A Thesis for the Degree of Ph.D. in Engineering

A Study on Fraud Detection for Sybil Accounts on Social Networking Services and Phishing Websites

February 2020

Graduate School of Science and Technology
Keio University

Shuichiro Haruta

Acknowledgments

I would like to express my special appreciation to my supervisor, Prof. Iwao Sasase. He gave me not only how to research but also the passable ability for every field. I consider the most important thing he taught me is finding root problem, solving it, and telling it in the form that everybody can understand. This instruction gave me a confidence that I can do on any research field. In addition to that, Prof. Sasase entrusted me with research instructions for junior student in the laboratory. I could learn the delight of research and the importance of leadership through that experience. Keeping his instructions in my heart, I would like to be active in the following career. I would also like to thank my committee members, Prof. Naoaki Yamanaka, Prof. Tomoaki Ohtsuki, and Prof. Yukitoshi Sanada for serving as my committee members even at hardship.

All members in Sasase Lab. were great persons. My first senior colleague, Dr. Kentaroh Toyoda, encouraged me to proceed Ph.D course. If he did not belong to Sasase Lab., I would not be who I am now. My second senior colleague, Mr. Hiromu Asahina, gave me very good advice especially about research. I will never forget special days I spent with all members.

I cannot thank my family enough for everything they have done for me. Especially, my parents, Michihiro Haruta and Mika Haruta, supported me in both mental and financial aspects. My grandparents Katsuhisa Haruta, Aiko Haruta, and Takayo Ashizawa always encouraged me to get Ph.D degree. Without their support, I could not finish my Ph.D thesis.

Finally, thank you to my girlfriend Anna Hiramatsu, for all her love and support.

Shuichiro Haruta

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Recent trends in security and the importance of fraud detection

In the late 20-th century, the networking technologies have been greatly improved. Those technologies including Internet have brought us a considerably convenient life. We can communicate with each other wherever we are and get the useful information at any time. In such situation, unfortunately, malicious users called "Attacker" have emerged and their activities threaten our secure use of networking technologies. For example, the eavesdropping attack threatens our privacy of communication and the spoofing attack disguises a communication from an unknown source as being from a known, trusted source [1]. The security researchers have dealt with them by cryptography, authentication technique, and so on. However, these traditional security techniques cannot be applied to all threats because of the divergence of networking services and that of attacking. In other words, these traditional techniques are not countermeasures themselves but the elements of constructing secure systems [2]. Under the circumstances, it is vital to detect malicious activities since we can take some countermeasures by knowing their existence. Such research area of detecting malicious activities or contents is called "Fraud Detection". The fraud detection is important since it can be a second protection for enhancing entire security system. In

this dissertation, we mainly deal with fraud detection in the fields of "Sybil accounts on online SNS (Social Networking Services)" and "Phishing websites". On the other hand, there are various fields where fraud detection is applied. Therefore, we first introduce these fields and describe main themes later.

1.2 The fields of fraud detection

There are various fields where fraud detection is applied. In each field, the target of detection and the type of data are different. Thus, it is necessary to use the fraud detection schemes dedicated to the cases. At the early stage of fraud detection, it is utilized in the limited fields, which are the detection of fraud use of credit card, money laundering fraud, health care insurance fraud and so on [3]–[8]. Recently, the fraud detection have become vital especially in many security fields since the fraudsters can directly attack users connected to Internet or widespread networking services. These emerging fields of fraud detection are classified into two types by the kinds of attacking targets: attacks for many unspecified users or services, and attacks for people who use specific services. In the following sections, we summarize the examples of fields of the fraud detection.

1.2.1 Fraud detection in attacks for many unspecified users or services

Computer virus detection

The personal computer are infected with computer viruses via network. There are many types of computer viruses including trojan horses and ransomware [9]–[13]. This attack covers a lot of malicious objectives. For example, in case of trojan horses, they allow an attacker to access users' personal information, delete users' file and so on. In the case of ransomware, it encrypts user's files and requests money in exchange for the decryption key. They are detected by so-called anti-virus software. Anti-virus software is a kind of fraud detection tool and schemes dedicated to each virus are

needed.

Malicious document detection

The de facto standard document files which are Microsoft Word, Adobe PDF (Portable Document Format), and HTML (Hyper Text Markup Language) are also target of attack [14]–[20]. Attackers insert malicious codes to their internal file structure. For example, a variety of behavior of browser can be written by Javascript in HTML. The objective of these attacks is to steal sensitive information such as ID and Password. Since many people casually open these files, the detection prior to opening is important.

DoS/DDoS attack detection

DoS (Denial of Service) and DDoS (Distributed DoS) attack is another example. In DoS attack, the network resources such as web servers are targeted by attackers. Attackers send a large amount of packets and make the network resources unavailable by the flood of packets [21]. In a DDoS attack, many different sources called "botnet" send packets in order to effectively make it impossible to stop the attack simply by blocking a single source [22]. Many researches try to detect botnets [23]–[25].

Nuisance behavior detection in emerging physical network

Emerging physical networks such as VANET (Vehicular Ad hoc Network), DTN (Delay Tolerant Network), and NDN (Named Data Network) might suffer from nuisance behavior by attackers [26]–[28]. VANET is a basic technique which realize automatic driving. DTN provides the reliable communication in the disaster area. NDN reduces the burden of the content servers. In each network, there might exist malicious entities which send the falsified information and disturb communication of others. For example, in VANET, although it is vital to share traffic information among vehicles, there might exist malicious ones which send falsified traffic information [29]. Since the falsified traffic information might incur traffic accidents, such vehicles should be detected.

1.2.2 Fraud detection in attacks for people who use specific services

Detection of malicious apps on smartphone

Smartphones have been major communication tool many people use all over the world. Especially, Android is the most popular smartphone platform occupying 85% of share in the world [30]. Unfortunately, smartphones apps running on Android system have become the main target of attackers due to its popularity. The Android apps released on Google Play which is the official store of apps are automatically evaluated because manual evaluation spends a lot of expenses and more time. Since such evaluation cannot completely prevent malicious apps from spreading, users are under the risk of installing them. Thus, this circumstance results in the urgency of detecting malicious apps [31]–[36].

Sybil accounts detection on SNS

Due to the widespread of the mobile devices like smartphones, SNS such as Facebook and Twitter has revolutionized the ways in which people interact, think and conduct business all over the world [37]. However, the system of SNS is vulnerable to the Sybil attack [38]. In this attack, the attackers can create an unlimited number of fake accounts (Sybil accounts) with the intention to bother legitimate users by the behavior such as sending spams to legitimate users and illegal voting to some contents [39]. Thus, the service providers have to detect Sybil accounts for securing SNS to prevent attacker's activity [39]–[41].

Phishing websites detection

Recently, people all over the world benefit from online services such as e-banking, e-commerce and so on [42], [43]. In this situation, the phishing websites have emerged for attackers to steal personal information (e.g. credit card number, login ID or password) of innocent users [44]–[46]. The phishing websites target the famous legitimate websites and mimic their appearance. Thus, unaware Internet users input

sensitive information to the phishing websites and the attack succeeds. Since the stolen information incurs a large amount of financial losses, phishing websites should be detected.

1.3 Main themes of this dissertation

Recently, Internet is used by people all over the world and the number of attacks using Internet increases. Since the web services are main platforms of today's Internet utilization, we propose some countermeasures to attacks for them. In this dissertation, we deal with the fraud detection of Sybil accounts on SNS and phishing websites. We mention the motivations below.

The reasons why we focus on detection of Sybil accounts on SNS are as follows; Sybil accounts on SNS can become the entrance of many kinds of attacks including phishing, spreading spams, and so on since SNS can be primary touch point with many legitimate users and attackers. It is also easy for attackers to target specific users. In addition to that, the number of SNS users will still increase in the future [47]. In other words, the number of potential victims of Sybil accounts is large. Therefore, we consider the research to detect Sybil accounts leads the large part of Internet environment to be secure and they can be a great impact on the field of security.

The reasons why we focus on detection of phishing websites are as follows; First, the number of victims and the financial loss by phishing websites are very large [48]. This is because phishing websites can be easily made by using parts of targeted legitimate website. Hence, everyone can be an attacker if he/she has the knowledge about HTML. Second, the phishing attack can be applied to all web services dealing with users' sensitive information. In other words, the risk of phishing attack always exists when new web services appear. Therefore, we consider the research to detect phishing websites is very important and can contribute to both services and innocent legitimate users.

In the following sections, we describe the detailed background of each research and the overview of defence mechanisms.

1.4 Sybil accounts on SNS and defences

Thanks to the widespread of the mobile devices like smartphones, SNS such as Facebook and Twitter are used as the communication tools all over the world. As mentioned in section 1.2.2, although these services are convenient to communicate, they are vulnerable to the Sybil attack. The Sybil attack is named after the subject of the book Sybil, which describes a person diagnosed with dissociative identity disorder [38]. In this attack, the attackers can create an unlimited number of fake accounts (Sybil accounts) with the intention to bother legitimate users by the behavior such as sending spams, illegal voting to social contents, and so on. Thus, the service providers have to detect Sybil accounts for securing SNS. However, today's SNS operators manually detect Sybil accounts and its coverage is not good. Today's major SNS such as Facebook have a function to report abusive accounts [49]. In this situation, SNS operators have to manually detect Sybil accounts based on the inspection of users' reports and it takes considerable cost and time. In fact, according to [50], Facebook estimates that 83.09 million users can be fake accounts and the manual inspection is inefficient.

Recently, many schemes have been proposed in order to combat Sybil accounts. We describe the overview of those schemes in the next section.

1.4.1 Defences against Sybil accounts

The schemes against Sybil accounts are classified into two types, namely, "Sybil tolerance schemes" and "Sybil detection schemes". Figure 1-1 shows a typical classification of defences against Sybil accounts on SNS. Sybil tolerance schemes [51]–[61] are classified into two types.

The first type of Sybil tolerance schemes [51]–[56] is designed for specific functionalities or purposes of SNS such as communications [51], [53], [55], voting [52], [54], [55], and analysis [56]. For example, in Ostra [51], that scheme prevents legitimate accounts from receiving malicious messages. By depressing the credit values of the directional relationship that Sybils use to send malicious messages based on user's

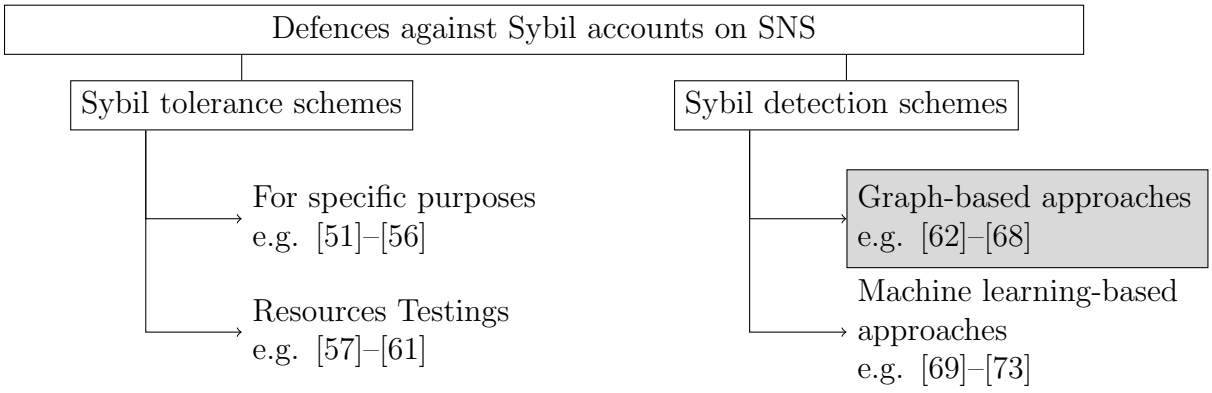


Figure 1-1: Classification of defences against Sybil accounts.

judgement, the credit value of Sybil accounts quickly depletes. Another example is SumUp [52] which aims at limiting the number of bogus votes that Sybil accounts cast in a voting functionality of SNS. Similar to Ostra, that scheme also uses a social network and assigns the credit values on the social relationships between users. That scheme chooses a vote collector and distributes voting tickets. To cast a vote, each voter must find a path to the vote collector with sufficient credit. If such path cannot be searched, the vote is considered to be invalid. It is difficult for Sybils to search such path since the number of the relationships between Sybils and legitimate accounts is small. Second type of Sybil tolerance schemes is resources testing [57]–[61]. The basic idea behind the resources testing schemes is that an account has to consume some resources when participating in SNS. If an attacker participates in SNS, those schemes succeed to limit Sybil accounts since he/she consumes a lot of resources because of the large number of Sybils. These resources may include computation power, memory, and so on. The simplest example is CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) [61]. CAPTCHA requests challenges (e.g. illegible characters) to users and requires human resource to solve. In [59], [60], such challenge is computation puzzle.

On the other hand, Sybil detection schemes are classified into two types, namely graph-based approaches [62]–[68] and machine learning-based approaches [69]–[73]. First, we introduce graph-based approaches [62]–[68]. Most of graph-based schemes

propose the attempt to detect Sybil accounts by utilizing topological feature of social graph on the basis of two assumptions. The first assumption is that Sybil accounts cannot make a lot of relationships between legitimate accounts since it is rare for legitimate accounts to accept friend requests from strangers even if Sybil accounts can make relationships among themselves. The second assumption is that the legitimate communities are "Fastmixing". Fastmixing means that if we take a random walk in a social graph, we will quickly arrive at the stationary distribution compared to Sybil communities [62]. The representative graph-based approach is SybilRank [66]. SybilRank distinguishes Sybil from non-Sybil accounts based on their trust values [66]. That scheme ranks all accounts' trust values and identifies low trust value accounts as Sybil ones by leveraging the fact that Sybil accounts might be isolated from non-Sybil accounts' community. That scheme uses power iteration [74], which is a technique to efficiently calculate the landing probability of random walks in large graphs. The trust values in legitimate communities are fastmixed by power iteration and low trust value might be distributed to Sybil accounts since legitimate accounts hardly connect with Sybil accounts.

Finally, we describe machine learning-based approaches [69]–[73]. Those approaches detect Sybil accounts by utilizing machine learning classifier such as SVM (Support Vector Machine) [75] and decision tree models [76]. The features fed into classifier are dependent on each scheme. For example, Yang et al. use the ratio of accepted incoming friend requests as a feature [69]. The friend requests of Sybil accounts tend not to be accepted. In [71], the clickstream model is proposed to characterize the click behaviors of users. Authors analyze clickstream activity of click patterns of legitimate and Sybils and the result is reflected to the models.

Sybil tolerance schemes are applied to limited aspects of SNS. The features in machine learning-based approaches can be easily avoided by attacker and it is difficult to obtain them without full-access to each account. Thus, although several schemes are proposed in various forms, we pay attention to graph-based approaches. Note that the proposal of this dissertation about Sybil detection on SNS is classified into graph-based approaches. Therefore, the box of graph-based approaches in Figure 1-1

is filled with gray. The detailed descriptions about graph-based approaches are shown in section 2.1.5.

1.5 Phishing websites and defences

Recently, people all over the world use online services such as e-banking, shopping and so on. As mentioned in section 1.2.2, in this situation, the phishing websites have emerged for attackers to steal personal information (e.g. credit card number, login ID or password) of innocent users [44]. Since attackers fish for a "P"ersonal information, this attack is referred to as "Phishing" attack [77]. The phishing websites target the famous legitimate websites and mimic their appearance. Thus, unaware Internet users input sensitive information to the phishing websites and the attack succeeds. According to the report of Anti-Phishing Working Group (APWG), in third quarter of 2018, more than 150 thousand unique phishing websites are found and their threat has continued [78]. Therefore, the defences against phishing websites are urgent demand.

Recently, many schemes have been proposed in order to combat phishing websites. We describe the overview of those schemes in the next section.

1.5.1 Defences against phishing websites

The defences against phishing websites are classified into three types, namely, "User training", "Phishing prevention", and "Phishing detection". Figure 1-2 shows classification of defences against phishing attack. This classification is based on the review paper [124].

The first one is user training based schemes [79]–[86]. The purpose of those scheme is to educate users' literacy and avoid the phishing attack. They are classified into two types, namely simulation-based training and game-based training. In simulation-based training schemes [79]–[83], those schemes simulate phishing attack such as receiving phishing e-mails. Various type of phishing websites are displayed and users can train the literacy for phishing websites. In game-based training [84]–[86], users can train their literacy via game of identifying phishing websites. The game-based

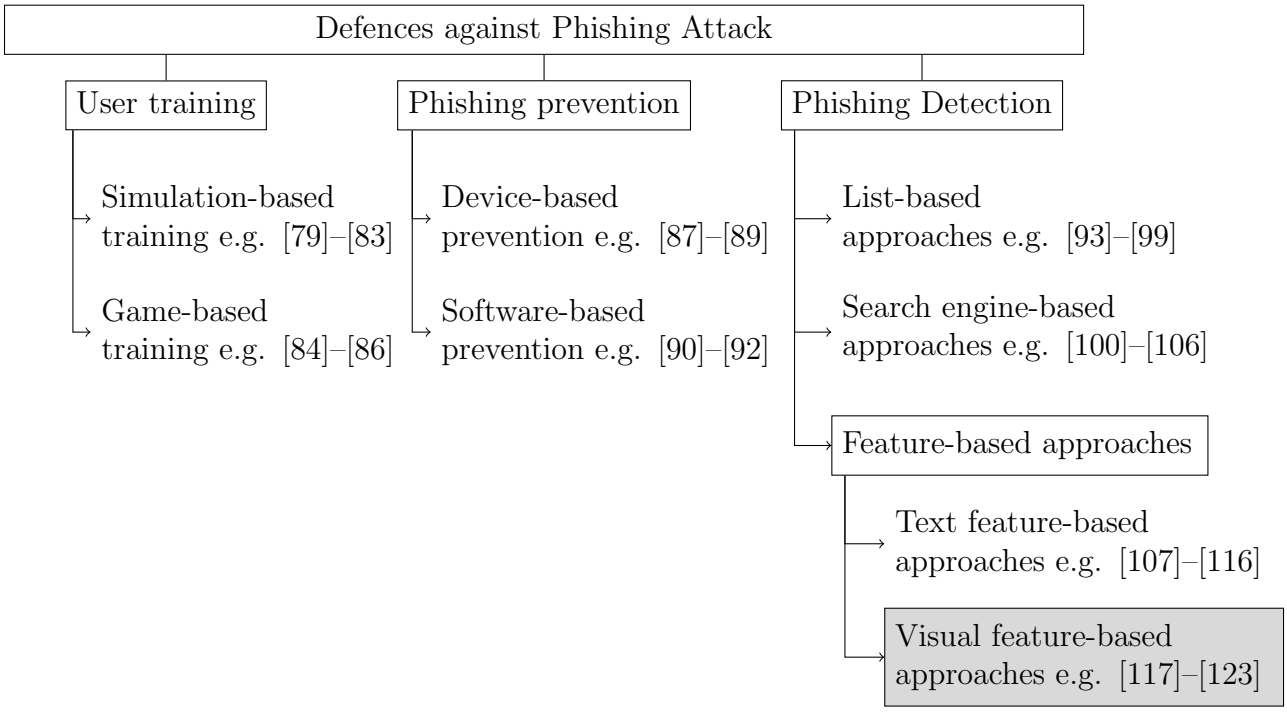


Figure 1-2: Classification of defences against phishing attack.

education allows users to learn through experience and the use of virtual environment while leading them to solve problem via critical thinking [85].

The second type of defences is phishing prevention schemes which are device-based prevention [87]–[89] and software-based prevention [90]–[92]. Those schemes try to prevent phishing attacks by providing an extra layer of authentication techniques which are similar to two factor authentication. This reduces the probability of a user being deceived by phishing website. Device-based prevention schemes utilize the external authentication devices. For example, in [89], a smart card which is created by web service provider a user want to login is utilized. When the user wants to login, he/she sends access token of smart card to the server and it returns PIN code to user's smart card. If that does not match the PIN which the smart card has, the login attempt fails. Since phishing websites cannot return correct PIN, the user can find it phishing. Software-based prevention schemes realizes the procedure like this based on algorithms such as watermarking and picture password. In the case of watermarking-based phishing prevention scheme [90], watermarking image and its

position which are set in registering the system are utilized to authenticate. When the user tries to login by access token, the service returns the watermarking image to the preset area. Since the user knows the image and its position, he/she can find it phishing if a different image is shown or its position is different.

The final type of defences is phishing detection. The phishing detection schemes are roughly classified into three types, list-based approaches [93]–[99], search engine-based approaches [100]–[106], and feature-based approaches [107]–[123]. Feature-based approaches are further classified into text feature-based approaches [107]–[116] and visual feature-based approaches [117]–[123].

The list-based approaches detect phishing websites by blacklist [93]–[96] and whitelist [97]–[99]. Blacklist-based approaches register the information of phishing websites such as URL (Uniform Resource Locator) and DNS (Domain Name Server) and detect phishing websites based on that. On the other hand, whitelist-based approaches register the information of legitimate website and raise alert when accessing to the websites which are not listed. List-based approaches are generally used and the interests of researches are how to update lists or searching time in most of cases. It is said that the detection rate of security products sold in the market is more than 99% and most of them are based on these approaches. However, there are many cases where the phishing websites which is not listed cannot be detected. In fact, although Google Chrome, a kind of web browser, has a phishing detection mechanism based on blacklist [94], some phishing websites can be accessed without alerts [125]. Thus, the notion of defence in depth and detecting them from many aspects are important.

