by Twitter yet. However, we found similar such accounts suspended by Twitter. Interestingly, our system was able to identify this account as a spammer.

These examples show that HMPS can identify spammers that use phone numbers, which are not detected by the baseline systems and / or Twitter, and is, therefore, more effective in detecting spammers that spread phone numbers to promote campaigns.

(a) Bio of the spammer account claiming itself as a promoter of pornographic content.

(b) Timeline of the spammer mentioned in (a) indicating similar content is being posted repeatedly.

Figure 6.9: An example spammer account that has not been suspended by Twitter yet, but our system could detect it as spammer.

6.5.5 One-class vs. Two-class Classifier

One may argue that the results reported in Table 6.1 may not reflect the original performance of the baseline methods since all the baseline methods originally used 2-class classifiers. Moreover, there

[3] https://support.twitter.com/articles/18311

was no empirical justification for adopting one-class classifier over 2-class classifier. To address these arguments, here we exactly replicate the baseline methods by considering the best 2-class classifier per baseline reported in the respective literature. We choose a balanced dataset of 150 suspended and 150 non-suspended users randomly sampled from our manually labeled dataset (see setting 2 in Section 6.5.2). For comparative evaluation, we consider several state-of-the-art 2-class classifiers (Logistic regression (LR), Latent Dirichlet Allocation (LDA), K-nearest neighbors (KNN), Decision Tree (DT), Naive Bayes (NB), and Support Vector Machine (SVM)) and adopt them into our active learning framework. Table 6.2 shows that none of the baselines and our adopted 2-class classifiers outperform our default one-class classifier (last row of Table 6.2). Our default method is 12.6%, 7.7%, 10.7% and 9.7% higher than the second-ranked method (Decision Tree) in terms of precision, recall, F1-score, and AUC respectively. We compute Wilson score interval to calculate the confidence intervals covering the classification error at 95% likelihood to evaluate the classifiers. The Wilson score interval can be computed by the following equation:

$$confidence\ interval = error + / - const\ * sqrt((error\ * (1 - error))/n) \tag{6.1}$$

Taking the constant value as 1.96 for 95% likelihood, confidence interval for each of the classifier is observed as, LR: [0.14, 0.36], LDA: [0.12, 033], KNN: [0.04, 0.25], DT: [0.02, 0.17], NB: [0.15, 0.37], SVM: [0.11, 0.32], RF: [0.04, 0.20], and HMPS (one-class): [0, 0.03]. This result indicates that one-class classification is always helpful for the application where there is limited labeled data, and the label of most of the instances is unknown.

Table 6.2: Results of 2-class classifiers and comparison with our default one-class classifier. Here, the best 2-class classifiers reported in the papers are considered for the baselines.

Method	**Precision**	**Recall**	**F1-score**	**AUC**
Baseline 1	0.68	0.69	0.65	0.50
Baseline 2	0.47	0.57	0.51	0.50
Baseline 3	0.79	0.78	0.78	0.57
HMPS + 2-class classifier				
LR	0.61	0.58	0.55	0.58
LDA	0.61	0.58	0.55	0.58
KNN	0.75	0.74	0.74	0.74
DT	0.83	0.83	0.83	0.83
NB	0.60	0.58	0.57	0.58
SVM	0.65	0.63	0.62	0.63
RF	0.83	0.82	0.82	0.82
Our default one-class classifier				
HMPS+OSN2	**0.95**	**0.90**	**0.93**	**0.92**

6.5.6 General vs. Active Leaning

We argue that since the number of training samples is not sufficient for individual classifiers, feedback from one classifier to another would increase the training set that in turn enhances the performance of the classifier. To verify our argument, we run our default method without active learning (without feedback) and observe that although recall increases significantly (0.98), it degrades the performance w.r.t. other performance measures – 57.1%, 40.8% and 41.3% degradation of precision, F1-score and AUC respectively (see the last row of Table 6.1). High recall with low precision indicates that most of the unknown users are classified as spammers by the general classifier. It happens due to the limited labeled data, which active learning can efficiently handle.

6.5.7 Feedback vs. Oversampling

Since the size of the training set is small for each campaign-specific classifier, we use feedback across campaigns to increase the training set. As an alternative, one can also use other state-of-the-art oversampling techniques such as SMOTE [62]. Here, we adapt SMOTE for increasing the size of the training data (i.e., target class: spammers). The training set is oversampled by taking each training sample and introducing synthetic examples along the line segments joining any / all k neighbors of the training sample. Depending on the amount of oversampling required, k-nearest neighbors are randomly chosen. Table 6.3 shows the results for different values of oversampling ratio, i.e., the fraction of training set taken as the number of synthetic samples. In addition, we perform the oversampling technique before dividing the data into training and validation to ensure that the information from the training set is used in building the classifier. Table 6.3 shows that even after varying the ratio for oversampling, none of the cases can achieve the accuracy obtained from our feedback-based learning approach. This indicates that our feedback-based learning strategy is superior to the other oversampling strategy.

After detecting spammers in the network, this intelligence is extrapolated to find spam phone numbers. A phone number is marked as spam if more than 50% users propagating that phone number is marked as a spammer. The input to the system, called SpamDoctor [4] is a phone number which needs to be checked (see Figure 6.10(a)). In case the phone number is not present in the database, the spam and confidence score is set to 'NA', as shown in Figure 6.10(b).

Figure 6.11 shows the system implementation which detects whether the phone number is spam or not spam. *Confidence* score is a metric that tells what fraction of users propagating a particular phone number are spammers i.e..,

$$Confidence_{(X=spam)} = \frac{\text{Users propagating phone number X which are spammers}}{\text{all the users propagating phone number X}}$$

[4] `http://labs.precog.iiitd.edu.in/phonespam/`

Table 6.3: Comparison of our feedback-based learning approach with standard oversampling approach (SMOTE). The term 'Ratio' indicates the fraction of training set taken as the number of synthetic samples generated by the oversampling technique.

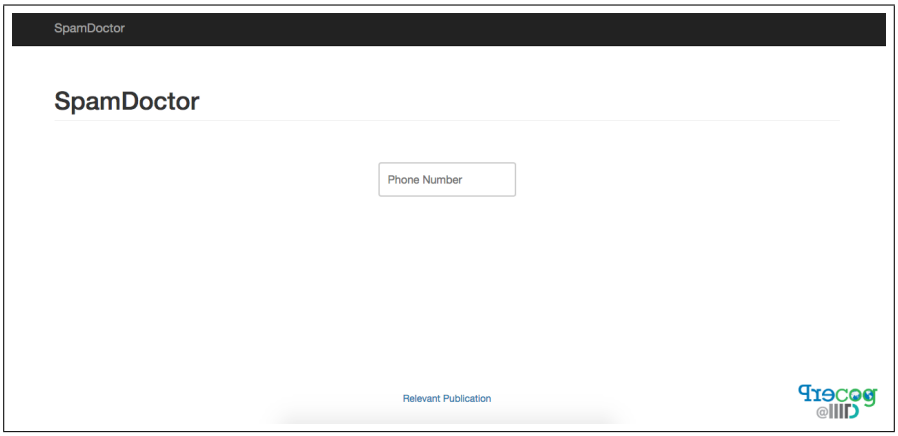
Oversampling + default one-class classifier				
	Precision	Recall	F1-Score	AUC
Ratio = 0.20	0.90	0.64	0.64	0.59
Ratio = 0.30	0.88	0.74	0.74	0.63
Ratio = 0.50	0.81	0.71	0.68	0.58
Ratio = 0.75	0.91	0.68	0.69	0.56
Ratio = 1	0.91	0.68	0.70	0.57
Feedback + default one-class classifier				
	0.95	**0.90**	**0.93**	**0.92**

More the number of spammers propagating a phone number, higher the confidence score. In case the phone number is not spam, confidence score is defined as:

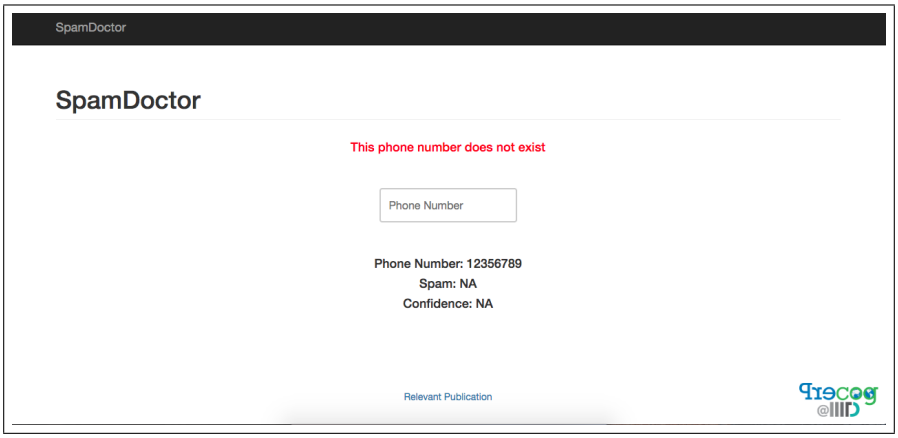
$$Confidence_{(X=not-spam)} = 1 - \frac{\text{Users propagating phone number X which are spammers}}{\text{all the users propagating phone number X}}$$

Here, higher confidence score means lesser users propagating phone number X are spammers.