In Search engine-based approaches [100]–[106], most of schemes use the result of search engine as whitelist. That is, the idea behind those approaches is that the website indexed by Google is not a phishing website. The representative approach is Cantina [100] by Zhang et al.. In that scheme, they utilize TF-IDF (Term Frequency and Inverse Document Frequency) algorithm [126] to extract keywords of a website and these keywords are searched by Google. If the website appears in the whitelist (search results), it is legitimate; otherwise phishing.

In feature-based approaches [107]–[123], many features have been proposed. Text feature-based approaches [107]–[116] include the text features such as entire length of URL, domain information, DNS information, Alexa [127] rank, Alexa reputation, number of links and so on. The different points in those schemes are the type and number of features extracted, the classifier which brings the best performance, the use of logic from other schemes and so on. On the other hand, the idea behind visual feature-based approaches [117]–[123] is that the visual contents of phishing websites tend to be similar to the targeted legitimate website since a phishing website naturally mimics the appearance of the target. Those approaches use "signature" which is feature map in layout, position of colors, etc. extracted from the targeted legitimate website or phishing websites. Signatures are stored in SDB (Signature DataBase) and the website whose signature is similar to SDB's one is detected as a phishing website. Thus, visual feature-based approaches are referred to as "signature-based approaches" or "visual similarity-based approaches". For example, in [122], that scheme uses position of colors as a signature. The similarity of signatures is calculated by EMD (Earth Mover's Distance) [128], [129]. Hereinafter, we refer visual feature-based approaches as "visual similarity-based approaches" since that name clearly indicates using visual features and implies using signatures.

In user training-based approaches, it is difficult to apply such educations to all low literacy users. In phishing prevention schemes, it is necessary to use additional devices or complicated knowledge about schemes. List-based phishing detection cannot deal with websites whose information is not listed and it incurs zero-day attack. Search engine-based approaches and text feature-based approaches cannot deal with attacker's manipulating of features. For example, the feature of URL is easily changed by attackers. In addition to that, although search engine-based approaches use search engine through a search API (Application Programming Interface) provided by search engines, it is expensive. Thus, we pay attention to visual similarity-based approaches. Note that the proposal of this dissertation about phishing detection is classified into visual similarity-based approaches. Therefore, the box of visual feature-based approaches in Figure 1-2 is filled with gray. The detailed descriptions about visual

similarity-based approaches are shown in section 2.2.4.

1.6 Research motivations and overviews

Although many researchers have solved important issues, they have shortcomings and require further improvements. In the research of Sybil detection on SNS, the promising approaches are SybilRank [66] and graph pruning scheme [67]. However, in the SybilRank, legitimate seeds, which are accounts the initial trust values are given, are concentrated on the specific communities because they are selected from nodes that have largest number of friends, and thus the trust value is not evenly distributed. In the graph pruning scheme, although it tries to prune relationships between legitimate accounts and Sybils for the purpose of limiting trust values Sybils get, a sophisticated attacker can avoid graph pruning by making relationships between Sybil accounts. Hence, we propose a robust seed selection and graph pruning scheme to detect Sybil accounts more accurately. To more evenly distribute trust value into legitimate nodes, we first detect communities in the SNS and select legitimate seeds from each detected community. In addition to that, by leveraging the fact that Sybils cannot make dense relationships with legitimate nodes, we also propose a graph pruning scheme based on the density of relationships between trusted accounts. We prune the relationships which have sparse relationships with trusted accounts and this enables robust pruning malicious relationships even if the attackers make a large number of common friends.

In the research of phishing detection, the promising approaches are visual similarity-based approaches which use signatures. However, they can only detect phishing websites whose signatures are highly similar to SDB' s one. This incurs the vulnerability of zero-day phishing attack. In order to address this issue, though an auto signature update mechanism is needed, the previous approaches' signatures are not suitable for auto updating since their similarity can be highly different among targeted legitimate website and subspecies of phishing website targeting that legitimate website. Hence, we propose a novel visual similarity-based phishing detection scheme using hue in-

formation with auto updating database. Since a phishing website is created based on targeted legitimate website or other subspecies whose hue information is similar each other, many phishing websites can be exhaustively detected by tracing similar colored subspecies. By repeating this procedure and automatically updating SDB, the detection scope can be effectively expanded. In order to avoid the misdetection of legitimate websites which have similar hue information to SDB's ones, the proposed scheme utilizes the fact that the combination of used colors is hard to be similar among legitimate and phishing websites.

1.7 Outline of dissertation

As shown in Figure 1-3, this dissertation is constructed as follows:

Chapter 2 deals with the related works of the proposals.

In chapter 3, we deal with Sybil detection scheme on SNS. We focus on the fact that the legitimate accounts on SNS belong some communities and that Sybil accounts establish sparse relationships with legitimate accounts. Based on this fact, we propose a scheme which effectively detect Sybil accounts. The effectiveness is shown by the computer simulation with real dataset.

In chapter 4, we deal with a detection scheme of phishing websites. Since a phishing website is created based on targeted legitimate website or other subspecies whose hue information is similar each other, many phishing websites can be exhaustively detected by tracing similar colored subspecies. By repeating this procedure, the detection scope can be effectively expanded. We demonstrate that the proposed scheme improves the detection performance as the number of detected phishing websites increases by the computer simulation with real phishing websites' dataset.

Chapter 5 concludes this dissertation and summarizes the contribution of this work.

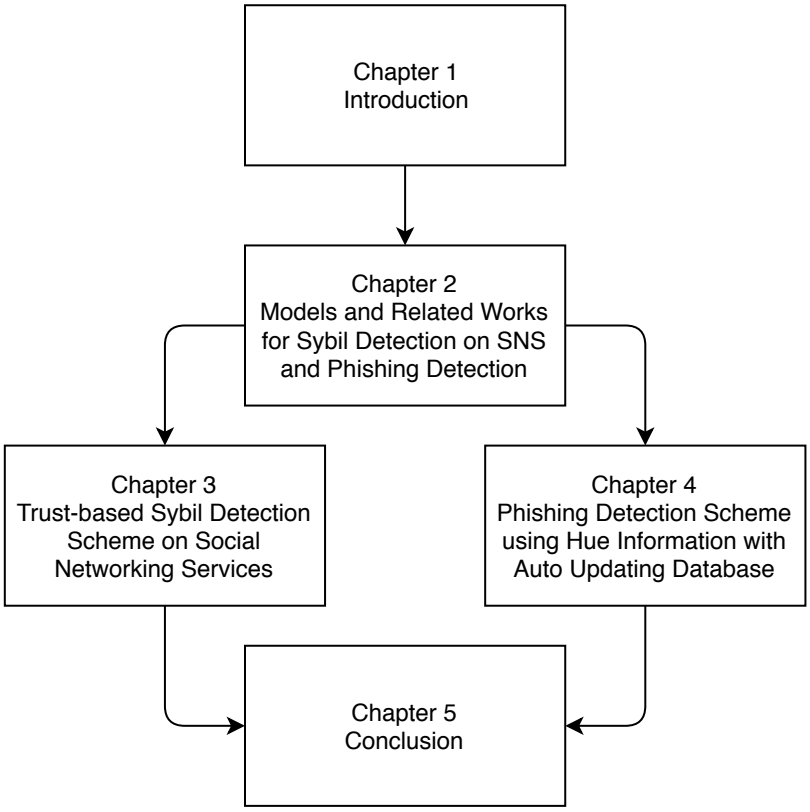


Figure 1-3: Outline of Dissertation.

Chapter 2

Models and Related Works for Sybil Detection on SNS and Phishing Detection

In this chapter, we first define the system and attacker model for each research. After the definition of models in each research, the related works of this dissertation and their shortcomings are described.

2.1 Graph-based Sybil detection on SNS

2.1.1 System model

We consider an undirected social network modeled as a graph $G = (V, E)$, where each node in V represents an account in the network and each edge in E represents a relationship (friendship) between accounts. In an undirected social network like Facebook, each account has to send friend requests when making friends. That is, accounts can make relationships after mutual agreement. We define n, n_L, and n_S as the total number of all accounts, legitimate accounts, and Sybil accounts, respectively, i.e. $|V| = n = n_L + n_S$. Similarly, we define m as the total number of edges, i.e. $m = |E|$. Hereinafter, according to the term used in a graph theory, let the term

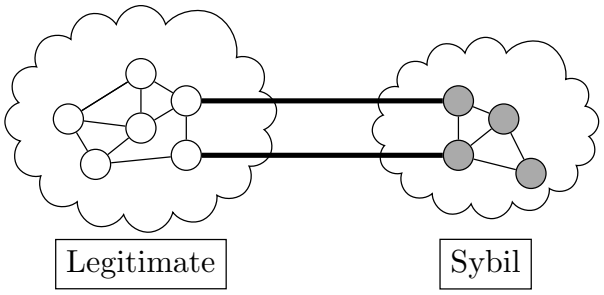


Figure 2-1: The simple structure of SNS.

"node" denote the account. Moreover, we refer to the number of friends of each node as "degree".

A structure model of legitimate region and Sybil region is shown in Figure 2-1. In Figure 2-1, the white nodes, grey nodes, and thick edge denote legitimate nodes, Sybil nodes, and attack edges, respectively. In this dissertation, we assume that the service provider of SNS knows all nodes and their relationships. However, it does not know which nodes are legitimate or Sybil nodes. Although today's SNS operators manually detect Sybil accounts, it typically takes considerable time. The objective of this research is for the service provider to detect as many Sybil nodes as possible without judging legitimate nodes as Sybil nodes. The structure model like this is adopted in many researches [66].

2.1.2 Definition of Sybils on SNS

Generally, Sybil attack is defined as "conducting nuisance behavior to network participants by using large number of network entities". In the case of SNS, accounts created for the following purposes are defined as Sybil accounts:

1. Illegal voting to web contents and spreading fake news
2. Pretending legitimate accounts
3. Preparing other accounts against main attack account's suspension

For these purposes, a large number of accounts is needed. Since many researches are based on this definition (e.g. [66]), we also follow this.

2.1.3 Attacker model

In our attacker model, we assume there are multiple attackers $\mathcal{A} = \{a_i | 1 \leq i \leq n_{att}\}$. n_{att} represents the number of attackers. The reason why we define n_{att} is to model the situation where multiple types of Sybils exist, e.g. sending spams and illegal vote [130], [131]. We assume each attacker creates $n_{S/Attacker}$ Sybils and thus the total number of Sybils n_S can be represented as $n_S = n_{att} \times n_{S/Attacker}$ in the evaluation. Each attacker can execute the following operations:

1. Creating $n_{S/Attacker}$ Sybil nodes.
2. Making relationships among Sybil nodes created by an attacker.
3. Making as many as g relationships called attack edges between Sybil nodes and legitimate nodes. The number of attack edges g is smaller value than m because it is rare for legitimate nodes to accept friend requests from strangers.

2.1.4 Validity of attack model

According to [132], general SNSs have "homophily property", which indicates two linked nodes share the same label with a high probability. That is, the friends of a legitimate node tend not to be Sybils but legitimate nodes. In fact, "homophily among legitimate nodes" and "non-homophily between a legitimate node and Sybils" are found in Tuenti which is largest SNS in Spain. The attack model in the previous section is valid since it models homophily property mentioned above. Graph-based approaches often work well in this assumption. Although there is the case where homophily cannot hold [69], the authors in [133] mention that the situation like this can be mitigated by machine learning based approaches. Hence, we basically follow this attack model.

2.1.5 Related works

As mentioned in section 1.4.1, there have been several Sybil detection schemes based on topological features on the basis of two assumptions. The first assumption is that

Sybil nodes cannot make a lot of relationships between legitimate nodes since it is rare for legitimate nodes to accept friend requests from strangers even if they can make relationships among themselves. The second assumption is that the legitimate communities are "Fastmixing". Fastmixing means that if we take a random walk in a social graph we will quickly arrive at the stationary distribution compared to Sybil communities [133]. Yu et al. propose SybilGuard and SybilLimit [62], [63] which are the first two protocols to exploit topological features to detect Sybil nodes. In SybilGuard they define verifier route composed with legitimate nodes and each node executes random walk with length $O(\sqrt{n}\log n)$. Since random walk in legitimate communities are fastmixing, random walks starting from legitimate nodes tend to intersect verifier route. Based on this notion, Sybils can be detected. Furthermore, SybilLimit can accept the larger number of attack edges than SybilGuard by using multiple walk. However, both schemes suffer from high false rate. Although SybilInfer uses the Bayesian inference that calculates the probability of being Sybil, it takes much computational cost [64]. Cao et al. propose SybilRank [66]. They give a trust value to "legitimate seeds" which is randomly selected from high degree nodes and its trust value is evenly distributed to its neighbors recursively. Since the trust value that Sybils obtain is only via attack edges, the final trust values that legitimate nodes have tend to be higher than Sybils. Since this trust distribution is repeated $O(\log n)$ times, total computational cost is $O(n\log n)$. In [67] by Zhang et al., in order to avoid the trust value from being distributed into Sybil nodes, they prune such suspicious relationships based on the number of common friends prior to performing SybilRank. This idea comes from the perspective that the relationships with a few common friends are to be suspicious. In contrast to all previously discussed approaches, Tran et al. propose Gatekeeper which does not leverage random walk for Sybil detection [68]. Rather than using random walk, Gatekeeper employs a breadth-first-search. In that scheme, a central authority called admission controller selects a number of ticket sources like legitimate seeds in SybilRank and gives them a number of tickets. Then, ticket sources evenly distributes the tickets to theirs neighbors. To be admitted into System, a node must obtain a certain number of tickets. The idea of Gatekeeper is

similar to SybilRank. However, in the modern scenario, it becomes easier for Sybils to obtain the trust values or tickets in SybilRank and Gatekeeper. The most easiest way for Sybils to obtain them is connecting attack edges near seeds. In order to avoid this, Zhang et al. propose a scheme to prune attack edges placed near seeds [67]. Although many graph-based approaches have been proposed, we pay attention to SybilRank [66] and the graph pruning scheme [67] which is extended work of SybilRank. The detection accuracy of SybilRank is higher than other schemes and the effectiveness is proven in real environment [66]. There are many schemes based on SybilRank and the graph pruning scheme [67] is a kind of it. The graph pruning scheme can be utilized without degrading performance and thus is promising. This is why we select them as the previous schemes.

2.1.6 SybilRank

Cao et al. propose a Sybil nodes detection scheme called "SybilRank" which distinguishes Sybil from non-Sybil nodes based on their trust values [66]. That scheme ranks all nodes' trust values and identifies nodes with low trust value as Sybil ones by leveraging the fact that Sybil nodes might be isolated from non-Sybil nodes' community. That scheme uses power iteration [74], which is a technique to efficiently calculate the landing probability of random walks in large graphs. The intuition behind this is that if each node evenly distributes its trust value to its neighbors, low trust value might be distributed to Sybil nodes since legitimate nodes hardly connect with Sybil nodes. More specifically, the power iteration scheme randomly selects M legitimate seeds from nodes that have high degree and gives an initial trust value to each node as

$$T^{(0)}(v) = \begin{cases} \frac{T_G}{M} & \text{if } v \text{ is a seed,} \\ 0 & \text{otherwise,} \end{cases} \tag{2.1}$$

where $T^{(0)}(v)$ denotes the initial trust value on node v and T_G indicates the total trust value given to M legitimate seeds. After giving initial trust value to each node, each node evenly distributes its trust value to its neighbors. Let $T^{(i)}(u)$ denote a node u's

trust value after ith iterations and it is represented as

$$T^{(i)}(u) = \sum_{u_j \in U_u} \frac{T^{(i-1)}(u_j)}{deg(u_j)}, \tag{2.2}$$

where U_u and $deg(u_j)$ denote the set of node u's neighbors and the degree of u_j, respectively. The scheme iterates the above procedures $w = \lceil \log n \rceil$ times and identifies nodes whose trust value is lower than the threshold C_{TH} as Sybil nodes. By terminating the iterative trust distribution procedure by $\lceil \log n \rceil$ times, it is possible to limit the trust value being given to Sybil regions and reduce the computational cost. Since this scheme requires $O(\log n)$ power iterations for each node, the total computational cost is $O(n \log n)$.

2.1.7 Graph pruning

Zhang et al. propose a graph pruning scheme which tries to cut attack edges prior to power iteration [67]. A graph pruning scheme reduces the possibility that a legitimate node gives its trust value to Sybil nodes at the power iteration scheme. Intuitively, when two nodes have few common friends, their relationships are appeared to be an attack edge. From this point of view, that scheme prunes relationships based on the number of common friends. That scheme randomly selects legitimate seeds and determines the size of pruning region G', which is defined by T_p. Pruning region G' is composed of legitimate seeds and their T_p hop neighbors. And then, this scheme calculates w_{ij}, where w_{ij} is the number of common friends of nodes i and j in G'. Finally, the edges whose $w_{ij} \leq T_s$ are to be pruned, where T_s is the threshold to determine whether to be pruned. The values of T_s and T_p used in the simulation are shown in Table 3.1. Figure 2-2 shows an example of the previous graph pruning. In this figure, thick edge, white nodes, and gray nodes indicate an attack edge, legitimate nodes and Sybil nodes, respectively. The number of common friends between node A and B is zero (i.e. $w_{AB} = 0$) and thus the edge between node A and B has the possibility of an attack edge. If we set the pruning threshold T_s as $T_s = 0$, since the inequality $w_{AB} \leq T_s$ holds, this thick edge is to be pruned.

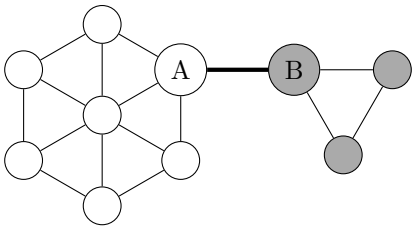


Figure 2-2: Example of graph pruning in previous scheme.

2.1.8 Shortcoming of legitimate seed selection

Both of the previous schemes randomly select legitimate seeds from nodes with high degree. However, since nodes with high degree generally tend to belong to the same community, the previous schemes select legitimate seeds only from specific communities as shown in Figure 2-3. In this figure, $\mathrm{LS}_1, \mathrm{LS}_2$, and LS_3 are nodes selected as legitimate seeds in the previous schemes. We can see that seeds are concentrated on a specific community. Figure 2-4 shows communities where the top $K\%$ highest degree nodes belong. We use a Facebook dataset in [134] and detect communities with the scheme in [135]. For example, when $K = 5\%$, most of high degree nodes belong to one of the 4 out of 13 communities. When $K = 30\%$, most of high degree nodes belong to one of the 5 out of 13 communities. From this result, legitimate nodes which belong to a small community and far from legitimate seeds may not get a sufficient trust value when power iteration is executed and are regarded as Sybil nodes.

2.1.9 Shortcoming of graph pruning

The previous scheme [67] prunes the relationships appeared to be an attack edge based on the number of common friends. However, since the number of common friends are easily increased by attackers' tactic, it is possible for attackers to avoid attack edges from being pruned. Figure 2-5 shows the example of avoiding pruning. In this figure, a white node L and grey nodes S_1, S_2, S_3, and S_4 show an legitimate node and Sybil nodes, respectively, and thick relationships are attack edges. Each Sybil node makes attack edges not only with L but also with other Sybil nodes. This assumption is reasonable since such a legitimate node accepts friend requests from

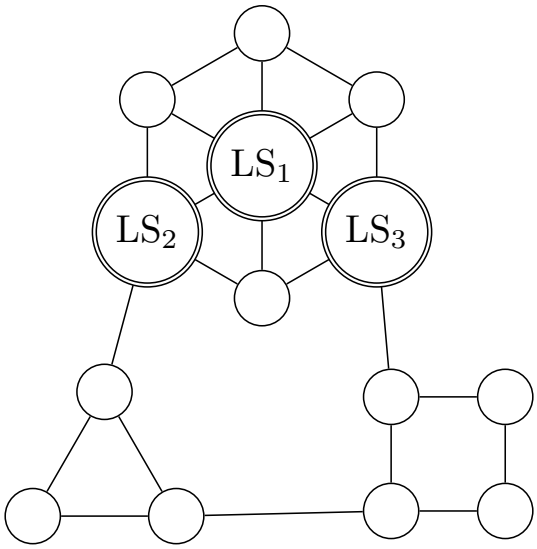


Figure 2-3: Example of seeds selected in the previous scheme.

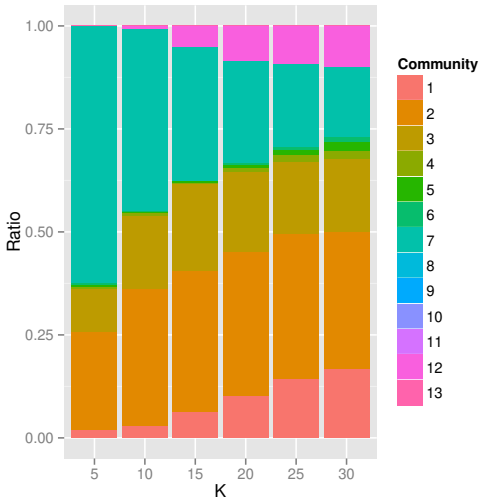


Figure 2-4: The ratio of nodes with high degree in each community.

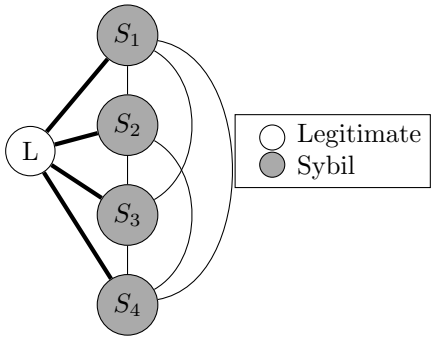


Figure 2-5: Example of avoiding pruning.

any unknown users in order to increase the number of friends for gaining popularity [136]. In this case, the number of common friends between L and a Sybil node is increased to three. Thus, since graph pruning scheme using the number of common friends cannot prune attack edges in Figure 2-5, the accuracy of finding Sybil nodes is degraded.

The scheme which solve these shortcomings is proposed in Chapter 3.

2.2 Visual similarity-based phishing detection

2.2.1 System model

Figure 2-6 shows the system model of phishing detection. As shown in Figure 2-6, we assume signature based phishing detection. A signature indicates website's feature. The user sends URL of visited website to the detection server when it is suspicious. In the detection server, there are detection scheme and SDB which possesses each targeted website's signature. SDB is maintained by a system administrator. The detection scheme creates a signature from the website sent by the user and searches a similar signature from SDB. If a similar signature is found, the detection scheme sends the user that it is phishing website; otherwise legitimate. Prior to the detection, the system administrator registers the signatures for detecting phishing websites by hand.

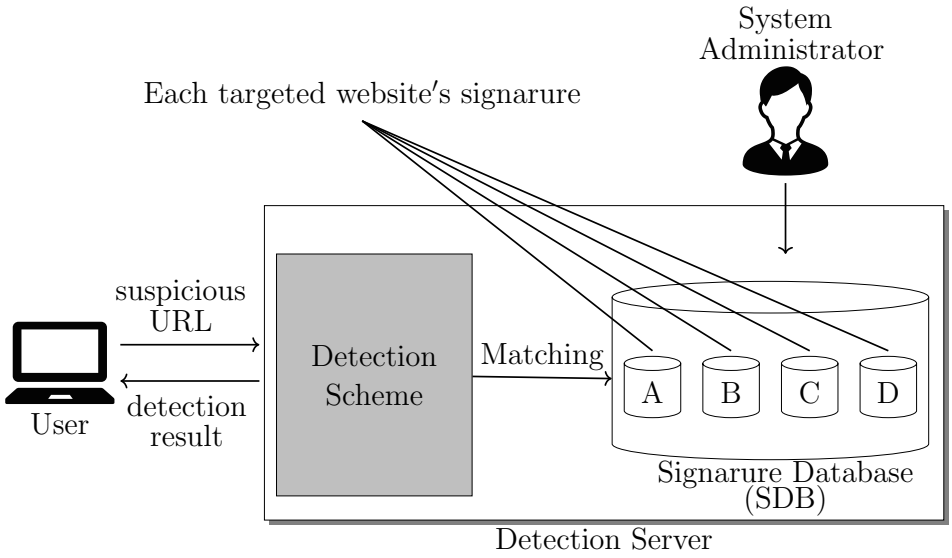


Figure 2-6: System model in the signature-based phishing detection.

2.2.2 Target of detection

In this section, we clearly define the target of detection. Figure 2-7 shows the classification of malicious websites. As shown in Figure 2-7, there are roughly two types of malicious websites, which are phishing website and exploit kit website [137]. Exploit kit websites make visitors download malicious files or programs and they conduct malicious activities. Although those malicious activities include leaking personal information of users, we do not define this as a kind of phishing attack. In this dissertation, we define phishing websites as "the website which mimics the appearance of famous legitimate website and steals personal information by letting unaware users input". We focus on phishing websites and do not deal with exploit kit websites.

2.2.3 Attacker model

Figure 2-8 shows the example of the attacker model we assume. The attacker steals innocent users' sensitive information as follows:

1. Attacker creates a phishing website whose URL is $\text{URL}_{\text{phishing}}$. The phishing website target a specific legitimate website and is visually similar to it.

2. Attacker sends $\text{URL}_{\text{phishing}}$ to users via e-mail or SNS.

3. Unaware users open $\text{URL}_{\text{phishing}}$, believe it to be a legitimate website, and input their sensitive information.

4. Attacker steals sensitive information through the phishing website.

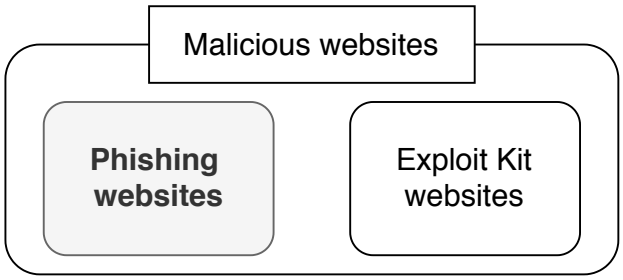


Figure 2-7: The classification of malicious websites.

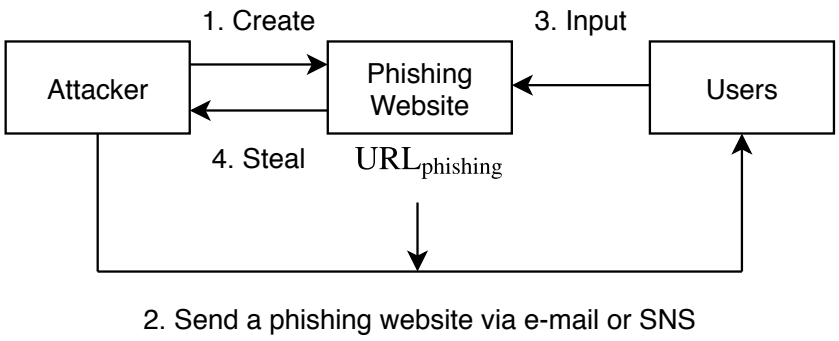


Figure 2-8: Attacker model of a phishing website.

2.2.4 Related works

As mentioned in section 1.5.1, there have been several visual similarity-based phishing detection schemes. In [117], Mao et al. focus on the similarity of CSS (Cascading Style Sheet) which defines the design of the website. However, attackers can create phishing websites without using CSS by embedding background image of targeted legitimate website. In other words, attackers can hide the features. From this point of view, since the information rendered by browsers can be always available, the schemes using screenshot of displayed website have been proposed [118]–[123]. In [118], Dalgic et al. propose to train the screenshots of phishing website targeting same legitimate website by machine learning technique. However, the detection performance depends on the quality of dataset and a large number of dataset is needed. For this issue, we focus on the schemes [119]–[123] which do not require a large number of dataset. These schemes use "signature" which is feature map in layout, position of colors, etc. extracted from targeted legitimate website or phishing websites. Signatures are stored in SDB and the website whose signature is similar to SDB's one is detected as phishing website. However, since they can only detect phishing websites whose signatures are highly similar to SDB's ones, the system administrator has to register many signatures by hand in order to achieve high detection performance.

2.2.5 Shortcomings and requirements for the proposed scheme

The visual similarity-based approaches can only detect phishing websites whose signatures are highly similar to SDB's ones which are registered by the system administrator. Thus, the system administrator has to register multiple signatures in order to achieve high detection performance. However, since there are generally many types of subspecies which target the same legitimate website, the cost of registering signatures becomes very high if the number of subspecies increases. This might incurs a zero-day attack. Generally, the zero-day attack is defined as the situation where an attacker exploits a vulnerability which the administrator or the developer does not notice. In the context of phishing, the zero-day attack is defined as the situation where a new type of phishing website cannot be detected until the system administrator adds the signature for it. A straightforward way to address this issue is to implement an auto signature update mechanism in the system. In signature-based phishing detection schemes, implementing automatic updating indicates adding the signature of detected phishing website to SDB and using it for the next and succeeding detection. By repeating this procedure, it is expected that the detection scope can be expanded. However, aforementioned approaches' signatures are not suitable for auto updating since their similarity can be highly different among targeted legitimate website and subspecies of phishing website targeting that legitimate website.

In order to prove this, we investigate the similarity among the legitimate website and phishing websites in the scheme [122]. Figure 2-9 shows the similarity distribution of Facebook's legitimate and phishing websites in the scheme [122]. As we can see from Figure 2-9, the similarities of phishing websites are distributed. This is because the scheme [122] uses the positions of colors which can be highly different in each phishing website. In this situation, even if the auto updating is applied, the highly similar signatures are gathered in SDB and the scope of detection cannot effectively be expanded. From these point of view, the requirements of signature based phishing detection system are as follows:

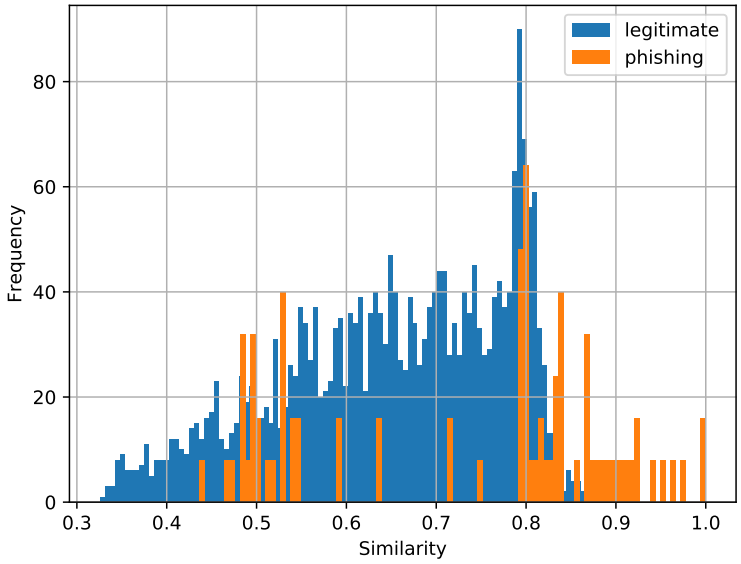


Figure 2-9: The similarity distribution of Facebook' s legitimate and phishing websites in the scheme [122]

- The system automatically updates SDB for the better detection performance and the zero-day phishing attack.
- To automatically update SDB, the signature used for detection has common feature among the targeted legitimate website and most of subspecies of phishing websites targeting that legitimate website.

The scheme which meets the requirements is proposed in Chapter 4.

2.2.6 Summarization of shortcomings

We summarize the shortcomings of the previous schemes as follows.

Sybil detection on SNS

- Seeds are concentrated in specific communities.
- Based on the number of common friends, there are cases where attack edges cannot be pruned

Phishing detection

- The system automatically updates SDB for the better detection performance and the zero-day phishing attack.
- To automatically update SDB, the signature used for detection has common feature among the targeted legitimate website and most of subspecies of phishing websites targeting that legitimate website.

Chapter 3

Trust-based Sybil Detection Scheme on Social Networking Services

3.1 Proposed scheme

In order to solve the shortcomings mentioned in section 2.1.8 and 2.1.9, we propose a seed selection scheme with community detection and graph pruning scheme based on the TA (Trusted Area) which is focused on the density of relationships with nodes. For the first shortcoming discussed in section 2.1.8, we first detect communities in the entire network and then select high degree legitimate seeds from them. This enables to select legitimate seeds uniformly from the network and avoid trust value from being concentrated on the specific communities. For the second shortcoming discussed in section 2.1.9, we recursively calculate the TA and prune the relationships with respect to how much nodes are in the TA. We prune the relationships with high probability when the nodes have less relationships with nodes in the TA and vice versa. This improves the accuracy of pruning attack edges even if the attackers make a large number of common friends. Our scheme consists of these two procedures and we explain each procedure in detail in the following sections. Figure 3-1 shows the

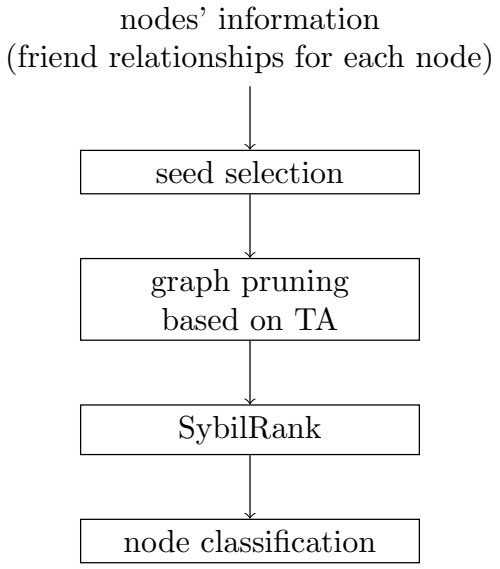


Figure 3-1: Flowchart of the proposed scheme.

flowchart of the proposed scheme. Since the procedure after the graph pruning we propose is the same as SybilRank, we omit the explanation in this section.

3.1.1 Seeds selecting scheme

Prior to graph pruning and SybilRank, we detect communities in SNS with so-called the fast greedy community detection scheme [135]. The reason why we choose the fast greedy community detection scheme is that its computation complexity is as low as $O(n \log^2 n)$ and it effectively works against large scale network like SNS [138]. According to recent work [139], although fast greedy community detection is proposed in 2004, it achieves enough performance compared with latest community detection algorithms. After the communities are detected, seeds are selected from legitimate nodes which have the top $K\%$ highest degree from each detected community. The reason why we choose high degree nodes as seed candidates is that such nodes can effectively distribute trust values toward a large number of friends. However, it could be the case where Sybils are chosen as seeds. In order to avoid this, each seed candidate is manually checked whether it is Sybil or not, which is the same procedure as the previous scheme [66]. It is not a big issue since finding legitimate nodes is further easier than finding Sybils. In the proposed scheme, at most only the number of

detected communities (e.g., 13 communities in [134]) should be inspected. Note that if no top $K\%$ degree node exists, we do not select any seeds from the community. Based on the above idea, the proposed seed selection scheme is formalized with equations as follows. Let C_p denote the nodes in p th community, where $1 \leq p \leq l$. In addition, let d_K denote the top $K\%$ highest degree of these nodes. We then represent seed candidates v_{seedcand_p} with C_p and d_K as

$$v_{\mathsf{seedcand}_p} = \begin{cases} \mathrm{MDN}(C_p) & \text{If } deg(\mathrm{MDN}(C_p)) \geq d_K, \\ \phi & \text{Otherwise,} \end{cases} \tag{3.1}$$

where $\mathrm{MDN}(C_p)$ denotes a function that returns the Maximum Degree Node in the community C_p. However, v_{seedcand_p} could be a Sybil and thus it is manually checked. If a chosen seed candidate is found to be a Sybil, it is not chosen as a seed. Hence, v_{seedcand_p} can be expressed as

$$v_{\mathsf{seedcand}_p} = \begin{cases} \phi & \text{If } v_{\mathsf{seedcand}_p} \text{ is Sybil,} \\ v_{\mathsf{seedcand}_p} & \text{Otherwise.} \end{cases} \tag{3.2}$$

Finally, since we look for seed candidates for each community from C_1 to C_l, $\mathbf{v}_{\mathsf{seeds}}$ which denotes the set of v_{seedcand_p}, can be represented as the union of v_{seedcand_p} in $1 \leq p \leq l$. That is,

$$\mathbf{v}_{\mathsf{seeds}} = \bigcup_{1 \leq p \leq l} v_{\mathsf{seedcand}_p}. \tag{3.3}$$

Figure 3-2 shows an example of seeds selection in the proposed scheme. This figure shows the situation where three communities A, B, and C are detected by the fast greedy community detection and we select one seed from each community. For example, the community A consists of nodes $A_1 - A_7$ and A_4 has the highest degree. In this case, we select A_4 as the seed for community A. The same procedure is applied for B and C. When there exist multiple nodes with highest degree in a community, we randomly select one node as a seed. For example, although community B has the same highest degree nodes B_1 and B_3, B_1 may be randomly selected as a seed.

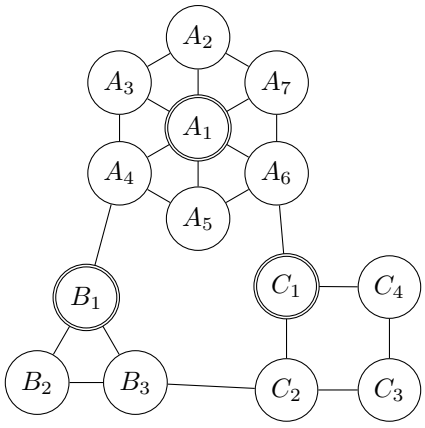


Figure 3-2: Example of seeds selected in the proposed scheme.

3.1.2 Pruning scheme based on trusted area

We prune attack edges based on seeds selected at section 3.1.1. In our scheme, each legitimate seed v_i in $\mathbf{v}_{\mathsf{seeds}}$ has its own TA and v_i's initial TA TA_{v_i} is denoted as

$$\mathrm{TA}_{v_i} = \{v_i, \mathrm{neig}(v_i)\}, \tag{3.4}$$

where $\{x\}$ and $\mathrm{neig}(v_i)$ denote a set of x and v_i's neighbors, respectively. We argue that the possibility that an initial TA includes Sybil nodes is very low because we choose legitimate seeds from nodes with considerably high degree. Such nodes are to be famous rather than ordinary people and they tend to have non-Sybil nodes as friends. For example, Shinzo Abe (the Prime minister of Japan) has more than 4,500 friends in Facebook[1], but it is unlikely that he easily accepts friend requests from strangers[140]. Here, we decide whether to add nodes just around the TA. We define node u's trust value $T_{\mathrm{TA}}(u)$ for calculating TA as

$$T_{\mathrm{TA}}(u) = \frac{deg_{\mathrm{in}}(u)}{deg(u)}, \tag{3.5}$$

where $deg_{\mathrm{in}}(u)$ denotes the number of nodes included in TA. $T_{\mathrm{TA}}(u)$ represents how much nodes are in the TA. The larger this value is, u has the closer relationships with TA and the smaller this values is, u has the more sparse relationships with TA.

[1]As of September 2015.

We argue that the nodes accepting the friend requests from strangers must accept almost all of requests, but the number of such nodes are relatively small in the entire SNS. Thus, our pruning criteria is much robust compared to the previous scheme. If u satisfies

$$T_{\mathrm{TA}}(u) \geq R_{\mathrm{TA}}, \tag{3.6}$$

we add u to TA. Here, R_{TA} is a threshold value to decide whether a node u should be included in a TA. R_{TA} is set to $\frac{2}{3}$ based on "Byzantine generals problem" [141], which is the problem of how much reliable persons are required to correctly communicate information to all persons in a specific group under the situation that some of them incorrectly inform it, namely traitors. In this situation, it has been proven that more than $\frac{2}{3}$ persons must be reliable [141]. We can adapt this notion to calculate TA. In our case, a reliable person is an legitimate node, whereas a traitor is a Sybil node. Since a trust value distributed toward them can be seen as information, more than $\frac{2}{3}$ friends of a node must be reliable when deciding whether to involve him/her in TA. Thus, based on above equations, updating TA_{v_i} is formulated as

$$\mathrm{TA}_{v_i} = \begin{cases} \mathrm{TA}_{v_i} \cup u_{cand} & \text{If } T_{\mathrm{TA}}(u_{cand}) \geq R_{\mathrm{TA}} = \frac{2}{3}, \\ \mathrm{TA}_{v_i} & \text{Otherwise,} \end{cases} \tag{3.7}$$

where u_{cand} denote the friend of nodes in TA. The above procedure is repeatedly executed whenever u_{cand} is included in TA_{v_i} and is finished when no u_{cand} exists. Next, we explain how to prune attack edges. We define the probability of pruning relationships between X and v, that is $P_{\mathrm{cut}}(X, v)$ as

$$P_{\mathrm{cut}}(X, v) = 1 - \frac{T_{\mathrm{TA}}(v)}{R_{\mathrm{TA}}}, \tag{3.8}$$

where X and v denote the nodes in TA which have connection with v and the node in non-TA, respectively. This possibility is higher when u has sparse relationships with nodes in TA and vice versa. Figure 3-3 shows the example of above procedures. In Figure 3-3, the white nodes, grey node, and black nodes represent as nodes in TA,

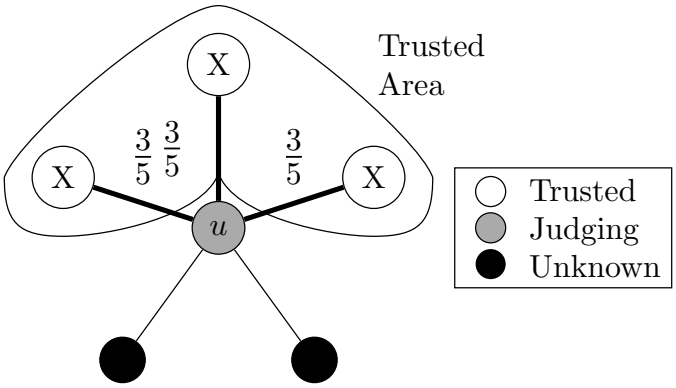
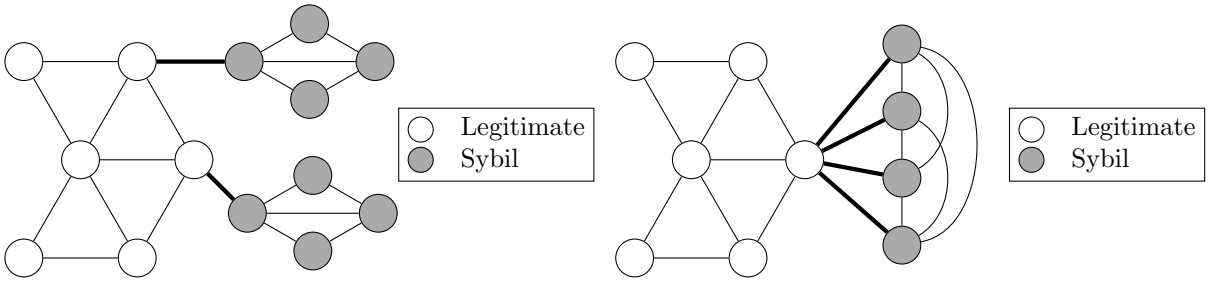


Figure 3-3: Example of judging if u is captured into trusted area.

node to be judged, and unknown nodes, respectively. The unknown nodes are the nodes which have not judged yet. Furthermore, the values beside thick edges denote $T_{\mathrm{TA}}(u)$. In this case, since u has 3 trusted friends ($deg_{\mathrm{in}}(u) = 3$) out of 5 friends ($deg(u) = 5$), the added coefficient $T_{\mathrm{TA}}(u) = \frac{3}{5}$. Hence, Eq. (3.6) does not hold and we finish calculating TA because there are no other nodes around the TA. The pruning probability $P_{\mathrm{cut}}(X, u) = 1 - \left(\frac{3}{5}\right) / \left(\frac{2}{3}\right) = \frac{1}{10}$. The proposed graph pruning scheme can automatically set the pruning probability with respect to closeness to its seed.