In this chapter, we detected spammers spreading spam campaigns using phone numbers on Twitter. We modeled Twitter as a heterogeneous network and proposed a collective classification approach that leverages heterogeneous nodes and their interconnections to identify unknown users as spammers. The significant contributions of our method are three-fold: (i) our proposed Hierarchical Meta-Path Score (HMPS) can efficiently measure how close an unknown user is w.r.t other known spammers; (ii) our proposed feedback-based active learning strategy is effective over three other baselines that use 2-class classifiers for spam detection; (iii) in case of small number of training instances, our proposed feedback strategy performs significantly better than other oversampling strategies. Through a case study and human-annotated dataset, we also showed that our method could find spammer accounts that are not suspended by Twitter yet. Finally, we created SpamDoctor, a phone blacklist to flag phone numbers used in spam on OSNs.

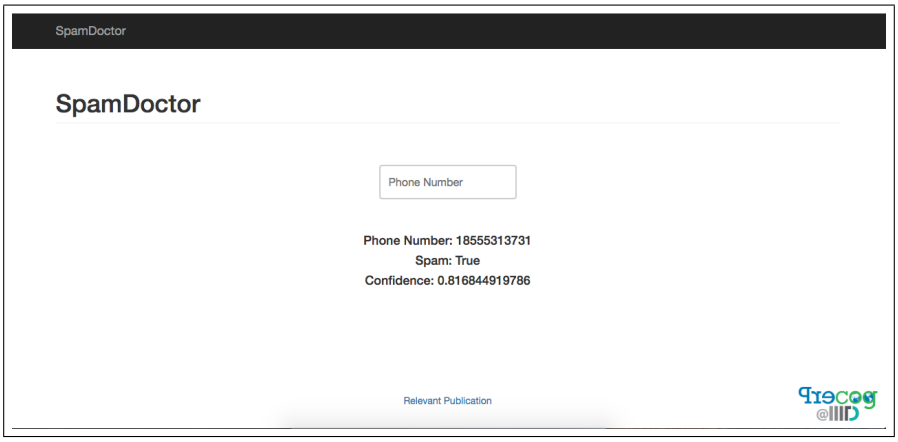


(a) First screen of SpamDoctor where input is the phone number.

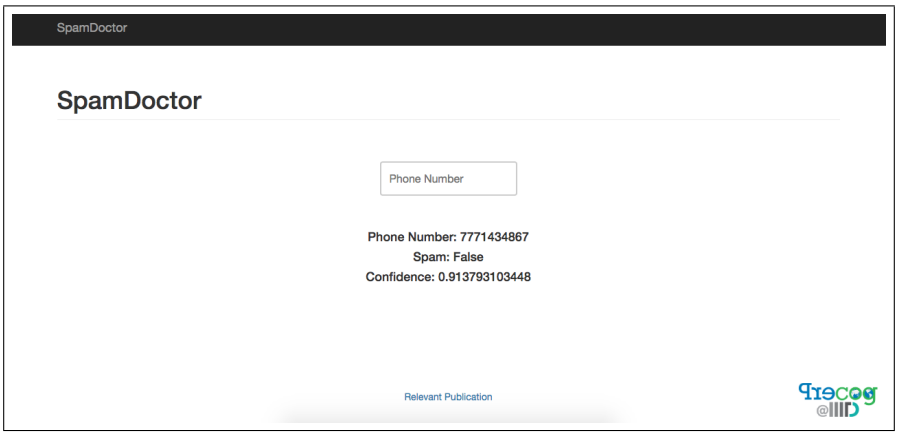


(b) SpamDoctor displays a message if the phone number to be searched does not exist in the database.

Figure 6.10: Screenshots of SpamDoctor.



(a) Phone number marked as spam.



(b) Phone number marked as non-spam.

Figure 6.11: Screenshots of SpamDoctor that labels a phone number as spam or not.

Chapter 7

Conclusions, Limitations, and Future Work

In this chapter, we summarize the various aspects of our work towards identifying and mitigating phone number abuse on Online Social Networks. We selected phone number as the primary action token because it is a trusted and instant form of communication, as discussed in Chapter 1. The goal of this thesis was to study and analyze phone number abuse on Online Social Networks, and propose and evaluate techniques for detecting phone numbers that are part of a spam campaign. Even though several blacklisting services for URLs like VirusTotal, SURBL, Phishtank exist for detecting spam URLs, no such service exists for phone numbers. Section 7.1 presents the summary of our research contributions. The limitations and future directions are discussed in Section 7.4.

7.1 Summary

This thesis aims to bring out methods to detect spammers abusing phone numbers on Online Social Networks. In addition, we built a system / service that detects whether the phone number given as input is good or bad. The research contributions of this thesis are - (1) Building automated framework to evaluate the effectiveness of phone based attacks on OTT, (2) Building automated detection method to identify phone based spam campaigns and the users behind it, (3) Evaluating the trustworthiness of current caller ID services to detect spam calls, (4) Evaluating supervised detection methods to identify spammers and building a robust and effective phone reputation score for phone numbers abused on OSNs. Each contribution is elaborated in further sections.

7.1.1 Understanding phone based attacks on OTT Messaging Applications

Smartphones have fueled a shift in the way we communicate with each other via Instant Messaging. With the convergence of Internet and telephony, new Over-The-Top or OTT messaging applications (e.g., WhatsApp, Viber, WeChat etc.) have emerged as an important means of communication for millions of users. These applications use phone numbers as the only means of authentication and are becoming an attractive medium for attackers to deliver spam and carry out more targeted attacks. In this thesis, we first studied various ways in which spammers can attack OTT messaging application users by leveraging information from OSNs (Chapter 3). We enumerated through a random pool of 1.16 million phone numbers and by exploiting address book syncing feature, we found 255K numbers on OTT messaging applications. We showed how non-targeted, spear, and social phishing attacks can be crafted against the owners of these phone numbers by exploiting cross-application features from multiple applications. We showed the success of these attacks by conducting an online roleplay user study with 314 Amazon MTurk participants. We found that social (69.2%) and spear (54.3%) phishing attacks are more successful than non-targeted phishing attacks (35.5%) on OTT messaging applications. Although similar results were found for other mediums like e-mail, we demonstrated that due to the significantly increased user engagement via OTT messaging applications and ease with which phone numbers allow collection of pertinent information, there is a clear need for better protection of this new medium. We also found that factors like frequently using WhatsApp to interact with large numbers of friends and being deficient in regulating such active usage of the medium, governed victim's vulnerability to fall for phishing attacks on OTT messaging applications. With this experimental study we realised that phone numbers are a trusted source of entity and there is no mechanism in place to label a phone number as good or bad. To devise such a system, sufficient metadata / information about a phone number is required. Since information about a phone number is not available on OTT messaging applications, we moved to richer source of information, i.e., Online Social Networks to assimilate intelligence for a phone number.

7.1.2 Characterizing the threat landscape of phone based attacks on OSNs

To derive intelligence for a phone number, we studied a platform that houses rich source of information viz., Online Social Networks. First, we identified and characterized several spam campaigns abusing phone numbers that are active on OSNs (Chapter 4). We specifically looked at the outgoing phone spam campaigns where attackers advertised phone numbers under their control for users to call them. To expand the reach of such spam campaigns, phone numbers were advertised across multiple platforms like Facebook, Twitter, GooglePlus, Flickr, and YouTube. We collected $\sim 23M$ posts containing $\sim 1.8M$ unique phone numbers from Twitter, Facebook, GooglePlus, Youtube, and Flickr over a period of six months. Clustering these posts helped us in identifying 202 campaigns operating across the globe with Indonesia, United States, India, and United Arab Emirates

being the most prominent originators. We found that even though Indonesian campaigns generate highest volume ($\sim 3.2M$ posts), only 1.6% of the accounts propagating Indonesian campaigns have been suspended so far. By examining campaigns running across multiple OSNs, we discovered that Twitter detects and suspends $\sim 93\%$ more accounts than Facebook. Therefore, sharing intelligence about abuse-related user accounts across OSNs can aid in spam detection. According to our 6 months dataset, around $\sim 35K$ victims and $\sim 8.8M$ could have been saved if intelligence was shared across the OSNs. In addition, we analysed the modus operandi of several campaigns to understand the monetization model of spammers behind such attacks. Finally, we compared the characteristic behavioral difference between the spam and legitimate phone based campaigns and found that spammers work as a strongly connected component, by sharing and mentioning the same phone number over time; lifetime of a spam phone number is shorter than legitimate phone number since spammers replenish the phone pool to avoid detection. The understanding of phone number threat landscape on OSNs helped us in devising solutions to detect bad phone numbers, as discussed in Section 7.1.4.

7.1.3 Investigating the effectiveness of current methods for detecting spammers

Before delving into building a phone reputation system, we looked at current state-of-the-art techniques used in literature to combat URL spam. In addition, we studied tools that give information about unknown incoming phone calls. Trust in the telephony channel has been exploited with a rise in voice phishing (vishing) attacks. Scammers impersonate legitimate organizations and lure people to divulge information like bank account number, credit card details, or other personal information. Often, attackers use caller ID spoofing to suppress their original identity / phone number. Because of the rise in fraudulent calls, users tend not to pick up calls from unknown phone numbers (callers without additional information). For this reason, many users are moving to crowd-sourced caller ID applications like Truecaller, Facebook's Hello, etc. Based on a survey of 426 MTurk participants, we found that users trust caller ID applications (like Truecaller) to assimilate information about incoming caller in an unknown phone call, as described in Chapter 5. These caller ID applications are vulnerable to fake registration and spoofing attacks which make them inefficient in correctly identifying spammers. Since caller ID applications do not verify users' personal information provided during registration, we showed that it is relatively easy to register multiple fake identities using different phone numbers. We also demonstrated how information from caller ID applications and personal attributes from social networking sites can be aggregated to launch targeted vishing attacks. As a proof of concept, we were able to collect information for approximately 1.8 million Indian phone numbers on one of the caller ID applications, Truecaller. Using this, we could aggregate public social attributes from Facebook like gender, work / school / employer details, birthday, etc., which can further be used to craft targeted and personalized vishing attacks, thus increasing the attack's success rate. This experimental study showed that even though users trust caller ID ap-

plications, they are an ineffective spam phone detection system. Further, we explored if supervised machine learning models based on OSN features can be used to identify spammers that use phone numbers. If we are able to find spammers, all the phone numbers used by them would be considered bad w.r.t OSN. However, our experimental results showed that OSN based models are prone to manipulation, therefore, not a reliable solution to identify spammers (Section 6.5). These detection models relied on the features that can be changed over time. They either considered URL attributes within the tweets or changes in profile characteristics between a legitimate and spam user account. Given these specificities, it is easy for a spammer to manipulate these features to evade OSN detection. In contrast, our detection algorithm, HMPS relied on the monetization infrastructure (phone numbers) to identify campaigns and spammers which is prone to such manipulation.

7.1.4 Building a robust phone reputation score for phone numbers on OSNs

To build a robust solution to uncover spammers and bad phone numbers, we modeled OSNs as a heterogeneous network by leveraging various interconnections between different types of nodes present in the dataset. We devised a metric, Hierarchical Meta-Path Score (HMPS) to measure the proximity of an unknown user to the other known pool of spammers, as described in (Chapter 6). We designed a feedback-based active learning strategy and show that it significantly outperformed three state-of-the-art baselines for the task of spam detection. We observed that our method achieved 6.9% and 67.3% higher F1-score and AUC, respectively compared to the best baseline method. To overcome the problem of less training instances for supervised learning, we showed that our proposed feedback strategy achieved 25.6% and 46% higher F1-score and AUC respectively than other oversampling strategies. Finally, we performed a case study to show how our method was capable of detecting those users as spammers who have not been suspended by Twitter yet. Finally, we used spammer metrics to design a phone reputation service, called SpamDoctor that can flag a potential bad phone number; the strength lies in the fact that is not based on temporal features and hence hard to get manipulated.

7.2 List of Publications

- Our work in Chapter 3 is published as:
  Gupta, Srishti, Payas Gupta, Mustaque Ahamad, and Ponnurangam Kumaraguru. “Exploiting phone numbers and cross-application features in targeted mobile attacks." In Proceedings of the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices, pp. 73-82. ACM, 2016.

- Our work in Chapter 4 is published as:
  Gupta, Srishti, Dhruv Kuchhal, Payas Gupta, Mustaque Ahamad, Manish Gupta, and Pon-

nurangam Kumaraguru. “Under the Shadow of Sunshine: Characterizing Spam Campaigns Abusing Phone Numbers Across Online Social Networks." In Proceedings of the 10th ACM Conference on Web Science, pp. 67-76. ACM, 2018.

- Our work in Section 4.5 is under review in Journal of WebScience as:
  Srishti Gupta, Gurpreet Singh Bhatia, Saksham Suri, Dhruv Kuchhal, Payas Gupta, Mustaque Ahamad, Manish Gupta, Ponnurangam Kumaraguru. “Angel or Demon? Characterizing Variations Across Twitter Timeline of Technical Support Campaigners."

- Our work in Chapter 6 is published as:
  Gupta, Srishti, Abhinav Khattar, Arpit Gogia, Ponnurangam Kumaraguru, and Tanmoy Chakraborty. “Collective Classification of Spam Campaigners on Twitter: A Hierarchical Meta-Path Based Approach." In Proceedings of the 2018 World Wide Web Conference on World Wide Web, pp. 529-538. International World Wide Web Conferences Steering Committee, 2018.

- Other publication that is not included in the thesis:
  Gupta, Srishti, and Ponnurangam Kumaraguru. “Emerging phishing trends and effectiveness of the anti-phishing landing page." In Electronic Crime Research (eCrime), 2014 APWG Symposium on, pp. 36-47. IEEE, 2014.

7.3 Phone Spam Mitigation: Suggestions to Users

Even though there is no solution against phone spam, we enumerate a list of techniques that a user can employ to avoid falling trap to phone spam.

- **Changing the phone number:** A crude way to avoid phone spam is changing the existing phone number in use. However, it could be possible that one is inheriting someone else’s phone spam when getting a new phone number. Phone companies tend to recycle numbers; person X’s “new" number might be person Y’s “old" number. If that number has been subjected to any kind of breach, X is essentially getting transferred to a new kind of spam.

- **Installing a call blocker:** It won’t stop the calls but can minimise the frequency. However, as mentioned above, these applications can stop incoming calls but are not effective for outgoing spam communication. The coverage of spam phone numbers in such applications is bleak.

- **Joining the Do Not Call Registry:** This helps in mitigate a very small percentage of the phone spam, but the DNCR only helps mitigate sales calls. Spammers also abuse the personal information given by users while registering to such lists. Further, businesses can easily get around the Registry rules as well. For example, non-profits can still make donation calls or

calls that are in effect sales calls even if they masquerading as though they're for non-profit purposes.

- **Reporting suspected spam numbers:** Crowdsourcing is a potential way for blacklisting services to aggregate data about potential bad phone numbers. Federal Trade Commission (FTC) has a page for reporting spam and scam caller, but, the FTC has a very hard time stopping spoofed numbers, which seem to be comprising the majority of spam calls these days.

7.4 Limitations and Future Work

We take a data-driven approach in this dissertation which suffers from certain limitations:

- **Data Sampling:** We acknowledge that our dataset may contain some bias, as we get only 1% Twitter sample available from the REST API. It can underestimate the spam campaigns observed on Twitter, however, we observed different kind of campaigns abusing phone numbers in our dataset. However, it is extremely challenging to obtain an ideal, unbiased dataset.
- **Limited Campaigns:** We started our data collection from Twitter. We might miss some of the popular campaigns prevalent on other social networks, but, Twitter provides a good sample of public content.
- **Dataset Completeness:** We relied on the data provided by APIs from all the networks. Therefore, we might miss some posts related to the campaigns. Given that scraping is not allowed, using APIs was the only viable option to gather data. Irrespective of this, we have been able to receive quite a significant amount of posts related to phone numbers. Further, if a search parameter is not supported by the API, it is challenging to retrieve posts relevant to a phone number.