3.2 Simulation results

In order to show the effectiveness of the proposed scheme, we evaluate the AUC (Area Under Curve) of ROC (Receive Operating Characteristic) curve [142]. We introduce the ROC curve and the AUC. The ROC curve indicates the performance of a binary classifier system. The curve is created by plotting the TPR (True Positive rate) against the FPR (False Positive rate) at various threshold settings. Here, we define the true positive rate and false positive rate as the ratio that Sybils are accurately classified and legitimate nodes inaccurately classified, respectively. Since the true positive rate and false positive rate depend on the specified threshold T, we want to compare the detection accuracy irrespective of threshold setting. Hence we first calculate ROC with TPR and FPR by varying the threshold and then AUC from ROC.



(a) Attack scenario 1 (b) Attack scenario 2

Figure 3-4: An example of relationships between legitimate nodes and Sybils.

AUC ranges between 0 and 1 and the higher value indicates the better classification algorithm. We use a Facebook dataset [134] for evaluation. Since all nodes in the dataset are legitimate nodes, we add Sybil nodes by assuming two attack scenarios. This situation is based on the previous schemes [66], [67]. Hereafter, we refer the legitimate nodes which have relationships with Sybil nodes as Sybil supporters. The first attack scenario is the same as [67] and we randomly choose 100 Sybil supporters from legitimate nodes and repeat connecting an attack edge between a randomly chosen Sybil supporter and a Sybil node for 200 times i.e. the total number of attack edges $g = 200$. Figure 3-4(a) shows the topology of the first attack scenario. As the second attack scenario, we assume the sophisticated attackers who try to avoid pruning attack edge mentioned in section 2.1.9. In this scenario, we randomly choose 20 Sybil supporters from legitimate nodes and add 10 attack edges for each Sybil supporter from randomly selected Sybil nodes, i.e., the total number of attack edges $g = 200$. Furthermore, in order to increase the number of common friends, Sybil nodes that have relationships with a certain Sybil supporter are connected each other as shown in Figure 3-4(b). In each simulation, we set the number of attackers $n_{\mathrm{att}} = 5$. Table 3.1 shows the simulation parameters.

3.2.1 Overall detection performance

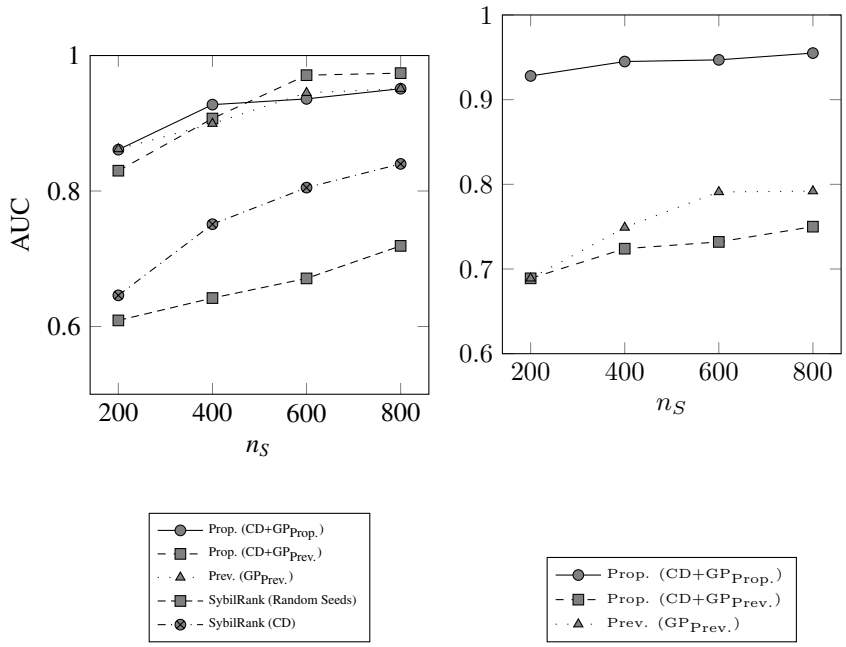
Figure 3-5(a) and 3-5(b) show AUC versus the total number of Sybil nodes in the attack scenario 1 and 2, respectively. Prop. (CD+$\mathrm{GP}_{\mathrm{Prop.}}$), Prop. (CD+$\mathrm{GP}_{\mathrm{Prev.}}$), and Prev. ($\mathrm{GP}_{\mathrm{Prev.}}$) indicate the proposed Community Detection-based seed selecting

Table 3.1. Parameter values used in the simulation.

parameter	value
dataset	Facebook [134]
number of nodes	4039
number of edges	88234
graph model	Random graph with BA model[143] with ave. degree 10
T_s in the previous scheme	1
T_p in the previous scheme	2
n_{att}	5
K	10
simulation tools	R with igraph package[144]

scheme with the proposed Graph Pruning, CD-based seed selecting scheme with the previous Graph Pruning, and the Previous scheme, respectively. In addition, SybilRank (Random Seeds) and SybilRank (CD) denote the original SybilRank [66] and SybilRank with community detection in Figure 3-5(a). We first discuss the result of attack scenario 1 in Figure 3-5(a). In this figure, the both proposed schemes achieve almost same accuracy with the previous scheme. This is because the efficiency of graph pruning approaches is significantly high regardless of the proposed and previous ones against the attack scenario 1. As we can see from this figure, all schemes that use graph pruning achieve high AUC values compared with the schemes without the graph pruning schemes. Furthermore, the proposed graph pruning scheme is also effective against the attack scenario 1. This is because the density of relationships among Sybils and legitimate nodes in the attack scenario 1 is sparse. We can also see that as the number of Sybils gets larger, we can obtain the better AUC regardless of schemes. This is because the total number of attack edges is fixed, as the total number of Sybils gets larger, true positive tends to be higher.

We then discuss the detection accuracy in the attack scenario 2 with Figure 3-5(b). From Figure 3-5(b), we can see that the proposed scheme with community detection and our graph pruning considerably improve AUC against the previous scheme. In this attack scenario, the previous graph pruning scheme cannot prune most of attack edges since Sybil nodes intentionally increase the number of common friends between legitimate nodes and Sybil nodes. On the contrary, the proposed graph pruning



(a) Attack scenario 1 (b) Attack scenario 2

Figure 3-5: AUC versus total number of Sybils n_S ($n_{\text{att}} = 5$).

scheme effectively prunes attack edges since the proposed scheme prunes the edges which have less closeness with trusted nodes. Then, we pay attention to the proposed seed selecting scheme with the previous graph pruning (Prop. (CD+GP$_{\text{Prev.}}$)) and previous scheme (Prev. (GP$_{\text{Prev.}}$)). From these results, we can see the degradation of detection accuracy. This is because the previous graph pruning does not prune attack edges and it is likely to give Sybils trust value because of the proposed seed selecting scheme.

In order to show the case where Sybils cannot be detected, we inspect the trust values Sybils get after distributing trust. As a results, we find that Sybils with attack edges tend to get more trust values than other Sybils. Table 3.2 shows the trust values Sybils get after distributing trust. In this table, AS indicates Attack Scenario and values are normalized by the trust values of Sybils with attack edges. As we can see from Table 3.2, there is difference between Sybils with attack edges and Sybils without attack edges. This is obvious result and that is to say, Sybils near seeds can tend to get more trust values. Although the effectiveness of the proposed graph

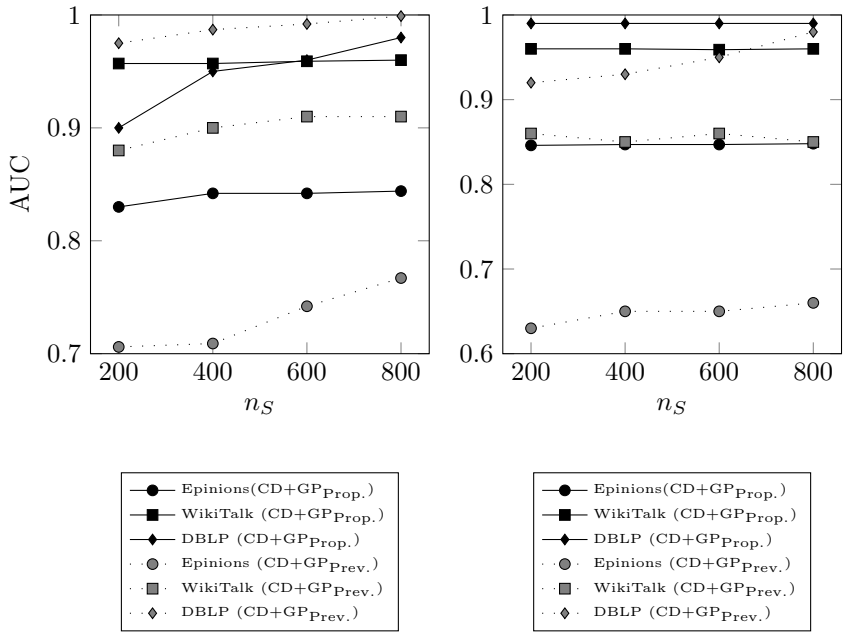
Table 3.2. Trust values Sybils get after distributing trust.

	Trust value	
	Sybils with AE	Sybils without AE
AS1	1.0	0.59
AS2	1.0	0.33

pruning is shown in Figure 3-5, it cannot prune attack edges in the case where they are directly added to seeds. This is because TA is initialized by neighbors of seeds. From these reasons, we can say that the performance of the proposed scheme gets worse when Sybils exist near seeds and especially, Sybils are directly connected to seeds.

From results mentioned above, the previous scheme does not work well especially in the attack scenario 2. We argue this is because the attack scenario 2 destroys homophily mentioned in section 2.1.4. As mentioned in section 2.1.4, although destroyed homophily can be mitigated by machine learning-based approaches, we consider they cannot detect all of Sybils. Under this circumstance, this scenario assumes the realistic situation where some legitimate nodes accept friend requests from unknown accounts in order to increase the number of friends for gaining popularity. Thus, it is meaningful to evaluate the graph-based approaches in the situation where homophily is not held. We believe that the proposed scheme is useful since it effectively detects Sybils in this scenario.

Furthermore, we evaluate our scheme with dataset described in [131]. Although the basic parameter settings are the same as other simulations, we randomly picked 10,000 nodes from the datasets in [131]. Figure 3-6(a) and Figure 3-6(b) show AUC versus the total number of Sybils n_S for other datasets, namely, Epinions, WikiTalk, and DBLP. The attack scenarios 1 and 2 are assumed in Figure 3-6(a) and 3-6(b), respectively. In the Figure 3-6(a), as n_S increases AUC remain stable or slightly gets better for all datasets. This tendency is same as the result in Figure 3-5(a). We can also see that our scheme outperforms the previous scheme in Epinions dataset. In order to clarify this, we analyze Epinions dataset. In the Epinions dataset we sampled, 2,166 nodes have only one friend, and 603 nodes have only two friends. In



(a) Attack scenario 1 (b) Attack scenario 2

Figure 3-6: AUC versus total number of Sybils n_S ($n_{\text{att}} = 5$) in other datasets.

this situation, the previous graph pruning mistakenly prunes many non-attack edges. As a consequence, a trust value may not be effectively distributed to such low degree legitimate nodes and degrades the AUC. In contrast, in the proposed graph pruning, low degree nodes are included in TA because there is few friend other than a friend in TA. As a result, the value of AUC is higher than that of the previous graph pruning scheme. From Figure 3-6(b), we can see that AUC improves for all datasets by using the proposed scheme. This result does not contradict with that of Facebook dataset. Therefore, it can be concluded that the proposed graph pruning scheme is effective against the attack scenario 2.

3.2.2 Evaluation of seed selecting scheme

We compare the proposed seed selecting scheme with the case where each node is randomly chosen as a seed. Since the proposed scheme intends to distribute trust values toward the entire legitimate node, we evaluate the mean and standard deviation of trust vales distributed to each node. If the standard deviation is decreased without lowering the mean value, it can be concluded that trust values are effectively distributed. Table 3.3 and 3.4 show mean and standard deviation of trust values each node obtains. In these tables, if an element of the columns "Community Detection" is T, the proposed graph pruning scheme is used, whereas if it is F, the community detection is not used. As we can see from both tables, legitimate nodes obtain almost the same mean values irrespective of use of community detection while the standard deviation is much decreased by the proposed scheme. In both attack scenarios, the mean trust value Sybils obtain are slightly higher than that of the schemes with the community detection. However, the distinguishability between Sybils and legitimate nodes is higher because the misclassified ratio of legitimate nodes is decreased.

Table 3.3. Statistics of trust values distributed to nodes in the attack scenario 1.

Node Type	Community Detection	Mean	Standard Deviation
Legitimate	F	8.9×10^2	1.6×10^3
Legitimate	T	8.9×10^2	1.3×10^3
Sybil	F	9.6×10	5.7×10
Sybil	T	1.5×10^2	5.6×10

Table 3.4. Statistics of trust values distributed to nodes in the attack scenario 2.

Node Type	Community Detection	Mean	Standard Deviation
Legitimate	F	8.9×10^2	1.8×10^3
Legitimate	T	8.9×10^2	1.3×10^3
Sybil	F	7.7×10	6.9×10
Sybil	T	1.1×10^2	7.5×10

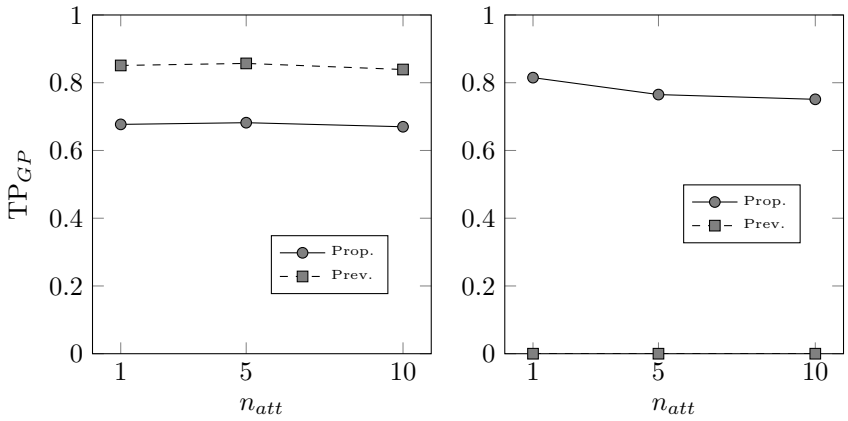
3.2.3 Evaluation of graph pruning

Figure 3-7 and 3-8 show how accurately each scheme prunes attack edges. We define TP_{GP} and FP_{GP} as the ratio that the attack edges are accurately pruned and non-attack edges are inaccurately pruned, respectively. Figure 3-8 shows the false positive rate in graph pruning FP_{GP} versus the number of attackers n_{att} and a numeric above a point is the average number of pruned edges. We first discuss the result in the attack scenario 1. From Figure 3-7(a) and 3-8(a), we can observe that the previous scheme effectively prunes attack edges with high TP_{GP} and low FP_{GP}. This is because the number of common friends between legitimate nodes and Sybil nodes is small and there are few legitimate nodes that do not have common friends. The proposed graph pruning scheme degrades TP_{GP} and FP_{GP} compared with the previous one. This is natural since attack scenario 1 is the situation where the previous graph pruning effectively works. Although FP_{GP} seems to be high in both scheme, it can be acceptable because the number of entire edges is about 90,000 as shown in Table 3.1, and the average number of total cut is relatively small.

In attack the scenario 2, from Figure 3-7(b) and 3-8(b), the previous scheme cannot accurately cut attack edges. This is because the number of common friends between legitimate nodes and Sybil nodes are increased in this scenario. On the other hand, in the proposed scheme, TP_{GP} is much higher than the previous one and we can say that graph pruning with TA works well in the attack scenario 2. Since attack edges are concentrated in specific nodes, the proposed graph pruning scheme can easily detect attack edges by checking nodes' closeness among trusted nodes.

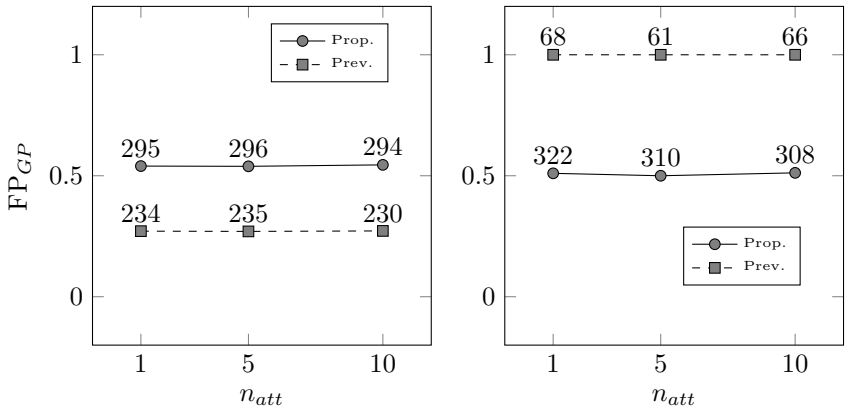
In both attack scenarios, the proposed graph pruning scheme cannot prune attack edges if they are directly added to seeds (see section 3.2.1). In addition to that, the proposed graph pruning cuts attack edges by using probability defined in Eq.(3.8) in order to prevent many legitimate relationships on boundary of TA from being pruned. Thus, there is the case where the proposed graph pruning scheme cannot prune attack edges even if they are detected by calculating TA.

Furthermore, we evaluate the calculation time of the proposed graph pruning.



(a) Attack scenario 1 (b) Attack scenario 2

Figure 3-7: TP$_{GP}$ versus n_{att}.



(a) Attack scenario 1 (b) Attack scenario 2

Figure 3-8: FP$_{GP}$ versus n_{att}.

Table 3.5 shows the calculation time and TA's properties. In Table 3.5, AS, AVG, and STD indicate Attack Scenario, AVeraGe, and STandard Deviation, respectively. Since the proposed graph pruning scheme cuts attack edges by recursively calculating TA, it is expected that the calculation time increases per iteration. However, as shown in Table 3.5, the standard deviation of all iterations time is small regardless of attack scenario. In other words, it takes almost the same time in each iteration. This is because the number of nodes included in TA is relatively small as shown in Table 3.5's right columns. Since initial TA is composed of neighbor nodes of seeds

Table 3.5. Calculation time and trusted area's properties.

	Iteration			Number of nodes in TA	
	Time (s)	AVG # of iter.	STD of all iter. time	Before expansion	After expansion
AS1	0.288	2	0.005	3718.9	3742.8
AS2	0.938	2	0.05	3718.1	3742.9

which are representative nodes in each community, the nodes which do not belong to initial TA tend not to have dense relationships with each community. Since the nodes which are not near seeds but have dense relationships only become the target of calculation, the calculation time is not large. Note that there may be the case where we should stop calculating TA to some extent (e.g. for real social network). In that case, attack edges can be pruned since the nodes which are not added to TA for a long time have sparse relationships with most of legitimate nodes.

3.2.4 Limitation and discussion

In order to show the limitation of the proposed scheme, we evaluate the classification performance with the attack scenario described in [131]'s Fig.1. In this scenario, almost all of Sybils' friends are legitimate nodes and the relationships among Sybils are few. This scenario corresponds to the real case where a legitimate node mutates into Sybil node via account trading. Since there is no formal definition of this attack in [131], we assume the situation where Sybils' friends are only legitimate nodes. In this evaluation, we add 200 Sybils and randomly choose 100 Sybil supporters from legitimate nodes. Each Sybil adds attack edges to k Sybil supporters. Since [131] shows that SybilRank's detection accuracy degrades in this attack scenario, we evaluate the community detection schemes with the previous and proposed graph pruning schemes. Figure 3-9 shows AUC versus the number of attack edges per Sybil. As we can see from this figure, the larger k is, the worse AUC is obtained in both graph pruning schemes. This is because attack edges are not accurately pruned in both schemes. In the previous one, the AUC is almost one when $k = 1$. This is because there are no common friends among Sybils and legitimate nodes and the

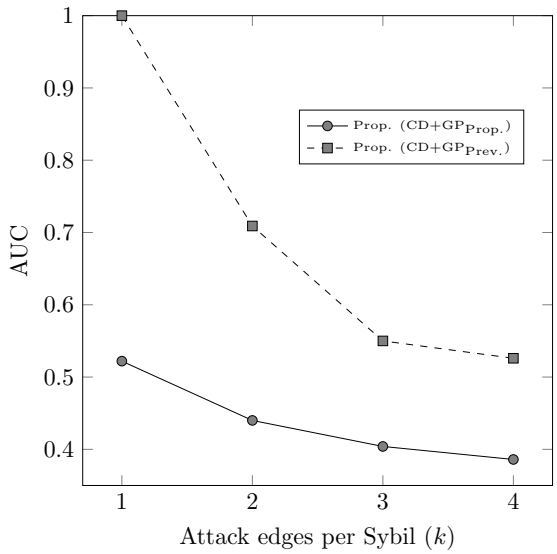


Figure 3-9: AUC versus the number of attack edges per Sybil.

previous scheme completely prunes attack edges when $k = 1$. However, as k gets larger, the number of common friends among Sybils and legitimate nodes increases. As a result, the previous scheme cannot prune attack edges, and AUC gets worse. In contrast, the AUC of the proposed scheme is considerably bad regardless of k. This is because all Sybils' friends are legitimate nodes and thus they are involved into TA with high probability. From this result, our scheme cannot deal with this attack. Finally, we summarize the situation where the proposed scheme effectively works as follows:

1. Sybils following the model described in section 2.1.3.

2. Sybils most of whose friends consist of Sybils.

The proposed scheme cannot detect Sybils which do not correspond to above two cases. Furthermore, the proposed scheme cannot deal with the dynamic situation described in [145]. In such cases, machine learning-based approaches might effectively work. Recently, Lê et al. propose a hybrid approach using graph based scheme and features of accounts' properties [146]. Since results of that scheme show that both graph based detection and accounts' properties effectively work, it might deal with many types of Sybils.

3.3 Conclusion and future works

In this chapter, we have proposed a Sybil nodes detection scheme with robust seed selection and graph pruning on SNS. The proposed scheme is composed of two proposals. The first one is a seed selecting scheme by detecting communities and choosing seeds from them. The second one is a graph pruning scheme that considers trusted area. We model SNS and Sybil nodes by two scenarios for evaluation. In the first scenario, we model the general SNS and Sybils from the homophily's point of view. The proposed scheme achieves almost the same detection accuracy of the previous scheme. In the second scenario, we assume more realistic case where some legitimate nodes accept friend requests from Sybil accounts in order to increase the number of friends for gaining popularity. The proposed scheme outperforms the previous scheme by about 20 % in the metric of AUC. Since the second scenario is the attack which destroys entire SNS's homophily, most of graph-based approaches might not effectively work. The strong point of the proposed scheme is that it effectively works especially in the situation where most of Sybils' friends consist of other Sybils like second scenario. As future works, in order to achieve lower errors and effective results in the real environment, we should consider the SNS's dynamic scenario and using other features of accounts in addition to graph-based properties and dynamic.

Chapter 4

Phishing Detection Scheme using Hue Information with Auto Updating Database

4.1 Proposed scheme

In order to meet the requirements mentioned in section 2.2.5, we propose a novel visual similarity-based phishing detection scheme using hue information with auto updating database. Since a phishing website is created based on targeted legitimate website or other subspecies whose hue information is similar each other, many phishing websites can be exhaustively detected by tracing similar colored subspecies. The hue information includes the common feature among the targeted legitimate website and subspecies of phishing websites, which meets the requirement for auto updating SDB. Based on this notion, the proposed scheme detects a new phishing website which has similar hue information to already detected phishing websites. By repeating this procedure and automatically updating SDB, the detection scope can be effectively expanded. To avoid the misdetection of legitimate websites which have similar hue information to SDB's ones, the proposed scheme utilizes the fact that the combination of used colors is hard to be similar among legitimate and phishing websites.

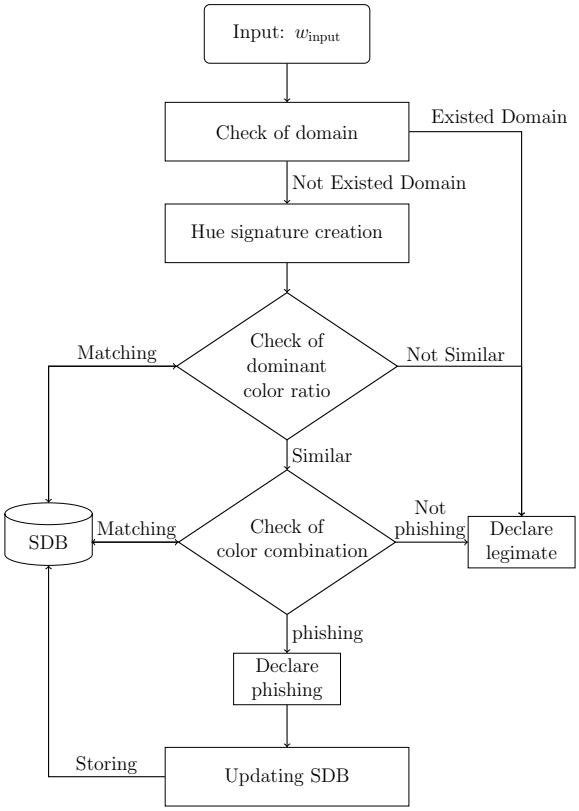


Figure 4-1: Overview of the proposed scheme.

As shown in Figure 4-1, the detection procedure of the proposed scheme consists of five phases, which are "check of domain", "hue signature creation", "check of dominant color ratio", "check of color combination" and "updating SDB". Note that the system administrator stores targeted legitimate website's hue signature(s) to SDB in the initial state. Let w_{input} denote the input website. In the first phase, if w_{input}'s URL contains the same domain information as the targeted website's one, it is judged as legitimate. In the second phase, our scheme accesses w_{input}'s URL and takes its screenshot. The hue signature $S(w_{\text{input}})$ is created from that screenshot. In the third phase, $S(w_{\text{input}})$ is compared with each hue signature stored in SDB. If there do not exist any similar signatures in SDB, w_{input} is declared as legitimate and the detection procedure is finished; otherwise the next fourth phase is started. In the fourth phase, if $S(w_{\text{input}})$ has the sufficiently same colors compared with signatures in SDB, w_{input} is declared as a phishing website; otherwise a legitimate website. In the fifth phase, if w_{input} is judged as a phishing website in the previous phase, the proposed scheme

adds $S(w_{\text{input}})$ to SDB if it is not completely the same as SDB's one. In the following sections, we detailedly explain each phase.

4.1.1 Check of domain

In the first phase, the proposed scheme checks if the w_{input} is the targeted legitimate website itself or not. If this phase does not exist, the targeted legitimate website can be judged as a phishing website in the case where itself is an input. This is realized by comparing the entity name of w_{input}'s domain and that of the targeted legitimate website. For example, consider the case where the system has SDB of Facebook (www.facebook.com) and the input website's domain is "www.faceboooook.co.jp". The entity names of them are "facebook" and "faceboooook". In this case, since the proposed scheme cannot judge if input website is legitimate or not by only using the domain information, the detection process goes to the next phase.

4.1.2 Hue signature creation

At the beginning of this phase, the proposed scheme accesses w_{input}'s URL and takes a screenshot of w_{input} for creating a hue signature. To reduce the computational cost, we resize that screenshot to a 100×100 image. Let O denote the matrix of the resized screenshot image. We define the component in the i th row and j th column of O as

$$o_{ij} = (r^o_{ij}, g^o_{ij}, b^o_{ij}), \tag{4.1}$$

where r^o_{ij}, g^o_{ij}, and b^o_{ij} denote o_{ij}'s color value of red, green, and blue. In order to eliminate colors which almost all of websites have, we remove grayscale colors from O. When each element in o_{ij} is converted to the grayscale color, it is converted according to

$$Y_{ij} = 0.299r^o_{ij} + 0.587g^o_{ij} + 0.114b^o_{ij}. \tag{4.2}$$

Suppose P is a grayscale expression of O and we can express the component in the i th row and j th column of P as

$$p_{ij} = (Y_{ij}, Y_{ij}, Y_{ij}). \tag{4.3}$$

Let M denote the matrix whose elements of grayscale are eliminated from O. The component in the i th row and j th column of M is expressed as

$$m_{ij} = \begin{cases} (r^o_{ij}, g^o_{ij}, b^o_{ij}) & \text{if } \|o_{ij} - p_{ij}\| > D_{\text{gray}}, \\ \text{Null} & \text{otherwise}, \end{cases} \tag{4.4}$$

where D_{gray} denotes the threshold value of this procedure. Since there can be too many color patterns in M, we degrade each color to N levels. Let M' denote degraded version of M. Each element in M' is expressed as

$$m'_{ij} = \begin{cases} \text{Null} & \text{if } m_{ij} = \text{Null}, \\ (\lfloor \frac{r^o_{ij}}{N/256} \rfloor, \lfloor \frac{g^o_{ij}}{N/256} \rfloor, \lfloor \frac{b^o_{ij}}{N/256} \rfloor) & \text{otherwise}. \end{cases} \tag{4.5}$$

Here, the set of colors included in M' is expressed as

$$C = \text{unique}(\{m'_{ij} | m'_{ij} \neq \text{Null}\}), \tag{4.6}$$

where the function "unique" returns non-duplicated set of argument. We define the set of used colors C_{used} as

$$C_{\text{used}} = \{c_k | 0 \leq k \leq |C|\}, \tag{4.7}$$

where c_k and $|C|$ are the k-th most occupied color included in M' and the number of elements included in C. Suppose n_{c_k} denotes the number of c_k appeared in M' and the hue signature of w_{input} is represented as

$$S(w_{\text{input}}) = \{(c_k, \frac{n_{c_k}}{\Sigma_k n_{c_k}}) | 0 \leq k \leq |C_{\text{used}}|\}, \tag{4.8}$$

where $\frac{n_{c_k}}{\Sigma_k n_{c_k}}$ indicates the ratio of color c_k in that signature.

4.1.3 Check of dominant color ratio

In this phase, the proposed scheme calculates the similarity of $S(w_{\text{input}})$ and each signature stored in SDB in terms of the dominant color. If at least one signature in SDB is similar to $S(w_{\text{input}})$, the proposed scheme goes to the next check of color combination phase. We use EMD (Earth Mover's Distance) [128] as a metric of similarity. EMD indicates the distance between two distributions. In order to calculate EMD, weight vectors and cost matrix of two distributions are needed. In this case, the weight vectors are two signatures' ratio of colors and the cost matrix is euclidean distance of each color pair of two signatures. Here, in order to extract dominant colors from the signature, we limit the number of used color by introducing D_{color}. For example, we suppose that $S(w_{\text{input}})$ and $S(w)$ stored in SDB are compared. Let $S(w)$ denote $\{(d_t, \frac{n_{d_t}}{\Sigma_t n_{d_t}}) | 0 \leq t \leq T\}$. The used part of signature is limited by D_{color}. Thus, the ranges of k and t are $0 \leq k \leq k_{\text{top}} = \min(|C_{\text{used}}|, D_{\text{color}})$ and $0 \leq t \leq t_{\text{top}} = \min(T, D_{\text{color}})$. The weight vectors are $(\frac{n_{c_0}}{\Sigma_k n_{c_0}}, \frac{n_{c_1}}{\Sigma_k n_{c_1}}, \ldots, \frac{n_{c_{k_{top}}}}{\Sigma_k n_{c_{k_{top}}}})$ and $(\frac{n_{d_0}}{\Sigma_t n_{d_t}}, \frac{n_{d_1}}{\Sigma_t n_{d_1}}, \ldots, \frac{n_{d_{t_{top}}}}{\Sigma_t n_{d_{t_{top}}}})$, and the component in the k-th row and t-th column of the cost matrix is $|c_k - d_t|/N$. If the value of EMD calculated by these values does not exceed the threshold value D_{EMD}, w_{input} is declared as legitimate. Note that EMD values indicates distance between two distributions. In Figure 4-2, the similarity is calculated by normalized EMD.

4.1.4 Check of color combination

The proposed scheme checks if $S(w_{\text{input}})$ has the sufficiently same colors of signatures in SDB. If the condition of the color combination is fulfilled, w_{input} is declared as phishing. We leverage Jaccard similarity coefficient[147] which is often used in calculating the similarity of two sets to compare the color combination. With Jaccard similarity, the condition of color combination of two signature ($S(w_{\text{input}})$ and $S(w)$)

is represented as

$$\frac{|\{c_k|0 \leq k \leq |C_{\text{used}}|\} \cap \{d_t|0 \leq t \leq T\}|}{|\{c_k|0 \leq k \leq |C_{\text{used}}|\} \cup \{d_t|0 \leq t \leq T\}|} > D_{\text{comb}}, \tag{4.9}$$

where D_{comb} denotes threshold value for the color combination.

4.1.5 Updating signature database

Finally, we store $S(w_{\text{input}})$ to SDB if there do not exist any completely same signature in SDB. Stored signatures are used for the next detection. In this situation, we consider the initial signature should be searched first when detection process is executed. This is because the initial signature is selected by the system administrator and the probability of contributing detection is high. Thus, in order to decide the searching order, we introduce "rank" for each signature stored in SDB. The rank of initial signature is zero and the signature which is similar to the signature whose rank is r has rank $r + 1$. Note that the similarity in this phase corresponds the color ratio mentioned in Section 4.1.3. Suppose $Rank(S(w))$ denotes the rank of $S(w)$ stored in SDB and $S(w_{\text{input}})$ is judged as similar to $S(w)$ in the phase of check of color ratio. The rank of $S(w_{\text{input}})$ is calculated as

$$Rank(S(w_{\text{input}})) = Rank(S(w)) + 1. \tag{4.10}$$

4.2 Simulation results

In order to demonstrate the effectiveness of the proposed scheme, we compare the proposed scheme with the scheme [123] which uses logo detection and is extended work of [122]. This is because that scheme is signature-based phishing detection and uses the information extracted from websites' screenshot. We call that scheme "previous scheme". Following [123], we get legitimate websites from Google's search result whose query keywords are bank, biology, car, Chinese, company, computer, English, entertainment, government, health, Hong Kong, house, Linux, money, movie,

network, phishing, regional, research, science, spam, sport, television, university, Web, and Windows. In each simulation, we randomly choose twenty categories as legitimate dataset and use the average of ten simulations as a simulation result. The dataset of phishing websites are collected from Phishtank [148] from Nov. to Dec. in 2018. In the collected phishing websites, we use the ones targeting famous legitimate websites, which are Facebook, Paypal and BoA (Bank of America). This is because these three kind of phishing websites have different features: their hue information is clearly different, and the appearances of Facebook's phishing websites are highly various and those of BoA's phishing websites are fewer various. Those of Paypal are medium degree.

The metric of the evaluations are TPR (True Positive Rate), and FPR (False Positive Rate) defined as

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{4.11}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}}, \tag{4.12}$$

where TP, TN, FP, and FN denote the number of True Positive, True Negative, False Positive, and False Negative, respectively. We evaluate the proposed scheme with these metrics by two scenarios which are the number of initial signatures is one and five. First of all, we have to decide suboptimal threshold value of D_{EMD} and D_{comb} for each of three kinds of websites since the optimal threshold value might be dependent on a dataset and is difficult to be decided. We conduct a grid search when the number of initial signature is one. In this simulation, the suboptimal values of D_{EMD} and D_{comb} are decided when the lowest FPR and the highest TPR are simultaneously achieved. The threshold value of the scheme [122] is similarly decided. Table 4.1 shows summarization of D_{EMD} and D_{comb}. Other parameters are shown in Table 4.2. For all simulations, we use a desktop computer which has Intel Core i7 3.5 GHz processor and 16 GB memory. Among all figures of results, Prop. represents the proposed scheme and Prev. represents the previous scheme.

Table 4.1. Summarization of D_{EMD} and D_{comb}.

Target legitimate website	D_{EMD}	D_{comb}
Facebook	0.12	0.15
Paypal	0.20	0.40
BoA	0.20	0.35

Table 4.2. Simulation parameters.

Name	Value
N	10
D_{gray}	20
D_{color}	10
Phishing dataset	Phishtank [148]
Number of Facebook phishing	656
Number of Paypal phishing	1295
Number of Bank of America phishing	992
Number of each simulation	10 (The result value is average)
Total number of legitimate websites	2435 (26 categories)
The tool of calculating EMD	Python with POT package [129]
The tool of image processing	Python with OpenCV [149]

4.2.1 Hue information similarity among phishing websites and a target website

Figure 4-2 shows the similarity distribution of Facebook' s legitimate and phishing websites in the proposed hue signature. As we can see from Figure 4-2, the similarities of phishing websites are concentrated compared with Figure 2-9. This indicates that hue signature has common feature among the targeted legitimate website and sub-species of phishing websites targeting that legitimate website. Thus, we argue that it meets the requirement of auto updating. Note that EMD values indicates distance between two distribution. In Figure 4-2, the similarity is calculated by normalized EMD.

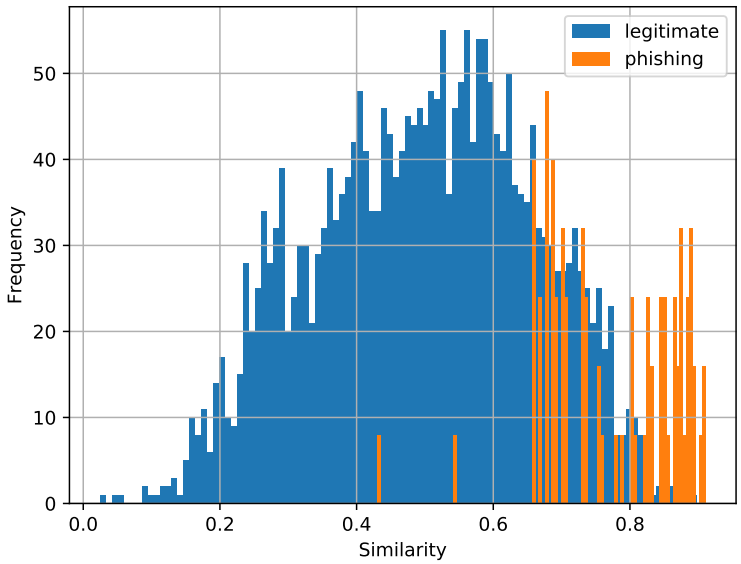
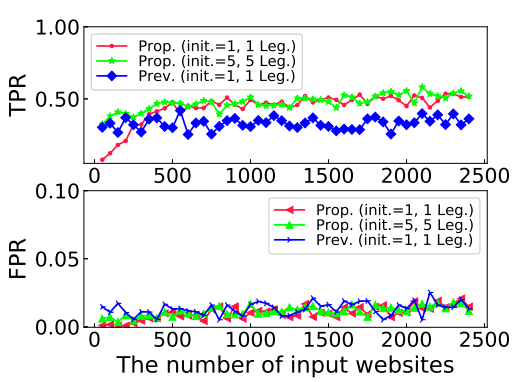


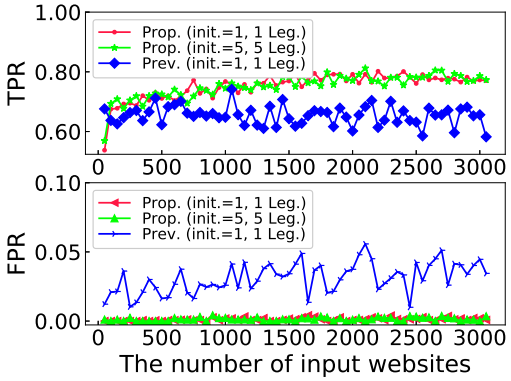
Figure 4-2: The similarity distribution of Facebook's legitimate and phishing websites in our hue signature.

4.2.2 Comparison of signature's suitability for auto update

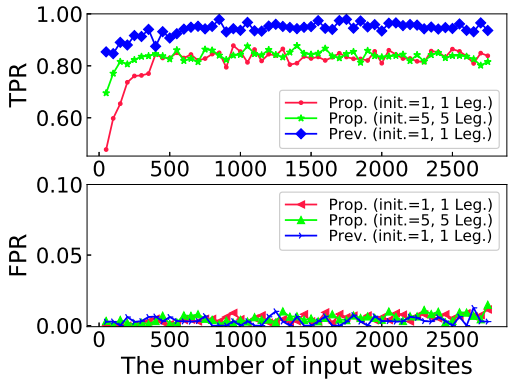
In order to show the proposed hue signature's suitability for auto update, we implement a naive auto update mechanism to the previous scheme. That is, the websites' signatures which are judged as phishing websites are added to SDB in the previous scheme. In this simulation, we use initial signatures which are created from subpages, e.g., "top page" , "login/signup page" , "forget password/help page" , "security page" and so on, of targeted legitimate websites. In order to show the expansion of the scope of detection, we evaluate true positive rate and false positive rate versus the number of input website. Figure 4-3 shows true positive rate and false positive rate versus the number of input websites. In these figures, init. indicates the number



(a) TPR and FPR in Facebook dataset



(b) TPR and FPR in Paypal dataset



(c) TPR and FPR in BoA dataset

Figure 4-3: TPR and FPR versus the number of input websites when all of initial signatures are created from subpages of targeted legitimate website. Prev. implements auto updating SDB.

of initial signatures of each scheme. Moreover, 1 Leg. and 5 Leg. indicate that one and five initial signatures which are created from the targeted legitimate website are used, respectively. As we can see from Figure 4-3(a), Figure 4-3(b), and Figure 4-3(c), both Prop.(init.= 1) and Prop.(init.= 5) can increase true positive rate as the number of input websites increases. This indicates the hue signature is suitable for the auto updating signature. Especially, true positive rate rapidly increases when the number of input website is smaller than 500. This is because similar colored phishing websites are concentrated and the expansion of detection scope is fast. Comparing Prop.(init.= 1) with Prop.(init.= 5), it can be observed that the detection scope of init.= 5 rapidly expands compared with init.= 1 when the number of input websites is small. This is reasonable because, in the initial state, the number of detectable phishing websites slightly increases compared with init.= 1. This can be a merit in terms of detecting zero-day phishing attacks. However, we cannot see significant differences between Prop. (init.= 1) and Prop.(init.= 5) in terms of true positive rate. There are two reasons for this result: (1) in case all of initial signatures are created from subpages of the targeted legitimate website, the ratio of dominant colors is almost the same among those signatures and thus a detection scope of one signature eventually covers that of the others, and in addition to that, (2) in case the system administrator selects a single initial signature, the signature is created from targeted legitimate website's subpage which tends to be especially targeted by attackers and thus effectively contributes to the detection. As a result, the detection scope of multiple initial signatures created from subpages of the targeted legitimate website becomes almost the same as that of single initial signature. Thus, although using many legitimate websites' signatures as initial signatures has a merit in terms of rapid detection scope expansion, it does not significantly improve the detection performance. The false positive rates do not increase in all cases. This is because the check of color combination effectively works. On the other hand, as we can see from Figure 4-3(a) and Figure 4-3(b), the true positive rate does not increase in the previous scheme with auto updating. This is because there are various appearances of phishing websites in Facebook and Paypal dataset. In other words, the previous scheme which uses the positions of colors

cannot effectively expand the scope of detection to cover the various subspecies of phishing website with different positions of colors. Thus, the true positive rate does not increase in spite of using auto update mechanism. However, in Figure 4-3(c), we can see the auto update mechanism effectively works in the BoA dataset. This is because the BoA dataset includes fewer various appearances of phishing websites. Figure 4-4 shows an example of the phishing website targeting BoA. Almost all of phishing websites in BoA have BoA's logo and red horizontal bar in the upper side of their screenshot. Although this commonness existed in subspecies of BoA's phishing websites enables auto update in the previous scheme, it might not detect phishing website if the attacker creates a phishing website with highly different color positions. Although it is forecasted that the false positive rate becomes higher with the growth of SDB, the previous scheme achieves low false positive rate. This is because the previous scheme uses logo detection which can effectively reduce false positive rate.