The dataset we used for our analysis comprises of only public posts. We were not able to find a way to validate if our dataset is a representative sample of the entire OSN stream. However, to the best of our knowledge, our dataset of 22 million posts containing by 1.9 million distinct phone numbers is one of the biggest datasets of phone campaigns ever analyzed as part of academic research.

SpamDoctor is trained as phone number blacklist; it can not tell if the phone number is good. We would like to develop a mechanism that can whitelist phone numbers based on OSNs' verified accounts; it can act as one-stop service to know information about any phone number that surfaces OSNs. SpamDoctor's evaluation is based on a limited set of labeled data / ground truth. Although, we consciously made an effort to label data that is representative of all the spam domains under

consideration, by randomizing the selection process for manually inspecting the domains, we recognize the need to scale this experiment and plan to do it in the future while adding more capabilities to our system. An intuitive way to attack the problem of automatically identifying spam campaigns using a phone number would be to understand the rationale behind the attack strategy. Given the diversity of spam campaigns, it was hard to identify monetisation model for all the spam campaigns without access to an entity that engages in such actions.

We believe that insights obtained from this thesis can be utilized by researchers and stakeholders to make social media environment safer and more informative. Based on our experience so far, we suggest the following directions:

Address different type of campaigns differently. Our dataset treats spam and scam campaign as a same unit. While monetization is a big factor in scam campaigns, spam campaigns can be targeted to collect personal information only. Further, we noticed some campaigns used URLs in addition to phone numbers to trick users; the modus operandi for such campaigns can be different. However, in this thesis we consider all kinds of campaigns having phone number as an action token as a common entity. Although we achieved a certain level of success, it is possible that modelling content in each of these categories separately might produce more accurate models which can be explored further.

Utilize crowdsourcing to personalize and improve the performance of automated techniques for spam campaign identification. SpamDoctor leverages intelligence from several OSNs to enrich the training phase, however crowdsourced intelligence can play a crucial role in designing and improving accuracy of solutions. We foresee a need for a way to accommodate users' feedback and preferences to enable customization of spam filtering techniques based on these preferences. Training phase could be augmented by user feedback in addition to the HMPS algorithm devised in this thesis. Instead of running SpamDoctor as an API / service, a browser extension or plugin can be created to capture user feedback on OSNs itself. This is likely to make automated tools more widely accepted and thus safeguard a wider audience. Further, SpamDoctor can be extended to other platforms like OTT messaging applications by using overlaying mechanisms to alert users on the fly when they use their smartphones for placing a call.

Explore the impact of images and cross referenced posts in OSN posts. Our analysis revealed presence of images and cross referenced posts in phone based spam campaigns. We call a post cross-referenced if it was posted to OSN X, but contains a URL redirecting to OSN Y. For instance, a Twitter post containing a link 'fb.me/xxxx' which would redirect to a different OSN, Facebook. Spammers either direct victims to existing posts or to another profile which is propagating the same campaign on a different OSN. Building a timeline of campaigns that appear

in several OSNs within small bursts of time can help in augmenting cross-platform intelligence described in this thesis. Further, we noticed presence of images in our OSN post that contain a phone number, both in the (post) text and image. It could be possible that there are certain posts containing a phone number only in the image and not in the post text, which would be missed in our current data collection methodology. Extracting text from images and checking the presence of a phone number in it can be seen as an enhancement to the current data collection methodology. Further, the impact of such images on end users' perception of the campaign is largely unexplored. It would be interesting to study whether success rate of a campaign is influenced by the presence of visual cues.

Investing spammer network and adaptation on OSNs: While this work touches upon the collusion between spammers involved in a spam campaign, it would be interesting to study in detail the entire follower graph of spammers; studying username similarity, creation dates, likes and shares within the community. Spammers usually collude to boost the content and enhance visibility; liking and sharing each other's content helps in expanding the user base. In addition, it will be worthwhile to study how spammers adapt to the suspension by OSNs; whether they become dormant for sometime or do they quickly create new accounts, how do they regulate the volume per account etc.

Bibliography

[1] 91 Leading Social Networks Worldwide. http://www.practicalecommerce.com/articles/86264-91-Leading-Social-Networks-Worldwide. (Accessed on 2018-12-13).

[2] Amazon Mechanical Turk. https://www.mturk.com/mturk/welcome. (Accessed on 2018-12-13).

[3] At least 14 election ridings blitzed with live calls from fake Liberals. http://news.nationalpost.com/news/canada/at-least-14-election-ridings-blitzed-with-live-calls-from-fake-liberals. (Accessed on 2018-12-13).

[4] Bank Phishing Scams. http://krebsonsecurity.com/2015/02/hacked-hotel-phones-fueled-bank-phishing-scams/. (Accessed on 2018-12-13).

[5] Contactive Universal Caller ID. http://contactive.com/. (Accessed on 2018-12-13).

[6] Estimated 24.9m americans lost $8.9b in phone scams as rate of spam calls jumps 22%, according to new report from truecaller. https://globenewswire.com/news-release/2018/04/26/1488339/0/en/Estimated-24-9M-Americans-Lost-8-9B-in-Phone-Scams-as-Rate-of-Spam-Calls-Jumps-22-Accordi
html. (Accessed on 2018-10-13).

[7] Ex-policeman under suspicion of voice phishing. http://koreajoongangdaily.joins.com/news/article/Article.aspx?aid=2997528. (Accessed on 2018-10-13).

[8] Facebook Graph API. https://developers.facebook.com/docs/graph-api. (Accessed on 2018-12-09).

[9] Facebook Hello. http://www.engadget.com/2015/04/22/facebook-hello/. (Accessed on 2018-12-13).

[10] Federal trade commission do-not-call complaints. https://www.ftc.gov/site-information/open-government/data-sets/do-not-call-data. (Accessed on 2018-10-13).

[11] Fetching friends from Graph API. http://stackoverflow.com/questions/11135053/fetching-list-of-friends-in-graph-api-or-fql-appears-to-be-missing-some-friend. (Accessed on 2018-10-13).

[12] Indian complaint board, vashikaran fraud. https://www.complaintboard.in/complaints-reviews/vashikaran-fake-vashikaran-fraud-cheater-money-taker-l149781.html. (Accessed on 2018-10-13).

[13] I.R.S Tech Support Scams. http://www.forbes.com/sites/michaelzakkour/2015/04/14/i-r-s-tax-phone-scam-claims-more-victims-than-ever-as-2015-tax-day-arrives/. (Accessed on 2018-10-13).

[14] Microsoft Tech Support Scams. https://www.microsoft.com/security/online-privacy/avoid-phone-scams.aspx. (Accessed on 2018-12-13).

[15] Over-the-top application (ott). https://www.techopedia.com/definition/29145/over-the-top-application-ott. (Accessed on 2018-12-13).

[16] Over-The-Top Messaging Apps Overtake SMS Messaging. http://mobilemarketingmagazine.com/over-the-top-messaging-overtakes-sms. (Accessed on 2018-10-13).

[17] Price for Vanity Numbers. http://articles.economictimes.indiatimes.com/2007-10-13/news/27675454_1_digit-numbers-mukul-khanna-minimum-price. (Accessed on 2018-10-13).

[18] Silent Phone Numbers cost 35$ a year. http://thenewdaily.com.au/news/2014/02/17/pay-35-silent-phone-number/. (Accessed on 2018-10-13).

[19] SMS Phishing. https://en.wikipedia.org/wiki/SMS_phishing. (Accessed on 2018-10-13).

[20] Tech support complaints, 800notes. https://800notes.com/Phone.aspx/1-800-549-5301/2. (Accessed on 2018-10-13).

[21] Truecaller. https://www.truecaller.com/. (Accessed on 2018-12-13).

[22] Using twitter as a data source: An overview of current social media research tools. http://blogs.lse.ac.uk/impactofsocialsciences/2015/07/10/social-media-research-tools-overview/. (Accessed on 2018-10-13).

[23] Vanity Numbers. http://www.openthemagazine.com/article/real-india/calling-9999999999. (Accessed on 2018-10-13).

[24] Vietnam arrests 24 Chinese, Taiwanese nationals in voice phishing scam. http://www.thanhniennews.com/society/vietnam-arrests-24-chinese-taiwanese-nationals-in-voice-phishing-scam-43743.html. (Accessed on 2018-10-13).

[25] Voice Phishing. https://en.wikipedia.org/wiki/Voice_phishing. (Accessed on 2018-10-13).

[26] Voice Phishing. http://en.wikipedia.org/wiki/Voice_phishing. (Accessed on 2018-10-13).

[27] Whaling? These Scammers Target Big Phish. http://www.scambusters.org/whaling.html. (Accessed on 2018-10-13).

[28] WhatsApp Spam Block Feature. http://www.ibtimes.co.uk/whatsapp-rolls-out-new-spam-blocker-feature-1497715. (Accessed on 2018-10-13).

[29] Whitepages Caller ID and Block. http://www.whitepages.com/caller-id. (Accessed on 2018-12-13).

[30] Who is most likely to use googleplus? http://insight.globalwebindex.net/chart-of-the-day-who-is-most-likely-to-use-google. (Accessed on 2018-10-13).

[31] Whoscall Caller ID and Block. http://whoscall.com/. (Accessed on 2018-12-13).

[32] Why do phishing attacks work better on mobile phones? http://www.welivesecurity.com/2011/01/20/why-do-phishing-attacks-work-better-on-mobile-phones/. (Accessed on 2018-12-13).

[33] Netcraft anti-phishing tool bar. http://toolbar.netcraft.com/, 2004. (Accessed on 2014-03-13).

[34] Phone scams. https://www.consumer.ftc.gov/articles/0076-phone-scams, 2014. (Accessed on 2016-05-20).

[35] Headsup for whatsapp. http://www.adaptivemobile.com/blog/headsup-for-whatsapp, 2015. (Accessed on 2016-01-15).

[36] Malwarebytes lab. https://blog.malwarebytes.com/%20tech-support-scams/, 2016. (Accessed on 2016-05-10).

[37] Number of social media users worldwide from 2010 to 2021 (in billions), 2017.

[38] Bsnl auction for vanity numbers. http://eauction.bsnl.co.in/auction1/index.aspx?id=74, 2018. (Accessed on 2018-03-15).

[39] Kayode Sakariyah Adewole, Nor Badrul Anuar, Amirrudin Kamsin, and Arun Kumar Sangaiah. *SMSAD: a framework for spam message and spam account detection.* Springer, 2017.

[40] Sadia Afroz and Rachel Greenstadt. Phishzoo: An automated web phishing detection approach based on profiling and fuzzy matching. Technical report, Technical Report DU-CS-09-03, Drexel University, 2009.

[41] Ammar Ali Deeb Al-Momani, Tat-Chee Wan, Karim Al-Saedi, Altyeb Altaher, Sureswaran Ramadass, Ahmad Manasrah, Loai Bani Melhiml, and Mohammed Anbar. An online model on evolving phishing e-mail detection and classification method. *Journal of Applied Sciences*, 11(18), 2011.

[42] Hélio Almeida, Dorgival Guedes, Wagner Meira, and Mohammed J Zaki. Is there a best quality metric for graph clusters? In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 44–59. Springer, 2011.

[43] Amit A Amleshwaram, Narasimha Reddy, Sandeep Yadav, Guofei Gu, and Chao Yang. Cats: Characterizing automation of twitter spammers. In *Communication Systems and Networks (COMSNETS), 2013 Fifth International Conference on*, pages 1–10. IEEE, 2013.

[44] David S Anderson, Chris Fleizach, Stefan Savage, and Geoffrey M Voelker. *Spamscatter: Characterizing internet scam hosting infrastructure.* PhD thesis, University of California, San Diego, 2007.

[45] Manos Antonakakis, Roberto Perdisci, David Dagon, Wenke Lee, and Nick Feamster. Building a dynamic reputation system for dns. In *USENIX security symposium*, pages 273–290, 2010.

[46] Spiros Antonatos, Iasonas Polakis, Thanasis Petsas, and Evangelos P Markatos. A systematic characterization of IM threats using honeypots. In *Proceedings of the 17th Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA*, 2010.

[47] Vijay Balasubramaniyan, Mustaque Ahamad, and Haesun Park. CallRank: Combating SPIT Using Call Duration, Social Networks and Global Reputation. In *Conference on Email and Anti-Spam, CEAS*, 2007.

[48] Vijay A Balasubramaniyan, Aamir Poonawalla, Mustaque Ahamad, Michael T Hunter, and Patrick Traynor. PinDr0p: using single-ended audio features to determine call provenance. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 109–120. ACM, 2010.

[49] Marco Balduzzi, Payas Gupta, Lion Gu, Debin Gao, and Mustaque Ahamad. Mobipot: Understanding mobile telephony threats with honeycards. In *Proceedings of the 11th ACM*

SIGSAC Symposium on Information, Computer and Communications Security, ASIA CCS '16, New York, NY, USA, 2016. ACM.

[50] Marco Balduzzi, Christian Platzer, Thorsten Holz, Engin Kirda, Davide Balzarotti, and Christopher Kruegel. Abusing Social Networks for Automated User Profiling. In *Recent Advances in Intrusion Detection*, pages 422–441. Springer, 2010.

[51] Fabricio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgilio Almeida. Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, volume 6, page 12, 2010.

[52] Fabrício Benevenuto, Tiago Rodrigues, Virgílio Almeida, Jussara Almeida, and Marcos Gonçalves. Detecting spammers and content promoters in online video social networks. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 620–627. ACM, 2009.

[53] Andre Bergholz, Jeong Ho Chang, Gerhard Paaß, Frank Reichartz, and Siehyun Strobel. Improved phishing detection using model-based features. In *CEAS*, 2008.

[54] Leyla Bilge, Thorsten Strufe, Davide Balzarotti, and Engin 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*, WWW '09, pages 551–560, 2009.

[55] Aaron Blum, Brad Wardman, Thamar Solorio, and Gary Warner. Lexical feature based phishing url detection using online learning. In *Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security*, pages 54–60. ACM, 2010.

[56] Yazan Boshmaf, Dionysios Logothetis, Georgos Siganos, Jorge Lería, Jose Lorenzo, Matei Ripeanu, and Konstantin Beznosov. Integro: Leveraging victim prediction for robust fake account detection in osns. In *NDSS*, volume 15, pages 8–11, 2015.

[57] Yigang Cai. Phonebook use to filter unwanted telecommunications calls and messages, December 21 2005. US Patent App. 11/314,108.

[58] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, pages 15–15. USENIX Association, 2012.

[59] Qiang Cao, Xiaowei Yang, Jieqi Yu, and Christopher Palow. Uncovering large groups of active malicious accounts in online social networks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 477–488. ACM, 2014.

[60] Juan Miguel Carrascosa, Roberto González, Rubén Cuevas, and Arturo Azcorra. Are trending topics useful for marketing. *Proc. COSN*, 2013.

[61] Martin Casado, Tal Garfinkel, Weidong Cui, Vern Paxson, and Stefan Savage. Opportunistic measurement: Extracting insight from spurious traffic. In *Proc. 4th ACM Workshop on Hot Topics in Networks (Hotnets-IV)*, 2005.

[62] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[63] Yao Cheng, Lingyun Ying, Sibei Jiao, Purui Su, and Dengguo Feng. Bind Your Phone Number with Caution: Automated User Profiling Through Address Book Matching on Smartphone. In *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, pages 335–340. ACM, 2013.

[64] S Chiappetta, Claudio Mazzariello, Roberta Presta, and Simon Pietro Romano. An anomaly-based approach to the analysis of the social behavior of VoIP users. *Computer Networks*, pages 1545–1559, 2013.

[65] Neil Chou, Robert Ledesma, Yuka Teraguchi, and John C Mitchell. Client-side defense against web-based identity theft. In *NDSS*, 2004.

[66] Nicolas Christin, Sally S Yanagihara, and Keisuke Kamataki. Dissecting one click frauds. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 15–26. ACM, 2010.

[67] Zi Chu, Indra Widjaja, and Haining Wang. Detecting social spam campaigns on twitter. In *International Conference on Applied Cryptography and Network Security*, pages 455–472. Springer, 2012.

[68] Andrei Costin, Jelena Isacenkova, Marco Balduzzi, Aurélien Francillon, and Davide Balzarotti. The role of phone numbers in understanding cyber-crime schemes. In *Privacy, Security and Trust (PST), 2013 Eleventh Annual International Conference on*, pages 213–220. IEEE, 2013.

[69] George Danezis and Prateek Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *NDSS*, pages 1–15. San Diego, CA, 2009.

[70] Ram Dantu and Prakash Kolan. Detecting spam in voip networks. In *Steps to Reducing Unwanted Traffic on the Internet Workshop, SRUTI'05*. USENIX Association, 2005.

[71] Clayton Allen Davis, Onur Varol, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. Botornot: A system to evaluate social bots. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 273–274. International World Wide Web Conferences Steering Committee, 2016.

[72] Rachna Dhamija and J Doug Tygar. The battle against phishing: Dynamic security skins. In *Proceedings of the 2005 symposium on Usable privacy and security*, pages 77–88. ACM, 2005.