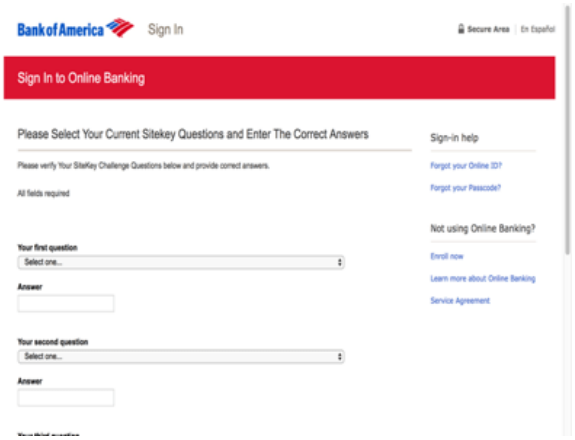
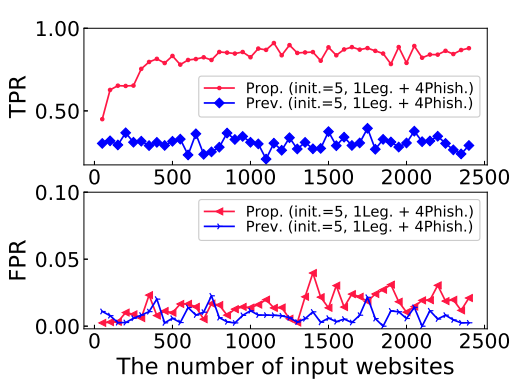


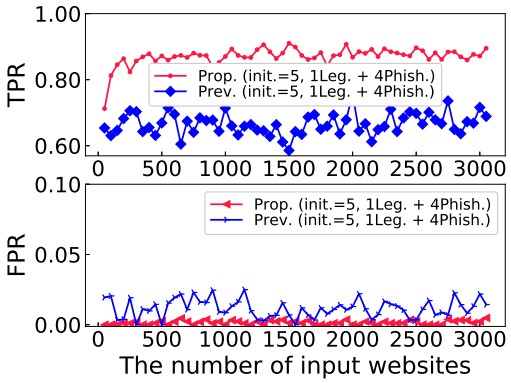
Figure 4-4: An example of phishing website targeting BoA.

4.2.3 True positive rate and false positive rate versus the number of input websites

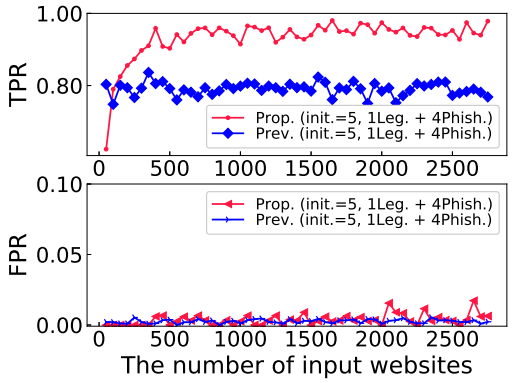
In order to compare the performance of the proposed scheme with that of previous scheme without auto updating, we evaluate true positive rate and false positive rate versus the number of input websites in the situation where the number of initial signatures is five. In addition to legitimate website's signature, we add four initial signatures created from the phishing websites' screenshot which are not detected in Section 4.2.2. Figure 4-5(a), 4-5(b), and 4-5(c) show true positive rate versus the number of input websites in each dataset. As we can see from Figure 4-5, the basic tendency is similar to the result shown in Figure 4-3. The different point is the true



(a) TPR and FPR in Facebook dataset

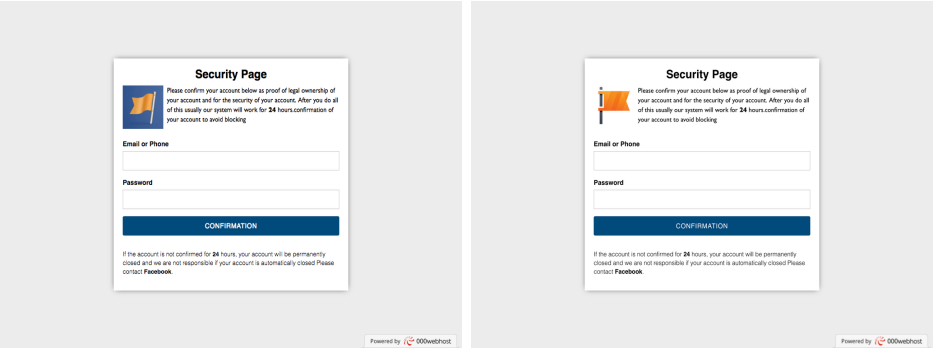


(b) TPR and FPR in Paypal dataset



(c) TPR and FPR in BoA dataset

Figure 4-5: TPR and FPR versus the number of input websites when initial signatures created from phishing websites are included in the initial SDB.



(a) Added initial signature (b) Newly detected phishing

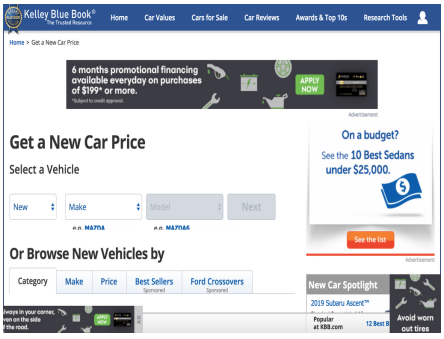
Figure 4-6: An example of Facebook phishing website.

positive rate of the proposed scheme. By using additional only four phishing signatures, the proposed scheme brings a great improvement in all dataset. This is because a hue signature's scope of detection is wide and this fact effectively works especially when using multiple initial phishing signatures. In other words, adding initial phishing signatures leads to an effective expansion of the detection scope. Especially, in the Facebook dataset, it can be observed that about 30% of improvement compared with Figure 4-3(a). This is because the types of Facebook's phishing websites are various and it is difficult for legitimate signatures to cover all types of them. For example, Figure 4-6 shows one of added initial signature and newly detected phishing website. Note that, this added signature is not detected when the initial signatures are created from legitimate website because of its different atmosphere from the legitimate Facebook website. By adding this screenshot as an initial signature, the proposed scheme can detect phishing websites like Figure 4-6(b). We consider the performance can be improved if more phishing signatures are added in the initial state. With regard to the previous scheme, by the same reason mentioned in Section 4.2.2, the tendency of the true positive rate and the false positive rate are similar to Figure 4-3 except for BoA dataset. The performance of the previous scheme in BoA dataset degrades compared with Figure 4-3(c) and the proposed scheme outperforms it. This result also indicates the previous scheme can only detect phishing websites which are highly similar to the signatures in SDB. Note that the performance can be better if there is

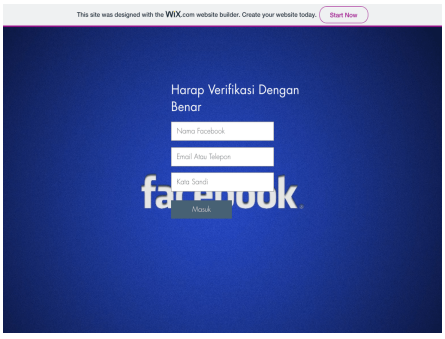
a sufficient number of initial signatures. However the cost to realize it is very high. Moreover, since the Facebook phishing websites often have no logo of Facebook, the true positive rate of the previous scheme is about 0.3, which is especially low.

4.2.4 False positive and false negative analysis

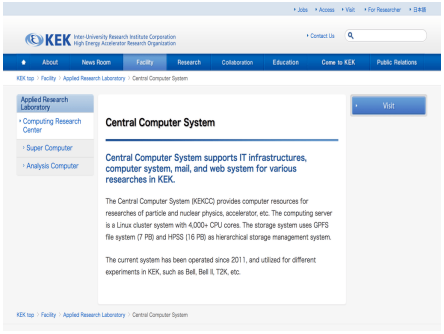
In this section, we analyze the websites resulted in the false positive and false negative. The false positive occurs in each dataset although the false positive rate of the proposed scheme is small as Figure 4-5 shows. There are two types of false positive in each dataset. The first type is the legitimate website which are detected as phishing by initial signatures. The second type is occurred when the scope of detection is over expanded. Unfortunately, it is not easy for the proposed scheme to completely avoid the second type of false positive, we discuss the first type of false positive. Figure 4-7(a), Figure 4-7(c), and Figure 4-7(e) show the first type of false positive of each website. Since our scheme only uses the dominant color information and its combination, they are classified as phishing. For example, in Figure 4-7(a), it uses Facebook's purple color on the upper side and light green which is used in the button of Facebook's legitimate website. In Figure 4-7(e), the website is judged as a phishing website since it uses completely same red and blue of the legitimate BoA and most of other grayscale colors are ignored in our hue signature. However, it is not a critical problem in the real environment because the scheme administrator has only to register their domain to the white list. We consider the cost of that action is not high since the false positive rate of the proposed scheme is small. Figure 4-7(b), Figure 4-7(d) and Figure 4-7(f) show false negative of each website. The Facebook phishing website shown in Figure 4-7(b) does not use purple but bluish color in the background image. That phishing website's atmosphere is far from legitimate Facebook. We can see the logo is hidden by the input area and login button. We consider this is an example way of logo detection avoidance. The Paypal phishing website shown in Figure 4-7(d) only uses grayscale colors while the legitimate Paypal uses bluish colors. The proposed scheme cannot detect this type of phishing website and other detection schemes are needed. The BoA phishing website shown in Figure 4-7(f)



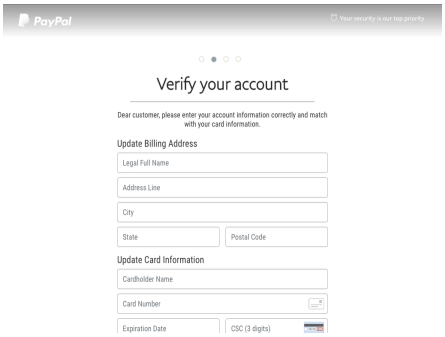
(a) FP of Facebook



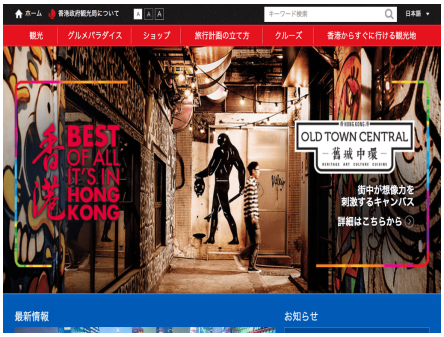
(b) FN of Facebook



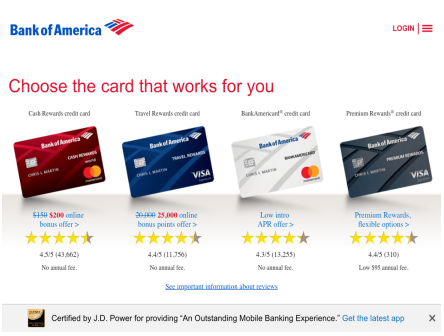
(c) FP of Paypal



(d) FN of Paypal



(e) FP of BoA



(f) FN of BoA

Figure 4-7: Example of FP and FN in each target website.

uses flash contents. Although in this case the colors of displayed credit card interrupt the detection, we might detect it if more types of phishing website are input. This is because other color factors are similar to the SDB's signature.

4.2.5 Evaluation of each phase

In order to show the effectiveness of the check of dominant color ratio and the color combination, we investigate how input websites are judged in each phase. Table 4.3 shows how input website are judged in each phase. As we can see from this table, almost all of the legitimate websites are correctly judged as legitimate and many phishing websites are judged as similar in terms of dominant color. The legitimate websites suspected by this phase are relieved in the phase of check of color combination in most cases. However, we can see the phase of color combination does not work effectively in Facebook's case. We consider there are two reasons for this. The first reason is that there exist many legitimate websites whose color is similar to Facebook. The second reason is that the number of websites which go to the phase of the color combination is small. On the other hand, most of the phishing websites which go to the phase of the color combination are accurately detected as phishing website.

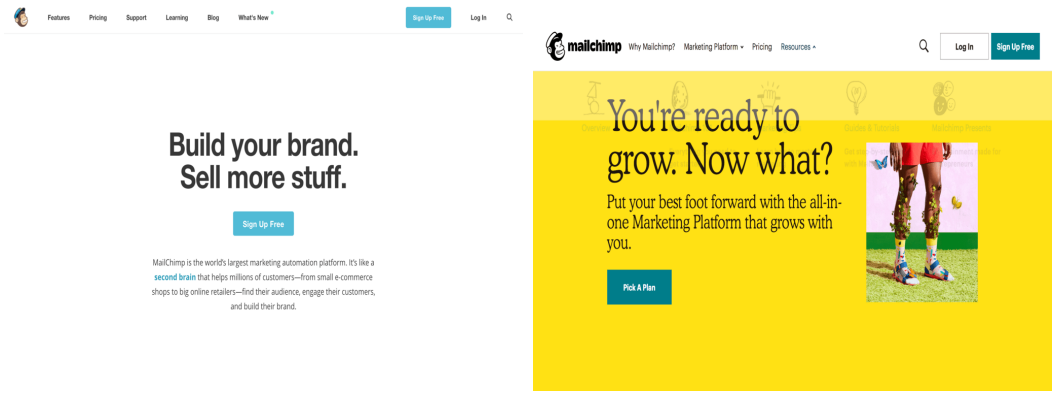
Table 4.3. The average number of websites which are judged in each phase.

Dataset	Facebook		Paypal		BoA	
Label	leg.	phish.	leg.	phish.	leg.	phish.
# of Dataset	1797.0	656.0	1802.0	1295.0	1817.0	992.0
Check of color ratio $\neq$ similar	1765.8	108.2	1674.1	116.3	1754.8	159.6
Check of color ratio= similar	31.5	547.8	128.7	1178.6	62.8	832.3
Check of color comb. $\neq$ phish.	2.6	9.6	125.9	51.9	53.9	22.4
Check of color comb. = phish.	28.9	538.2	2.8	1126.6	8.8	809.9

4.2.6 Limitations

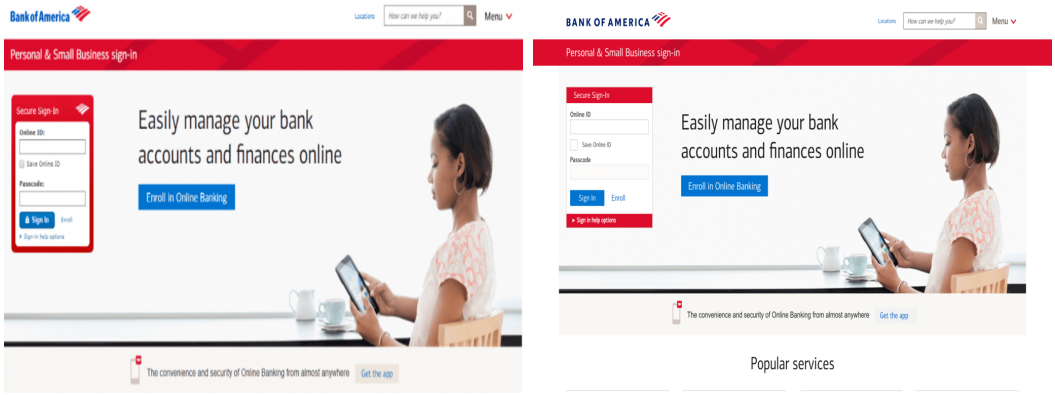
Impact of legitimate website's redesign

There are cases where the proposed scheme cannot detect the phishing website mimicking the renewal design, and it can cause an increase of false negatives (note that the false positive keeps same degree since the old signatures stored in SDB are not changed by redesign.). For example, Mailchimp, which is one of legitimate web service, has been drastically redesigned in 2018. Figure 4-8(a) and Figure 4-8(b) show the redesign of Mailchimp's top page (The old version of website can be obtained



(a) Before redesign (b) After redesign

Figure 4-8: The example of a legitimate website's redesign in Mailchimp case.



(a) Before redesign (b) After redesign

Figure 4-9: The example of a legitimate website's redesign in BoA case.

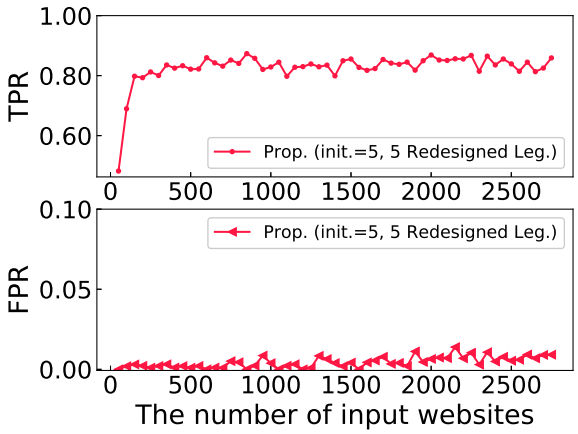


Figure 4-10: The case where initial signatures are created from redesigned BoA.

from Wayback Machine [150]). We evaluate EMD value and combination similarity which are leveraged in the proposed scheme. The EMD value and the combination similarity of these two figures are 0.82 and 0.03, respectively. Both of them indicate that their hue information is highly different from each other. In such cases, the system administrator has to add the signature of Figure 4-8(b) to SDB as another website, otherwise the phishing website mimicking the new design cannot be detected. However, if the change of redesign is small, the proposed scheme can detect phishing website with the new design. The redesign of BoA in 2019 is a good example of such a situation. Figure 4-9(a) and Figure 4-9(b) show the login pages of BoA before and after the redesign, respectively. As we can see from these figures, the used colors are almost the same between them except for the logos on upper left. In order to prove our argument, we conduct a simulation in the situation where five new design subpages of legitimate BoA are in the initial SDB [1]. Figure 4-10 shows true positive rate and false positive rate versus the number of input websites in the BoA dataset. As we can see from this figure, the scope of detection effectively expands. This indicates that redesigned phishing websites can be detected by signatures with old design if the change caused by redesign is small.

[1]Note that, only in this simulation, we use initial signatures created from the redesigned legitimate BoA. In other simulations, we use initial signatures created from BoA with the old design.

Phishing website with manipulated color

The proposed scheme is unable to detect a phishing website with drastically different hue information from a legitimate website, without modifying its contents such as layout, component, and messages. Figure 4-11 shows an example of such a website in Facebook case. In this figure, we painted the purple part of Facebook red without changing its contents. The EMD value and combination similarity between legitimate Facebook and this are 0.795 and 0.19, respectively. This result implies that the proposed scheme cannot detect the phishing website like this. As mentioned in a review paper [77], almost all of visual similarity-based phishing detection schemes are based on the fact that phishing websites look very similar in appearance to their corresponding legitimate websites to attract large number of Internet users. From this point of view, while it is difficult for most of visual similarity-based approaches to detect such phishing websites, the possibility that Facebook's users are deceived by red colored Facebook is low due to its strange looks. Moreover, in this case, other schemes which uses contents of HTML [100], [151] perform well because the contents are not changed. Since these schemes and the proposed scheme can be used complementarily, the hybrid approach is suitable for detecting such websites.

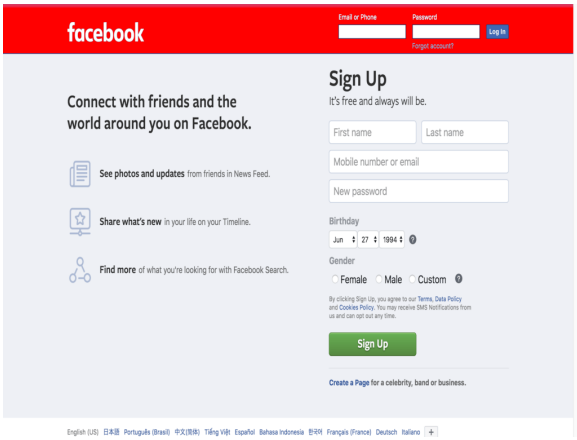


Figure 4-11: Red colored facebook.