[73] Rachna Dhamija, J Doug Tygar, and Marti Hearst. Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 581–590. ACM, 2006.

[74] Rachna Dhamija, J Doug Tygar, and Marti Hearst. Why Phishing Works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 581–590. ACM, 2006.

[75] Julie S Downs, Mandy B Holbrook, and Lorrie Faith Cranor. Decision strategies and susceptibility to phishing. In *Proceedings of the second symposium on Usable privacy and security*, pages 79–90. ACM, 2006.

[76] Julie S Downs, Mandy B Holbrook, and Lorrie Faith Cranor. Decision strategies and susceptibility to phishing. In *Proceedings of the second symposium on Usable privacy and security*, pages 79–90. ACM, 2006.

[77] Manuel Egele, Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Compa: Detecting compromised accounts on social networks. In *NDSS*, pages 1–17, 2013.

[78] Manuel Egele, Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Towards detecting compromised accounts on social networks. *IEEE Transactions on Dependable and Secure Computing*, 14(4):447–460, 2017.

[79] Michalis Faloutsos. Detecting malware with graph-based methods: traffic classification, botnets, and facebook scams. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 495–496. ACM, 2013.

[80] Adrienne Porter Felt and David Wagner. Phishing on mobile devices. 2011.

[81] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. The rise of social bots. *Communications of the ACM*, 59(7):96–104, 2016.

[82] Ian Fette, Norman Sadeh, and Anthony Tomasic. Learning to detect phishing emails. In *Proceedings of the 16th international conference on World Wide Web*, pages 649–656. ACM, 2007.

[83] BJ Fogg, Jonathan Marshall, Othman Laraki, Alex Osipovich, Chris Varma, Nicholas Fang, Jyoti Paul, Akshay Rangnekar, John Shon, Preeti Swani, et al. What makes web sites credible?: a report on a large quantitative study. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 61–68. ACM, 2001.

[84] Anthony Y Fu, Liu Wenyin, and Xiaotie Deng. Detecting phishing web pages with visual similarity assessment based on earth mover's distance (emd). *Dependable and Secure Computing, IEEE Transactions on*, 3(4):301–311, 2006.

[85] Hongyu Gao, Jun Hu, Christo Wilson, Zhichun Li, Yan Chen, and Ben Y Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 35–47. ACM, 2010.

[86] Saptarshi Ghosh, Bimal Viswanath, Farshad Kooti, Naveen Kumar Sharma, Gautam Korlam, Fabricio Benevenuto, Niloy Ganguly, and Krishna Phani Gummadi. Understanding and combating link farming in the twitter social network. In *Proceedings of the 21st international conference on World Wide Web*, pages 61–70. ACM, 2012.

[87] Chris Grier, Kurt Thomas, Vern Paxson, and Michael 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.

[88] Slade E Griffin and Casey C Rackley. Vishing. In *Proceedings of the 5th annual conference on Information security curriculum development*, pages 33–35. ACM, 2008.

[89] DJ Guan, Chia-Mei Chen, and Jia-Bin Lin. Anomaly based malicious url detection in instant messaging. In *Proceedings of the Joint Workshop on Information Security (JWIS)*, 2009.

[90] Payas Gupta, Mustaque Ahamad, Jonathan Curtis, Vijay Balasubramaniyan, and Alex Bobotek. M3AAWG Telephony Honeypots: Benefits and Deployment Options. Technical report, 2014.

[91] Payas Gupta, Swapna Gottipati, Jing Jiang, and Debin Gao. Your Love is Public Now: Questioning the Use of Personal Information in Authentication. In *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, ASIA CCS '13, pages 49–60. ACM.

[92] Payas Gupta, Roberto Perdisci, and Mustaque Ahamad. Towards measuring the role of phone numbers in twitter-advertised spam. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pages 285–296. ACM, 2018.

[93] Payas Gupta, Bharat Srinivasan, Vijay Balasubramaniyan, and Mustaque Ahamad. Phoneypot: Data-driven Understanding of Telephony Threats. In *22nd Annual Network and Distributed System Security Symposium, NDSS 2015*, 2015.

[94] Srishti Gupta, Payas Gupta, Mustaque Ahamad, and Ponnurangam Kumaraguru. Exploiting phone numbers and cross-application features in targeted mobile attacks. In *Proceedings of the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices*, pages 73–82. ACM, 2016.

[95] Shuang Hao, Nadeem Ahmed Syed, Nick Feamster, Alexander G Gray, and Sven Krasser. Detecting spammers with snare: Spatio-temporal network-level automatic reputation engine. In *USENIX Security Symposium*, 2009.

[96] Amir Herzberg and Ahmad Gbara. Trustbar: Protecting (even naive) web users from spoofing and phishing attacks. *Computer Science Department Bar Ilan University*, 6, 2004.

[97] Markus Huber, Martin Mulazzani, Edgar Weippl, Gerhard Kitzler, and Sigrun Goluch. Friend-in-the-middle attacks: Exploiting social networking sites for spam. *Internet Computing, IEEE*, 15(3):28–34, 2011.

[98] AHM Rahmatullah Imon and Ali S Hadi. Identification of multiple outliers in logistic regression. *Communications in Statistics-Theory and Methods*, 37(11):1697–1709, 2008.

[99] Smart Insights. Mobile marketing statistics. http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/, July 2015. (Accessed on 2018-10-13).

[100] Adam N Ioinson and Carina B Paine. Self-disclosure, privacy and the Internet. *The Oxford handbook of Internet psychology*, 2007.

[101] Jelena Isacenkova, Olivier Thonnard, Andrei Costin, Aurélien Francillon, and David Balzarotti. Inside the scam jungle: A closer look at 419 scam email operations. *EURASIP Journal on Information Security*, 2014(1):4, 2014.

[102] Tom N Jagatic, Nathaniel A Johnson, Markus Jakobsson, and Filippo Menczer. Social Phishing. *Communications of the ACM*, 50(10):94–100, 2007.

[103] Tom N Jagatic, Nathaniel A Johnson, Markus Jakobsson, and Filippo Menczer. Social phishing. *Communications of the ACM*, 50(10):94–100, 2007.

[104] Markus Jakobsson and Jacob Ratkiewicz. Designing ethical phishing experiments: a study of (rot13) ronl query features. In *Proceedings of the 15th international conference on World Wide Web*, pages 513–522. ACM, 2006.

[105] Jaeyeon Jung and Emil Sit. An empirical study of spam traffic and the use of dns black lists. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 370–375. ACM, 2004.

[106] Rishabh Kaushal, Srishti Chandok, Paridhi Jain, Prateek Dewan, Nalin Gupta, and Ponnurangam Kumaraguru. Nudging nemo: Helping users control linkability across social networks. In *International Conference on Social Informatics*, pages 477–490. Springer, 2017.

[107] U. U. S. Khan, M. Ali, A. Abbas, S. Khan, and A. Zomaya. Segregating spammers and unsolicited bloggers from genuine experts on twitter. *IEEE Transactions on Dependable and Secure Computing*, 2016.

[108] Eunhyun Kim, Kyungwon Park, Hyoungshick Kim, and Jaeseung Song. I've got your number: Harvesting users' personal data via contacts sync for the. In *Information Security Applications: 15th International Workshop, WISA 2014, Jeju Island, Korea, August 25-27, 2014. Revised Selected Papers*, volume 8909, page 55. Springer, 2015.

[109] Hyung-Jong Kim, Myuhng Joo Kim, Yoonjeong Kim, and Hyun Cheol Jeong. DEVS-based modeling of VoIP spam callers' behavior for SPIT level calculation. *Simulation Modelling Practice and Theory*, 17(4):569–584, 2009.

[110] Engin Kirda and Christopher Kruegel. Protecting users against phishing attacks with antiphish. In *Computer Software and Applications Conference, 2005. COMPSAC 2005. 29th Annual International*, volume 1, pages 517–524. IEEE, 2005.

[111] Pranam Kolari, Tim Finin, and Anupam Joshi. SVMs for the blogosphere: Blog identification and splog detection. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 92–99, 2006.

[112] Xiangnan Kong, Bokai Cao, and Philip S Yu. Multi-label classification by mining label and instance correlations from heterogeneous information networks. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622. ACM, 2013.

[113] Xiangnan Kong, Philip S Yu, Ying Ding, and David J Wild. Meta path-based collective classification in heterogeneous information networks. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1567–1571. ACM, 2012.

[114] Xiangnan Kong, Jiawei Zhang, and Philip S Yu. Inferring anchor links across multiple heterogeneous social networks. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 179–188. ACM, 2013.