4.2.7 Computational cost

Figure 4-12 shows the computational time per detection versus the number of input websites in Facebook dataset. As we can see from Figure 4-12, the computational time increases as the number of input websites increases. Since the proposed scheme stores detected phishing websites' signatures to SDB, it is natural that the computational cost becomes larger. When the number of input websites is small, the computational time of the proposed scheme is smaller than that of the previous scheme. This is because the previous scheme uses many colors for calculating similarity and uses logo detection scheme. From the point of computational cost, some readers may consider auto updating can be stopped when true positive rate is saturated. However, we cannot assert to stop auto updating due to zero-day attack. For all signature-based phishing detection scheme, the computational cost becomes larger as the number of signatures stored in SDB increases regardless of applying automatic updating. We consider detecting phishing websites with low computational cost is a challenge for all signature-based phishing detection scheme.

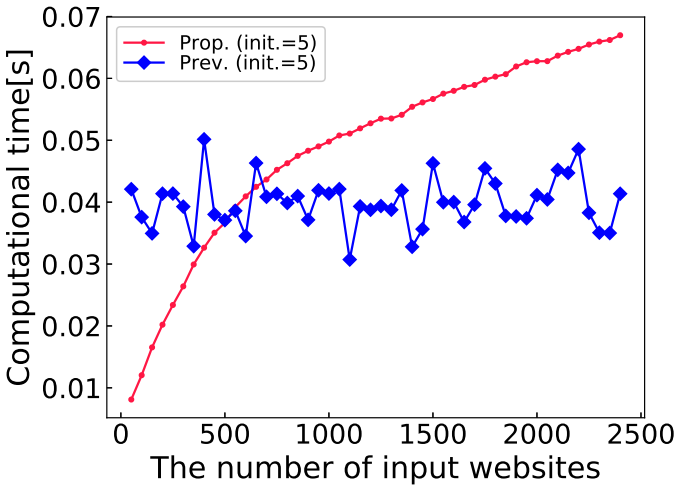


Figure 4-12: The computational time in Facebook dataset.

4.2.8 Discussion

Although the proposed scheme achieves 80%-90% of true positive rate, we should discuss that it is enough or not. This is because it is obvious that we should aim for 100% of true positive rate. Although there are some schemes achieving more than 99% of true positive rate in the field of phishing detection, it is not necessarily important due to the issues of the dataset dependency and zero-day attack. From these points of view, the combination of various detection schemes is practically needed. As shown in section 4.2.4, 10%-20% of the phishing websites which are not detected in the proposed scheme are far from the targeted legitimate websites. Thus, they are not the target of the proposed scheme and should be detected by other schemes. On the other hand, in the phishing websites detected in the proposed scheme include the ones which cannot be detected by the scheme like text feature-based approaches. Figure 4-13 shows an example of HTML source code of a phishing website which cannot be detected by text feature-based approaches. Although this is Paypal's phishing website, there are any strings of "paypal". The screenshot of Paypal's login page is embedded in the highlighted part. Hence, text feature-based approaches cannot deal with it though humans can recognize it as Paypal's website. Although other signature-based schemes can detect it if a signature for it is prepared, the proposed scheme can considerably reduce the cost of system administrator and thus is useful. We argue that true positive rate achieved in the proposed scheme is enough in terms of aiming for 100% by combining various schemes.

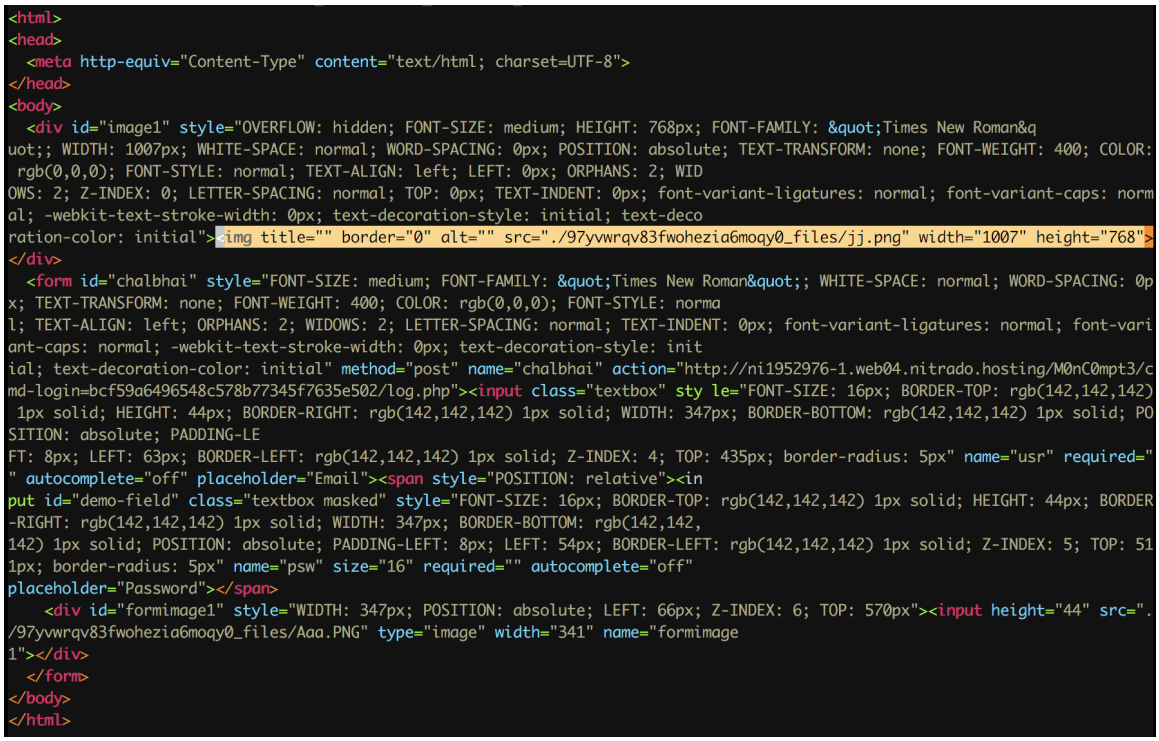


Figure 4-13: An example of HTML source code of a phishing website which cannot be detected by text feature-based approaches.

4.3 Conclusion and future works

We have proposed a novel visual similarity-based phishing detection scheme using hue information with auto updating database. Our proposal is the hue signature which is suitable for auto updating of signature database in phishing detection. By applying auto updating to our hue signature, the proposed scheme can have tolerance to zero-day phishing attack while reducing human cost. By the computer simulation with real dataset, we show the proposed scheme achieves high detection performance compared with the previous scheme.

As the future works, we consider as follows:

Although the proposed scheme in this paper utilizes hue signature with auto updating, the problems may be solved by incremental learning techniques. Furthermore, GAN (Generative Adversarial Network) which is a kind of deep learning technique, is promising for visual feature-based approaches. This is because it learns images of specific category and can generate other images of that category. In the context of phishing websites, it is easier to take countermeasures since researchers can generate phishing websites targeting specific legitimate website. The fields of detecting phishing websites will march toward using GAN or some deep learning techniques.

Chapter 5

Conclusions

This dissertation has discussed a study of fraud detection for Sybil accounts on social networking services and phishing websites. More and more people all over the world use Internet. The web services such as social networks, e-commerce, e-banking and so on are main platforms of today' s Internet utilization. They yield not only convenient lives for us but also threats that cannot be solved by the traditional defence approaches. Therefore, the countermeasures against these attacks are needed. One of the countermeasures is fraud detection. In each field of fraud detection, the target of detection and the type of data are different. Thus, it is necessary to use the fraud detection schemes dedicated to the case. In this dissertation, we have solved two issues regarding detection of Sybil accounts on SNS and that of phishing websites. The motivations for these are as follows;

Sybil accounts on SNS can become the entrance of many kinds of attacks including phishing, spreading spams, and so on since SNS can be primary touch point with many legitimate users and attackers. We consider the research to detect Sybil accounts lead the large part of Internet environment to be secure and they can be a great impact on the field of security.

Phishing websites incur the number of victims and the financial loss since everyone can be an attacker if he/she has the knowledge about HTML. Moreover, the phishing attack can be applied to all web services dealing with users' sensitive information. We consider the research to detect phishing websites is very important

and can contribute to both services and innocent legitimate users.

In Chapter 3, we have proposed a Sybil nodes detection scheme with robust seed selection and graph pruning on SNS. Our scheme is composed of two proposals. The first one is a seed selecting scheme by detecting communities and choosing seeds from them. The second one is a graph pruning scheme that considers trusted area. By the computer simulation, we show that our scheme achieves almost the same Sybil detection accuracy in the conventional attack scenario and outperforms the conventional scheme even if the attackers make a large number of common friends.

In Chapter 4, we have proposed a novel visual similarity-based phishing detection scheme using hue information with auto updating database. Our proposal is the hue signature which is suitable for auto updating of signature database in phishing detection. By applying auto updating to our hue signature, our system can have tolerance to zero-day phishing attack while reducing human cost. By the computer simulation with real dataset, we show our system achieves high detection performance compared with the previous scheme.

Bibliography

[1] M. Uma and G. Padmavathi, "A survey on various cyber attacks and their classification," *IJ Network Security*, vol. 15, no. 5, pp. 390–396, 2013.

[2] A. Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system: A survey," *Journal of Network and Computer Applications*, vol. 68, pp. 90–113, 2016.

[3] R. J. Bolton, D. J. Hand, *et al.*, "Unsupervised profiling methods for fraud detection," *Credit scoring and credit control*, vol. 7, pp. 235–255, 2001.

[4] L. Delamaire, H. Abdou, J. Pointon, *et al.*, "Credit card fraud and detection techniques: A review," *Banks and Bank systems*, vol. 4, no. 2, pp. 57–68, 2009.

[5] K. Sherly and R. Nedunchezhian, "Boat adaptive credit card fraud detection system," in *IEEE International Conference on Computational Intelligence and Computing Research*, 2010, pp. 1–7.

[6] W.-S. Yang and S.-Y. Hwang, "A process-mining framework for the detection of healthcare fraud and abuse," *Expert Systems with Applications*, vol. 31, no. 1, pp. 56–68, 2006.

[7] R. Kelley, "Where can $ 700 billion in waste be cut annually from the us healthcare system," *Ann Arbor, MI: Thomson Reuters*, vol. 24, 2009.

[8] M. K. Sparrow, *License to steal: how fraud bleeds America's health care system*. Routledge, 2019.

[9] N. Wu, Y. Qian, and G. Chen, "A novel approach to trojan horse detection by process tracing," in *IEEE International Conference on Networking, Sensing and Control*, 2006, pp. 721–726.

[10] H. Li, Q. Liu, J. Zhang, and Y. Lyu, "A survey of hardware trojan detection, diagnosis and prevention," in *IEEE International Conference on Computer-Aided Design and Computer Graphics (CAD/Graphics)*, 2015, pp. 173–180.

[11] N. Andronio, S. Zanero, and F. Maggi, "Heldroid: Dissecting and detecting mobile ransomware," in *Springer International Symposium on Recent Advances in Intrusion Detection*, 2015, pp. 382–404.

[12] A. Kharaz, S. Arshad, C. Mulliner, W. Robertson, and E. Kirda, "A large-scale, automated approach to detecting ransomware," in *USENIX Security Symposium*, 2016, pp. 757–772.

[13] H. Zhang, X. Xiao, F. Mercaldo, S. Ni, F. Martinelli, and A. K. Sangaiah, "Classification of ransomware families with machine learning based on n-gram of opcodes," *Future Generation Computer Systems*, vol. 90, pp. 211–221, 2019.

[14] A. Cohen, N. Nissim, L. Rokach, and Y. Elovici, "Sfem: Structural feature extraction methodology for the detection of malicious office documents using machine learning methods," *Expert Systems with Applications*, vol. 63, pp. 324–343, 2016.

[15] N. Nissim, A. Cohen, and Y. Elovici, "Aldocx: Detection of unknown malicious microsoft office documents using designated active learning methods based on new structural feature extraction methodology," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 3, pp. 631–646, 2016.

[16] P. Laskov and N. Šrndić, "Static detection of malicious javascript-bearing pdf documents," in *ACM 27th annual computer security applications conference*, 2011, pp. 373–382.

[17] D. Maiorca, D. Ariu, I. Corona, and G. Giacinto, "A structural and content-based approach for a precise and robust detection of malicious pdf files," in *IEEE International Conference on Information Systems Security and Privacy (ICISSP)*, 2015, pp. 27–36.

[18] N. Nissim, A. Cohen, R. Moskovitch, A. Shabtai, M. Edri, O. BarAd, and Y. Elovici, "Keeping pace with the creation of new malicious pdf files using an active-learning based detection framework," *Security Informatics*, vol. 5, no. 1, p. 1, 2016.

[19] M. Kamizono, M. Nishida, E. Kojima, and Y. Hoshizawa, "Categorizing hostile javascript using abstract syntax tree analysis," *IPSJ Journal*, vol. 54, no. 1, pp. 349–356, 2013.

[20] M. Nishida, Y. Hoshizawa, T. Kasama, M. Etou, D. Inoue, and K. Nakao, "Obfuscated malicious javascript detection using machine learning with character frequency," *Information processing society of Japan SIG Technical report*, 2014.

[21] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks," *IEEE communications surveys & tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013.

[22] S. García, A. Zunino, and M. Campo, "Survey on network-based botnet detection methods," *Security and Communication Networks*, vol. 7, no. 5, pp. 878–903, 2014.

[23] M. Bailey, E. Cooke, F. Jahanian, Y. Xu, and M. Karir, "A survey of botnet technology and defenses," in *IEEE Cybersecurity Applications & Technology Conference for Homeland Security*, 2009, pp. 299–304.

[24] S. S. Silva, R. M. Silva, R. C. Pinto, and R. M. Salles, "Botnets: A survey," *Computer Networks*, vol. 57, no. 2, pp. 378–403, 2013.

[25] R. A. Rodríguez-Gómez, G. Maciá-Fernández, and P. García-Teodoro, "Survey and taxonomy of botnet research through life-cycle," *ACM Computing Surveys (CSUR)*, vol. 45, no. 4, p. 45, 2013.

[26] R. Mishra, A. Singh, and R. Kumar, "Vanet security: Issues, challenges and solutions," in *IEEE International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, 2016, pp. 1050–1055.

[27] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, "Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 607–640, 2011.

[28] X. He, K. Wang, H. Huang, and B. Liu, "Qoe-driven big data architecture for smart city," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 88–93, 2018.

[29] K. Zaidi, M. B. Milojevic, V. Rakocevic, A. Nallanathan, and M. Rajarajan, "Host-based intrusion detection for vanets: A statistical approach to rogue node detection," *IEEE transactions on vehicular technology*, vol. 65, no. 8, pp. 6703–6714, 2015.

[30] IDC, *Smartphone os market share, 2018 q3*, `https://www.idc.com/promo/smartphone-market-share/os`.

[31] L. Deshotels, V. Notani, and A. Lakhotia, "Droidlegacy: Automated familial classification of android malware," in *ACM SIGPLAN on program protection and reverse engineering workshop*, 2014, p. 3.

[32] K. Xu, Y. Li, and R. H. Deng, "Iccdetector: Icc-based malware detection on android," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1252–1264, 2016.

[33] M. Sun, X. Li, J. C. Lui, R. T. Ma, and Z. Liang, "Monet: A user-oriented behavior-based malware variants detection system for android," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1103–1112, 2017.

[34] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-an, and H. Ye, "Significant permission identification for machine-learning-based android malware detection," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3216–3225, 2018.

[35] S. Wang, Q. Yan, Z. Chen, B. Yang, C. Zhao, and M. Conti, "Detecting android malware leveraging text semantics of network flows," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1096–1109, 2018.

[36] S. Garg, S. K. Peddoju, and A. K. Sarje, "Network-based detection of android malicious apps," *International Journal of Information Security*, vol. 16, no. 4, pp. 385–400, 2017.

[37] J. Heidemann, M. Klier, and F. Probst, "Online social networks: A survey of a global phenomenon," *Computer networks*, vol. 56, no. 18, pp. 3866–3878, 2012.

[38] J. R. Douceur, "The sybil attack," in *International workshop on peer-to-peer systems*, Springer, 2002, pp. 251–260.

[39] M. Al-Qurishi, M. Al-Rakhami, A. Alamri, M. Alrubaian, S. M. M. Rahman, and M. S. Hossain, "Sybil defense techniques in online social networks: A survey," *IEEE Access*, vol. 5, pp. 1200–1219, 2017.

[40] R. John, J. P. Cherian, and J. J. Kizhakkethottam, "A survey of techniques to prevent sybil attacks," in *IEEE International Conference on Soft-Computing and Networks Security (ICSNS)*, 2015, pp. 1–6.

[41] R. Gunturu, "Survey of sybil attacks in social networks," *arXiv:1504.05522*, 2015.

[42] S. D. Vyas, "Impact of e-banking on traditional banking services," *arXiv preprint arXiv:1209.2368*, 2012.

[43] M. Subramani and E. Walden, "The impact of e-commerce announcements on the market value of firms," *Information Systems Research*, vol. 12, no. 2, pp. 135–154, 2001.

[44] A. Almomani, B. Gupta, S. Atawneh, A. Meulenberg, and E. Almomani, "A survey of phishing email filtering techniques," *IEEE communications surveys & tutorials*, vol. 15, no. 4, pp. 2070–2090, 2013.

[45] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: A literature survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2091–2121, 2013.

[46] A. Tewari, A. Jain, and B. Gupta, "Recent survey of various defense mechanisms against phishing attacks," *Journal of Information Privacy and Security*, vol. 12, no. 1, pp. 3–13, 2016.

[47] J. Clement, *Number of social network users worldwide*, `https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/`.

[48] FBI, *2018 INTERNET CRIME REPORT*, `https://pdf.ic3.gov/2018_IC3Report.pdf`.

[49] Facebook, *Reporting abusive accounts*, `https://www.facebook.com/help/263149623790594/`.

[50] K. S. Adewole, N. B. Anuar, A. Kamsin, K. D. Varathan, and S. A. Razak, "Malicious accounts: Dark of the social networks," *Journal of Network and Computer Applications*, vol. 79, pp. 41–67, 2017.

[51] A. Mislove, A. Post, P. Druschel, and P. K. Gummadi, "Ostra: Leveraging trust to thwart unwanted communication.," in *NSDI*, vol. 8, 2008, pp. 15–30.

[52] D. N. Tran, B. Min, J. Li, and L. Subramanian, "Sybil-resilient online content voting.," in *NSDI*, vol. 9, 2009, pp. 15–28.

[53] A. Post, V. Shah, and A. Mislove, "Bazaar: Strengthening user reputations in online marketplaces," in *Proceedings of NSDI*, vol. 11, 2011, p. 183.

[54] N. Chiluka, N. Andrade, J. Pouwelse, and H. Sips, "Leveraging trust and distrust for sybil-tolerant voting in online social media," in *ACM 1st Workshop on Privacy and Security in Online Social Media*, 2012, p. 1.

[55] B. Viswanath, M. Mondal, K. P. Gummadi, A. Mislove, and A. Post, "Canal: Scaling social network-based sybil tolerance schemes," in *ACM 7th european conference on Computer Systems*, 2012, pp. 309–322.

[56] J. Zhang, R. Zhang, J. Sun, Y. Zhang, and C. Zhang, "Truetop: A sybil-resilient system for user influence measurement on twitter," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2834–2846, 2015.

[57] M. J. Freedman and R. Morris, "Tarzan: A peer-to-peer anonymizing network layer," in *ACM 9th conference on Computer and communications security*, 2002, pp. 193–206.

[58] F. Cornelli, E. Damiani, S. D. C. Di Vimercati, S. Paraboschi, and P. Samarati, "Choosing reputable servents in a p2p network," in *ACM 11th international conference on World Wide Web*, 2002, pp. 376–386.

[59] N. Borisov, "Computational puzzles as sybil defenses," in *IEEE International Conference on Peer-to-Peer Computing (P2P'06)*, 2006, pp. 171–176.

[60] F. Li, P. Mittal, M. Caesar, and N. Borisov, "Sybilcontrol: Practical sybil defense with computational puzzles," in *ACM workshop on Scalable trusted computing*, 2012, pp. 67–78.

[61] L. Von Ahn, M. Blum, N. J. Hopper, and J. Langford, "Captcha: Using hard ai problems for security," in *Springer International Conference on the Theory and Applications of Cryptographic Techniques*, 2003, pp. 294–311.

[62] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "Sybilguard: Defending against sybil attacks via social networks," in *ACM SIGCOMM Computer Communication Review*, vol. 36, 2006, pp. 267–278.

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

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

[65] A. Mohaisen, N. Hopper, and Y. Kim, "Keep your friends close: Incorporating trust into social network-based sybil defenses," in *IEEE INFOCOM*, 2011, pp. 1943–1951.

[66] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro, "Aiding the detection of fake accounts in large scale social online services," in *USENIX NSDI*, 2012, pp. 1–14.

[67] H. Zhang, C. Xu, and J. Zhang, "Exploiting trust and distrust information to combat sybil attack in online social networks," in *Trust Management VIII*, 2014, pp. 77–92.

[68] N. Tran, J. Li, L. Subramanian, and S. S. Chow, "Optimal sybil-resilient node admission control," in *IEEE INFOCOM*, 2011, pp. 3218–3226.

[69] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai, "Uncovering social network sybils in the wild," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 8, no. 1, p. 2, 2014.

[70] G. Wang, M. Mohanlal, C. Wilson, X. Wang, M. Metzger, H. Zheng, and B. Y. Zhao, "Social turing tests: Crowdsourcing sybil detection," *arXiv preprint arXiv:1205.3856*, 2012.

[71] G. Wang, T. Konolige, C. Wilson, X. Wang, H. Zheng, and B. Y. Zhao, "You are how you click: Clickstream analysis for sybil detection," in *USENIX 22nd Security Symposium (USENIX Security 13)*, 2013, pp. 241–256.

[72] N. Z. Gong, M. Frank, and P. Mittal, "Sybilbelief: A semi-supervised learning approach for structure-based sybil detection," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 6, pp. 976–987, 2014.