[115] Ponnurangam Kumaraguru, Lorrie Faith Cranor, and Laura Mather. Anti-phishing landing page: Turning a 404 into a teachable moment for end users. *Conference on Email and Anti-Spam*, 2009.

[116] Ponnurangam Kumaraguru, Lorrie Faith Cranor, and Laura Mather. Anti-phishing landing page: Turning a 404 into a teachable moment for end users. In *Sixth Conference on Email and Anti-Spam*, 2009.

[117] Ponnurangam Kumaraguru, Justin Cranshaw, Alessandro Acquisti, Lorrie Cranor, Jason Hong, Mary Ann Blair, and Theodore Pham. School of phish: A Real-World Evaluation of Anti-Phishing Training. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, page 3. ACM, 2009.

[118] Ponnurangam Kumaraguru, Yong Rhee, Steve Sheng, Sharique Hasan, Alessandro Acquisti, Lorrie Faith Cranor, and Jason Hong. Getting users to pay attention to anti-phishing education: Evaluation of retention and transfer. *e-Crime Researchers Summit, Anti-Phishing Working Group*, 2007.

[119] Ponnurangam Kumaraguru, Steve Sheng, Alessandro Acquisti, Lorrie Faith Cranor, and Jason Hong. Teaching Johnny Not to Fall for Phish. *ACM Transactions on Internet Technology (TOIT)*, 10(2):7, 2010.

[120] Sebastian Kurowski. Using a whatsapp vulnerability for profiling individuals. *Open Identity Summit,GI-Edition - Lecture Notes in Informatics (LNI) - Proceedings 237*, pages 140–146, 2014.

[121] Ni Lao and William W Cohen. Relational retrieval using a combination of path-constrained random walks. *Machine learning*, 81(1):53–67, 2010.

[122] Kyumin Lee, James Caverlee, and Steve 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.

[123] Kyumin Lee, Brian David Eoff, and James Caverlee. Seven months with the devils: A long-term study of content polluters on twitter. In *ICWSM*, 2011.

[124] Martin Lee. Who's next? identifying risks factors for subjects of targeted attacks. In *Proc. Virus Bull. Conf*, pages 301–306, 2012.

[125] Sangho Lee and Jong Kim. Warningbird: Detecting suspicious urls in twitter stream. In *NDSS*, volume 12, pages 1–13, 2012.

[126] Xiang Li, Ben Kao, Yudian Zheng, and Zhipeng Huang. On transductive classification in heterogeneous information networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 811–820. ACM, 2016.

[127] Changchang Liu, Peng Gao, Matthew Wright, and Prateek Mittal. Exploiting temporal dynamics in sybil defenses. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 805–816. ACM, 2015.

[128] Jienan Liu, Babak Rahbarinia, Roberto Perdisci, Haitao Du, and Li Su. Augmenting telephone spam blacklists by mining large cdr datasets. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pages 273–284. ACM, 2018.

[129] Linqing Liu, Yao Lu, Ye Luo, Renxian Zhang, Laurent Itti, and Jianwei Lu. Detecting" smart" spammers on social network: A topic model approach. *arXiv preprint arXiv:1604.08504*, 2016.

[130] Cristian Lumezanu and Nick Feamster. Observing common spam in twitter and email. In *Proceedings of the 2012 ACM conference on Internet measurement conference*, pages 461–466. ACM, 2012.

[131] Justin Ma, Lawrence K Saul, Stefan Savage, and Geoffrey M Voelker. Learning to detect malicious urls. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):30, 2011.

[132] Claudio Marforio, Ramya Jayaram Masti, Claudio Soriente, Kari Kostiainen, and Srdjan Capkun. Personalized Security Indicators to Detect Application Phishing Attacks in Mobile Platforms. *arXiv preprint arXiv:1502.06824*, 2015.

[133] Eva García Martín, Niklas Lavesson, and Mina Doroud. Hashtags and followers. *Social Network Analysis and Mining*, 6(1):1–15, 2016.

[134] Aude Marzuoli, Hassan A Kingravi, David Dewey, and Robert Pienta. Uncovering the landscape of fraud and spam in the telephony channel. In *Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on*, pages 853–858. IEEE, 2016.

[135] McAfee. Mcafee siteadvisor. *http://www.siteadvisor.com/*.

[136] D Kevin McGrath and Minaxi Gupta. Behind phishing: An examination of phisher modi operandi. *LEET*, 8:4, 2008.

[137] Changping Meng, Reynold Cheng, Silviu Maniu, Pierre Senellart, and Wangda Zhang. Discovering meta-paths in large heterogeneous information networks. In *Proceedings of the 24th International Conference on World Wide Web*, pages 754–764. International World Wide Web Conferences Steering Committee, 2015.

[138] Xiaofeng Meng, Chuan Shi, Yitong Li, Lei Zhang, and Bin Wu. Relevance measure in large-scale heterogeneous networks. In *Asia-Pacific Web Conference*, pages 636–643. Springer, 2014.

[139] Najmeh Miramirkhani, Oleksii Starov, and Nick Nikiforakis. Dial one for scam: A large-scale analysis of technical support scams. In *Proceedings of the 24th Network and Distributed System Security Symposium (NDSS)*, 2017.

[140] Daisuke Miyamoto, Hiroaki Hazeyama, and Youki Kadobayashi. An evaluation of machine learning-based methods for detection of phishing sites. In *Advances in Neuro-Information Processing*, pages 539–546. Springer, 2009.

[141] Tyler Moore, Nektarios Leontiadis, and Nicolas Christin. Fashion crimes: trending-term exploitation on the web. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 455–466. ACM, 2011.

[142] Hossen Mustafa, Wenyuan Xu, Ahmad Reza Sadeghi, and Steffen Schulz. You Can Call but You Can't Hide: Detecting Caller ID Spoofing Attacks. In *Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on*, pages 168–179.

[143] Sky News. Losses from telephone banking fraud rise 95 percent. http://news.sky.com/story/1562860/losses-from-telephone-banking-fraud-rise-95-percent, October 2015. (Accessed on 2018-08-13).

[144] Shirin Nilizadeh, François Labrèche, Alireza Sedighian, Ali Zand, José Fernandez, Christopher Kruegel, Gianluca Stringhini, and Giovanni Vigna. Poised: Spotting twitter spam off the beaten paths. *arXiv preprint arXiv:1708.09058*, 2017.

[145] Yuan Niu, Francis Hsu, and Hao Chen. iPhish: Phishing Vulnerabilities on Consumer Electronics. In *UPSEC*, 2008.

[146] Federal Bureau of Investigation. Tech support scam - federal bureau of investigation. https://www.ic3.gov/media/2016/160602.aspx, June 2016. (Accessed on 2018-09-13).

[147] Miles Osborne and Mark Dredze. Facebook, twitter and google plus for breaking news: Is there a winner? In *ICWSM*, 2014.

[148] Raphael Ottoni, Diego B Las Casas, Joao Paulo Pesce, Wagner Meira Jr, Christo Wilson, Alan Mislove, and Virgílio AF Almeida. Of pins and tweets: Investigating how users behave across image-and text-based social networks. In *ICWSM*, 2014.

[149] El Pais. The premium-rate text-messaging scam worth 5 million euros. `http://elpais.com/elpais/2015/04/20/inenglish/1429529298_001329.html`, April 2015. (Accessed on 2018-10-13).

[150] Sharbani Pandit, Roberto Perdisci, Mustaque Ahamad, and Payas Gupta. Towards measuring the effectiveness of telephony blacklists. In *Network and Distributed System Security Symposium, NDSS*, 2018.

[151] Chao-Ying Joanne Peng, Kuk Lida Lee, and Gary M Ingersoll. An introduction to logistic regression analysis and reporting. *The Journal of Educational Research*, 96(1):3–14, 2002.

[152] Jürgen Quittek, Saverio Niccolini, Sandra Tartarelli, Martin Stiemerling, Marcus Brunner, and Thilo Ewald. Detecting SPIT calls by checking human communication patterns. In *Communications, 2007. ICC'07. IEEE International Conference on*, pages 1979–1984.

[153] Md Sazzadur Rahman, Ting-Kai Huang, Harsha V Madhyastha, and Michalis Faloutsos. Frappe: detecting malicious facebook applications. In *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, pages 313–324. ACM, 2012.

[154] Anirudh Ramachandran and Nick Feamster. Understanding the network-level behavior of spammers. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 291–302. ACM, 2006.

[155] Hootan Rashtian, Yazan Boshmaf, Pooya Jaferian, and Konstantin Beznosov. To Befriend Or Not? A Model of Friend Request Acceptance on Facebook. In *Symposium on Usable Privacy and Security (SOUPS)*, 2014.

[156] Bhavani Raskutti and Adam Kowalczyk. Extreme re-balancing for svms: a case study. *ACM Sigkdd Explorations Newsletter*, 6(1):60–69, 2004.

[157] Gustavo Resende, Philipe Melo, Hugo Sousa, Johnnatan Messias, Marisa Vasconcelos, Jussara Almeida, and Fabricio Benevenuto. Mis)information dissemination in whatsapp: Gathering, analyzing and countermeasures. In *Proceedings of the 2019 World Wide Web Conference*, WWW '19. International World Wide Web Conferences Steering Committee, 2019.

[158] Stefan A Robila and James W Ragucci. Don't be a phish: steps in user education. In *ACM SIGCSE Bulletin*, volume 38, pages 237–241. ACM, 2006.

[159] Merve Sahin and Aurélien Francillon. On the effectiveness of the national do-not-call registries, 2018. (Accessed on 2018-12-13).

[160] Merve Sahin, Aurélien Francillon, Payas Gupta, and Mustaque Ahamad. Sok: Fraud in telephony networks. In *Proceedings of the 2nd IEEE European Symposium on Security and Privacy (EuroS&P'17), EuroS&P*, volume 17, 2017.

[161] Merve Sahin, Marc Relieu, and Aurélien Francillon. Using chatbots against voice spam: Analyzing lenny's effectiveness. In *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*, pages 319–337, 2017.

[162] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010.

[163] J Reynaldo A Santos. Cronbach's alpha: A tool for assessing the reliability of scales. *Journal of extension*, 37(2):1–5, 1999.

[164] Marialisa Scat'a and Aurelio La Corte. User-profile framework against a spit attack on VoIP systems. *International Journal for Information Security Research*, pages 121–131, 2011.

[165] Sebastian Schrittwieser, Peter Frühwirt, Peter Kieseberg, Manuel Leithner, Martin Mulazzani, Markus Huber, and Edgar R Weippl. Guess Who's Texting You? Evaluating the Security of Smartphone Messaging Applications. In *19th Annual Network and Distributed System Security Symposium, NDSS 2012*.

[166] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93, 2008.

[167] Steve Sheng, Mandy Holbrook, Ponnurangam Kumaraguru, Lorrie Faith Cranor, and Julie Downs. Who Falls for Phish? A Demographic Analysis of Phishing Susceptibility and Effectiveness of Interventions. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10.

[168] Chuan Shi, Xiangnan Kong, Yue Huang, S Yu Philip, and Bin Wu. Hetesim: A general framework for relevance measure in heterogeneous networks. *IEEE Transactions on Knowledge and Data Engineering*, 26(10):2479–2492, 2014.

[169] Chuan Shi, Xiangnan Kong, Philip S Yu, Sihong Xie, and Bin Wu. Relevance search in heterogeneous networks. In *Proceedings of the 15th International Conference on Extending Database Technology*, pages 180–191. ACM, 2012.

[170] Jaeseung Song, Hyoungshick Kim, and Athanasios Gkelias. iVisher: Real-Time Detection of Caller ID Spoofing. *ETRI Journal*, 36(5):865–875, 2014.

[171] Bharat Srinivasan, Payas Gupta, Manos Antonakakis, and Mustaque Ahamad. Understanding cross-channel abuse with sms-spam support infrastructure attribution. In *European Symposium on Research in Computer Security*, pages 3–26. Springer, 2016.

[172] Bharat Srinivasan, Athanasios Kountouras, Najmeh Miramirkhani, Monjur Alam, Nick Nikiforakis, Manos Antonakakis, and Mustaque Ahamad. Exposing search and advertisement abuse tactics and infrastructure of technical support scammers. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 319–328. International World Wide Web Conferences Steering Committee, 2018.

[173] Wiat Staff. Biggest phone scam in irs history continues to grow as tax season approaches. http://wiat.com/2016/01/20/biggest-phone-scam-in-irs-history-continues-to-grow-as-tax-season-approaches/, January 2016. (Accessed on 2018-10-13).

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

[175] Gianluca Stringhini, Pierre Mourlanne, Gregoire Jacob, Manuel Egele, Christopher Kruegel, and Giovanni Vigna. EVILCOHORT: Detecting communities of malicious accounts on online services. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 563–578, Washington, D.C., 2015. USENIX Association.

[176] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11):992–1003, 2011.

[177] Yizhou Sun, Brandon Norick, Jiawei Han, Xifeng Yan, Philip S Yu, and Xiao Yu. Pathselclus: Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 7(3):1–23, 2013.

[178] Kurt Thomas, Chris Grier, Justin Ma, Vern Paxson, and Dawn Song. Design and evaluation of a real-time url spam filtering service. In *2011 IEEE Symposium on Security and Privacy*, pages 447–462. IEEE, 2011.

[179] Kurt Thomas, Chris Grier, Dawn Song, and Vern 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.

[180] v3.co.uk. Instant messaging to overtake email as biggest digital communication platform. `http://www.v3.co.uk/v3-uk/news/2416558/instant-messaging-to-overtake-email-as-biggest-digital-communication-platform`, July 2015. (Accessed on 2018-10-13).

[181] Alexander Johannes Aloysius Maria van Deursen and Willem Jan Pieterson. The internet as a service channel in the public sector. `https://research.utwente.nl/en/publications/the-internet-as-a-service-channel-in-the-public-sector`, 2006. Accessed on 2016-08-11.

[182] Shoba Venkataraman, Subhabrata Sen, Oliver Spatscheck, Patrick Haffner, and Dawn Song. Exploiting network structure for proactive spam mitigation. 2007.

[183] Arun Vishwanath. Habitual Facebook Use and its Impact on Getting Deceived on Social Media. *Journal of Computer-Mediated Communication*, 20:83–98, 2014.

[184] Bimal Viswanath, Muhammad Ahmad Bashir, Mark Crovella, Saikat Guha, Krishna P Gummadi, Balachander Krishnamurthy, and Alan Mislove. Towards detecting anomalous user behavior in online social networks. In *USENIX Security Symposium*, pages 223–238, 2014.

[185] Alex Hai 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.

[186] Steve Webb, James Caverlee, and Calton Pu. Social honeypots: Making friends with a spammer near you. In *CEAS*, 2008.

[187] Colin Whittaker, Brian Ryner, and Marria Nazif. Large-scale automatic classification of phishing pages. In *NDSS*, 2010.

[188] Yu-Sung Wu, Saurabh Bagchi, Navjot Singh, and Ratsameetip Wita. Spam detection in voice-over-ip calls through semi-supervised clustering. In *Dependable Systems & Networks, 2009. DSN'09.*, pages 307–316. IEEE, 2009.

[189] Guang Xiang, Jason Hong, Carolyn P Rose, and Lorrie Cranor. Cantina+: a feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security (TISSEC)*, 14(2):21, 2011.

[190] Zhi Xu and Sencun Zhu. Abusing Notification Services on Smartphones for Phishing and Spamming. In *WOOT*, pages 1–11, 2012.

[191] Kuldeep Yadav, Ponnurangam Kumaraguru, Atul Goyal, Ashish Gupta, and Vinayak Naik. Smsassassin: crowdsourcing driven mobile-based system for sms spam filtering. In *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*, pages 1–6. ACM, 2011.

[192] Sarita Yardi, Daniel Romero, Grant Schoenebeck, et al. Detecting spam in a twitter network. *First Monday*, 15(1), 2009.

[193] Ezer Osei Yeboah-Boateng and Priscilla Mateko Amanor. Phishing, smishing & vishing: an assessment of threats against mobile devices. *Journal of Emerging Trends in Computing and Information Sciences*, 5(4):297–307, 2014.

[194] Mingxi Zhang, Hao Hu, Zhenying He, and Wei Wang. Top-k similarity search in heterogeneous information networks with x-star network schema. *Expert Systems with Applications*, 42(2):699–712, 2015.

[195] Xianchao Zhang, Zhaoxing Li, Shaoping Zhu, and Wenxin Liang. Detecting spam and promoting campaigns in twitter. *ACM Trans. Web*, 10(1):4:1–4:28, 2016.

[196] Yue Zhang, Jason I Hong, and Lorrie F Cranor. Cantina: a content-based approach to detecting phishing web sites. In *Proceedings of the 16th international conference on World Wide Web*, pages 639–648. ACM, 2007.

[197] Elena Zheleva and Lise Getoor. Privacy in social networks: A survey. In *Social network data analytics*, pages 277–306. Springer, 2011.