[73] P. Gao, N. Z. Gong, S. Kulkarni, K. Thomas, and P. Mittal, "Sybilframe: A defense-in-depth framework for structure-based sybil detection," *arXiv preprint arXiv:1503.02985*, 2015.

[74] A. N. Langville and C. D. Meyer, "Deeper inside pagerank," *Internet Mathematics*, vol. 1, no. 3, pp. 335–380, 2004.

[75] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[76] A. Liaw, M. Wiener, *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[77] A. K. Jain and B. B. Gupta, "Phishing detection: Analysis of visual similarity based approaches," *Security and Communication Networks*, 2017.

[78] *Anti phishing working group 2018*, https://www.antiphishing.org.

[79] R. C. Dodge Jr, C. Carver, and A. J. Ferguson, "Phishing for user security awareness," *computers & security*, vol. 26, no. 1, pp. 73–80, 2007.

[80] P. Kumaraguru, Y. Rhee, A. Acquisti, L. F. Cranor, J. Hong, and E. Nunge, "Protecting people from phishing: The design and evaluation of an embedded training email system," in *ACM SIGCHI conference on Human factors in computing systems*, 2007, pp. 905–914.

[81] P. Kumaraguru, Y. Rhee, S. Sheng, S. Hasan, A. Acquisti, L. F. Cranor, and J. Hong, "Getting users to pay attention to anti-phishing education: Evaluation of retention and transfer," in *anti-phishing working groups 2nd annual eCrime researchers summit*, 2007, pp. 70–81.

[82] I. Kirlappos and M. A. Sasse, "Security education against phishing: A modest proposal for a major rethink," *IEEE Security & Privacy*, vol. 10, no. 2, pp. 24–32, 2011.

[83] P. Kumaraguru, S. Sheng, A. Acquisti, L. F. Cranor, and J. Hong, "Lessons from a real world evaluation of anti-phishing training," in *IEEE eCrime Researchers Summit*, 2008, pp. 1–12.

[84] S. Sheng, B. Magnien, P. Kumaraguru, A. Acquisti, L. F. Cranor, J. Hong, and E. Nunge, "Anti-phishing phil: The design and evaluation of a game that teaches people not to fall for phish," in *ACM 3rd symposium on Usable privacy and security*, 2007, pp. 88–99.

[85] M. Dixon, N. A. Gamagedara Arachchilage, and J. Nicholson, "Engaging users with educational games: The case of phishing," in *ACM Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, 2019, LBW0265.

[86] M. L. Hale, R. F. Gamble, and P. Gamble, "Cyberphishing: A game-based platform for phishing awareness testing," in *IEEE 48th International Conference on System Sciences*, 2015, pp. 5260–5269.

[87] J.-C. Liou, G. Egan, J. K. Patel, and S. Bhashyam, "A sophisticated rfid application on multi-factor authentication," in *IEEE International Conference on Information Technology: New Generations*, 2011, pp. 180–185.

[88] B. Parno, C. Kuo, and A. Perrig, "Phoolproof phishing prevention," in *Springer International Conference on Financial Cryptography and Data Security*, 2006, pp. 1–19.

[89] R. S. Pippal, C. Jaidhar, and S. Tapaswi, "Robust smart card authentication scheme for multi-server architecture," *Wireless Personal Communications*, vol. 72, no. 1, pp. 729–745, 2013.

[90] R. Dhamija and J. D. Tygar, "The battle against phishing: Dynamic security skins," in *ACM symposium on Usable privacy and security*, 2005, pp. 77–88.

[91] N. Fraser, "The usability of picture passwords," *Tricerion Group plc*, 2006.

[92] B. Ross, C. Jackson, N. Miyake, D. Boneh, and J. C. Mitchell, "Stronger password authentication using browser extensions.," in *USENIX Security Symposium*, Baltimore, MD, USA, 2005, pp. 17–32.

[93] S. Sheng, B. Wardman, G. Warner, L. F. Cranor, J. Hong, and C. Zhang, "An empirical analysis of phishing blacklists," in *Sixth Conference on Email and Anti-Spam (CEAS)*, California, USA, 2009.

[94] *Google safe browsing*, https://developers.google.com/safe-browsing/.

[95] Google, *Protocolv2Spec*, `https://code.google.com/archive/p/google-safe-browsing/wikis/Protocolv2Spec.wiki`.

[96] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "Phishnet: Predictive blacklisting to detect phishing attacks," in *IEEE INFOCOM*, 2010, pp. 1–5.

[97] J. Kang and D. Lee, "Advanced white list approach for preventing access to phishing sites," in *IEEE International Conference on Convergence Information Technology (ICCIT)*, 2007, pp. 491–496.

[98] Y. Cao, W. Han, and Y. Le, "Anti-phishing based on automated individual white-list," in *ACM 4th workshop on Digital identity management*, 2008, pp. 51–60.

[99] Y. Wang, R. Agrawal, and B.-Y. Choi, "Light weight anti-phishing with user whitelisting in a web browser," in *IEEE Region 5 Conference*, 2008, pp. 1–4.

[100] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: A content-based approach to detecting phishing web sites," in *ACM 16th international conference on World Wide Web*, 2007, pp. 639–648.

[101] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, "Cantina+: A feature-rich machine learning framework for detecting phishing web sites," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 2, p. 21, 2011.

[102] G. Varshney, M. Misra, and P. K. Atrey, "A phish detector using lightweight search features," *Computers & Security*, vol. 62, pp. 213–228, 2016.

[103] G. Ramesh, I. Krishnamurthi, and K. S. S. Kumar, "An efficacious method for detecting phishing webpages through target domain identification," *Decision Support Systems*, vol. 61, pp. 12–22, 2014.

[104] J. H. Huh and H. Kim, "Phishing detection with popular search engines: Simple and effective," in *Springer International Symposium on Foundations and Practice of Security*, 2011, pp. 194–207.

[105] E. H. Chang, K. L. Chiew, W. K. Tiong, *et al.*, "Phishing detection via identification of website identity," in *IEEE International Conference on IT Convergence and Security (ICITCS)*, 2013, pp. 1–4.

[106] M. Dunlop, S. Groat, and D. Shelly, "Goldphish: Using images for content-based phishing analysis," in *IEEE Fifth International Conference on Internet Monitoring and Protection*, 2010, pp. 123–128.

[107] P. Singh, Y. P. Maravi, and S. Sharma, "Phishing websites detection through supervised learning networks," in *IEEE International Conference on Computing and Communications Technologies (ICCCT)*, 2015, pp. 61–65.

[108] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Predicting phishing websites based on self-structuring neural network," *Neural Computing and Applications*, vol. 25, no. 2, pp. 443–458, 2014.

[109] R. Mohammad, T. McCluskey, and F. A. Thabtah, "Predicting phishing websites using neural network trained with back-propagation," World Congress in Computer Science, Computer Engineering, and Applied Computing, 2013.

[110] L. A. T. Nguyen, B. L. To, H. K. Nguyen, and M. H. Nguyen, "A novel approach for phishing detection using url-based heuristic," in *IEEE International Conference on Computing, Management and Telecommunications (ComManTel)*, 2014, pp. 298–303.

[111] N. Abdelhamid, A. Ayesh, and F. Thabtah, "Phishing detection based associative classification data mining," *Expert Systems with Applications*, vol. 41, no. 13, pp. 5948–5959, 2014.

[112] N. Abdelhamid, "Multi-label rules for phishing classification," *Applied Computing and Informatics*, vol. 11, no. 1, pp. 29–46, 2015.

[113] F. Kausar, B. Al-Otaibi, A. Al-Qadi, and N. Al-Dossari, "Hybrid client side phishing websites detection approach," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 5, no. 7, pp. 132–140, 2014.

[114] P. A. Barraclough, M. A. Hossain, M. Tahir, G. Sexton, and N. Aslam, "Intelligent phishing detection and protection scheme for online transactions," *Expert Systems with Applications*, vol. 40, no. 11, pp. 4697–4706, 2013.

[115] B. Eshete, "Effective analysis, characterization, and detection of malicious web pages," in *ACM 22nd International Conference on World Wide Web*, 2013, pp. 355–360.

[116] L. Xu, Z. Zhan, S. Xu, and K. Ye, "Cross-layer detection of malicious websites," in *ACM third conference on Data and application security and privacy*, 2013, pp. 141–152.

[117] J. Mao, W. Tian, P. Li, T. Wei, and Z. Liang, "Phishing-alarm: Robust and efficient phishing detection via page component similarity," *IEEE Access*, vol. 5, pp. 17 020–17 030, 2017.

[118] F. C. Dalgic, A. S. Bozkir, and M. Aydos, "Phish-iris: A new approach for vision based brand prediction of phishing web pages via compact visual descriptors," in *2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2018, pp. 1–8.

[119] R. S. Rao and S. T. Ali, "A computer vision technique to detect phishing attacks," in *Fifth International Conference on Communication Systems and Network Technologies*, Apr. 2015, pp. 596–601.

[120] K. Chen, J. Chen, C. Huang, and C. Chen, "Fighting phishing with discriminative keypoint features," *IEEE Internet Computing*, vol. 13, no. 3, pp. 56–63, May 2009, ISSN: 1089-7801.

[121] A. S. Bozkir and E. A. Sezer, "Use of hog descriptors in phishing detection," in *4th International Symposium on Digital Forensic and Security (ISDFS)*, Apr. 2016, pp. 148–153.

[122] A. Y. Fu, L. Wenyin, and X. Deng, "Detecting phishing web pages with visual similarity assessment based on earth mover's distance (emd)," *IEEE transactions on dependable and secure computing*, vol. 3, no. 4, pp. 301–311, 2006.

[123] Y. Zhou, Y. Zhang, J. Xiao, Y. Wang, and W. Lin, "Visual similarity based anti-phishing with the combination of local and global features," in *IEEE 13th*

International Conference on Trust, Security and Privacy in Computing and Communications, 2014, pp. 189–196.

[124] G. Varshney, M. Misra, and P. K. Atrey, "A survey and classification of web phishing detection schemes," *Security and Communication Networks*, vol. 9, no. 18, pp. 6266–6284, 2016.

[125] K. L. Chiew, C. L. Tan, K. Wong, K. S. Yong, and W. K. Tiong, "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system," *Information Sciences*, vol. 484, pp. 153–166, 2019.

[126] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[127] Amazon, *Alexa*, `https://www.alexa.com/siteinfo`.

[128] F. L. Hitchcock, "The distribution of a product from several sources to numerous localities," *Journal of mathematics and physics*, vol. 20, no. 1-4, pp. 224–230, 1941.

[129] R. Flamary and N. Courty, *Pot python optimal transport library*, `https://github.com/rflamary/POT`, 2017.

[130] *Fake Accounts in Facebook - How to Counter It*. [Online]. Available: `http://ezinearticles.com/?Fake-Accounts-in-Facebook---How-to-Counter-It&id=3703889`.

[131] D. Koll, J. Li, J. Stein, and X. Fu, "On the state of osn-based sybil defenses," in *IEEE IFIP Networking*, 2014, pp. 1–9.

[132] B. Wang, J. Jia, L. Zhang, and N. Z. Gong, "Structure-based sybil detection in social networks via local rule-based propagation," *IEEE Transactions on Network Science and Engineering*, 2018.

[133] L. Alvisi, A. Clement, A. Epasto, S. Lattanzi, and A. Panconesi, "Sok: The evolution of sybil defense via social networks," in *IEEE Symposium on Security and Privacy*, 2013, pp. 382–396.

[134] J. Leskovec and J. J. Mcauley, "Learning to discover social circles in ego networks," in *Advances in neural information processing systems*, 2012, pp. 539–547.

[135] M. E. Newman, "Fast algorithm for detecting community structure in networks," *Physical review E*, vol. 69, no. 6, p. 066 133, 2004.

[136] H. Rashtian, Y. Boshmaf, P. Jaferian, and K. Beznosov, "To befriend or not? a model of friend request acceptance on facebook," in *Symposium on Usable Privacy and Security (SOUPS)*, 2014.

[137] I. Nikolaev, M. Grill, and V. Valeros, "Exploit kit website detection using http proxy logs," in *ACM Fifth International Conference on Network, Communication and Computing*, 2016, pp. 120–125.

[138] F. B. de Sousa and L. Zhao, "Evaluating and comparing the igraph community detection algorithms," in *IEEE Brazilian Conference on Intelligent Systems*, 2014, pp. 408–413.

[139] S. Pérez-Peló, J. Sánchez-Oro, R. Martín-Santamaría, and A. Duarte, "On the analysis of the influence of the evaluation metric in community detection over social networks," *Electronics*, vol. 8, no. 1, p. 23, 2019.

[140] *Shinzo Abe Facebook*, https://www.facebook.com/abeshinzo?fref=ts.

[141] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, 1982.

[142] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic ROC curve.," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[143] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999.

[144] G. Csardi and T. Nepusz, "The igraph software package for complex network research," *InterJournal*, vol. Complex Systems, p. 1695, 2006. [Online]. Available: `http://igraph.org`.

[145] C. Liu, P. Gao, M. Wright, and P. Mittal, "Exploiting temporal dynamics in sybil defenses," in *ACM 22nd SIGSAC Conference on Computer and Communications Security*, 2015, pp. 805–816.

[146] N. C. Lê, M.-T. Dao, H.-L. Nguyen, T.-N. Nguyen, and H. Vu, "An application of random walk on fake account detection problem: A hybrid approach," *arXiv preprint arXiv:1911.07609*, 2019.

[147] S.-S. Choi, S.-H. Cha, and C. C. Tappert, "A survey of binary similarity and distance measures," *Journal of Systemics, Cybernetics and Informatics*, vol. 8, no. 1, pp. 43–48, 2010.

[148] *Phishtank*, http://www.phishtank.com.

[149] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[150] *Wayback machine*, `https://archive.org/web/`.

[151] A. P. Rosiello, E. Kirda, F. Ferrandi, *et al.*, "A layout-similarity-based approach for detecting phishing pages," in *IEEE International Conference on Security and Privacy in Communications Networks and the Workshops-SecureComm*, 2007, pp. 454–463.

Appendix A

Publication List

A.1 Journals

[1] S. Haruta, K. Toyoda, and I. Sasase, "Trust-based Sybil Nodes Detection with Robust Seed Selection and Graph Pruning on SNS, " *IEICE Transactions Special Section on Internet Architectures and Management Methods that Enable Flexible and Secure Deployment of Network Services*, vol.E99-B, no.5, pp.1002–1011, 2016.

[2] R. Negishi, S. Haruta, C. Inamura, K. Toyoda, and I. Sasase, "Monetary Fair Battery-based Load Hiding Scheme for Multiple Households in Automatic Meter Reading System, " *Journal of Telecommunications and Information Technology*, pp.110–119, 2016.

[3] M. Yoshida, H. Asahina, S. Haruta, and I. Sasase, "A false density information attack detection scheme using overlap of communication range in VANET, " *IEICE Communications Express* vol.8, no.5, pp.135-140, 2019.

[4] K. Osuge, H. Kato, S. Haruta, and I. Sasase, "An Effective Feature Selection Scheme for Android ICC-Based Malware Detection Using the Gap of the Appearance Ratio, " *IEICE Transactions on Information and Systems*, vol. E102-D, no.6, pp.1136-1144, 2019

[5] H. Asahina, K. Sakuma, S. Haruta, H. Kato, and I. Sasase, "Traceroute-based target link flooding attack detection scheme by analyzing hop count to the destination, " *IEICE Communications Express*, vol.8, no.7, pp.251-256, 2019.

[6] S. Haruta, H. Asahina, F. Yamazaki, and I. Sasase, "Hue Signature Auto Update System for Visual Similarity-based Phishing Detection with Tolerance to Zero-day Attack, " *IEICE Transactions on Information and Systems* vol.E102-D, no.12, pp.2461-2471, Dec. 2019.

[7] Y. An, S. Haruta, S. Choi, and I. Sasase, "Traffic feature-based botnet detection scheme emphasizing the importance of long patterns, " in *IEICE Communications Express*, vol.9, no.1, pp.7-12, 2019.

[8] H. Kato, S. Haruta, and I. Sasase, "Android Malware Detection Scheme Based on Level of SSL Server Certificate, " in *IEICE Transactions on Information and Systems*, vol.E103-D, no.2, pp.-, Feb. 2020 (Accepted).

A.2 Conferences Proceedings (peer-reviewed)

[1] S. Haruta, K. Toyoda, and I. Sasase, "Trust-based Sybil nodes Detection with Robust Seed Selection and Graph Pruning on SNS, " in *IEEE Workshop on Information Forensics and Security (WIFS)*, Rome, Italy, 16-19 Nov., 2015.

[2] H. Kato, S. Haruta, and I. Sasase, "Malicious PDF Detection Scheme Using the Useful Feature Based on Non-Frequent Keywords in a File, " in *IEICE Information and Communication Technology Forum (ICTF)* , Poznan, Poland, 4–6 July, 2017.

[3] S. Choi, S. Haruta, H. Asahina, and I. Sasase, "Cost Effective Dummy Generation Scheme in Non-Trusted LBS, " in *IEICE Information and Communication Technology Forum (ICTF)*, Poznan, Poland, July 4-6, 2017.

[4] K. Sakuma, S. Haruta, H. Asahina, and I. Sasase, "Traceroute-based Target Link Flooding Attack Detection Scheme by Analyzing Hop Count to the Desti-

nation, " in *IEEE Asia-Pacific Conference on Communications (APCC)*, Perth, Australia, 11-13, Dec. 2017.

[5] S. Morishige, S. Haruta, H. Asahina, and I. Sasase, "Obfuscated Malicious JavaScript Detection Scheme Using the Feature Based on Divided URL, " in *IEEE Asia-Pacific Conference on Communications (APCC)*, Perth, Australia, 11-13, Dec. 2017.

[6] S. Haruta, H. Asahina, and I. Sasase, "Visual Similarity-based Phishing Detection Scheme using Image and CSS with Target Website Finder, " in *IEEE Global Telecommunications Conference Communication (GLOBECOM)*, Singapore, 7 Dec. 2017.

[7] M. Yoshida, H. Asahina, S. Haruta, and I. Sasase, "A False Information Attack Detection Scheme using Density of Vehicles and Overlap of Communication Range in VANET, " in *IEICE Information and Communication technology Forum (ICTF)*, " Graz, Austria, July 11-13, 2018.

[8] K. Osuge, H. Kato, S. Haruta, and I. Sasase, "Feature Selection Scheme for Android ICC-related Features Based on the Gap of the Appearance Ratio, " in *IEICE Information and Communication technology Forum (ICTF)*, Graz, Austria, July 11-13, 2018.

[9] K. Arai, S. Haruta, Hiromu Asahina, and Iwao Sasase, "Encounter Record Reduction Scheme based on Theoretical Contact Probability for Flooding Attack Mitigation in DTN, " in *IEEE Asia-Pacific Conference on Communications (APCC)*, Nimbou, China, Nov.12-14, 2018.

[10] Y. An, S. Haruta, S. Choi, and I. Sasase, "Traffic Feature-based Botnet Detection Scheme Emphasizing the Importance of Long Patterns, " in *IEICE Information and Communication Technology Forum (ICTF)*, Bydgoszcz, Poland, September 11–13, 2019.

[11] H. Kato, S. Haruta, and I. Sasase, "Android Malware Detection Scheme Based on Level of SSL Server Certificate, " in *IEEE Global Telecommunications Conference Communication (GLOBECOM)*, Dec. 9 – 13, Waikoloa, Hawaii, USA 2019.

[12] H. Nakano, H. Kato, S. Haruta, M. Yoshida, and I. Sasase, "Trust-based Verification Attack Prevention Scheme using Tendency of Contents Request on NDN, " in *IEEE Asia-Pacific Conference on Communications (APCC)*, Ho Chi Minh city, Vietnam, Nov. 6-8, 2019.

[13] S. Haruta, F. Yamazaki, H. Asahina, and I. Sasase, "A Novel Visual Similarity-based Phishing Detection Scheme using Hue Information with Auto Updating Database, " in *IEEE Asia-Pacific Conference on Communications (APCC)*, Ho Chi Minh city, Vietnam, Nov. 6-8, 2019.

A.3 Conferences Proceedings (in Japanese, without peer-review)

[1] 春田秀一郎,豊田健太郎,笹瀬巌,"SNSの不正アカウント検出においてコミュニティ構造に着目したグラフ剪定及びシード選択法, " 情報処理学会第 69 回コンピュータセキュリティ第29回インターネットと運用技術合同研究発表会, 別府, 2015 年 5 月 21-22 日.

[2] S. Haruta, H. Asahina, and I. Sasase, "Visual Similarity-based Phishing websites Detection Scheme using Image and CSS with Target Website Finder, " 電子情報通信学会情報通信システムセキュリティ研究会，2017年3月14日.

[3] 加藤広野, 春田秀一郎, 笹瀬巌, "ファジィ推論をキーワード及びその種類数に適用して得られる特徴を用いた悪性 PDF 検知法, "電子情報通信学会情報通信システムセキュリティ研究会，2017 年 3 月 13 日.

[4] 崔相勳, 春田秀一郎, 朝比奈啓, 笹瀬巌, "Non-Trusted LBS を用いた低コスト

なダミー生成方式, ” 電子情報通信学会通信方式研究会, vol. 117, no. 156, CS2017-14, pp. 7-12, 2017 年 7 月 27 日.

[5] 森重翔也, 春田秀一郎, 朝比奈啓, 笹瀬巌, “URLを分割する難読化が施された悪性 JavaScript の検出法, ” 電子情報通信学会通信方式研究会, vol. 117, no. 156, CS2017-15, pp. 13-18, 2017 年 7 月 27 日.

[6] 佐久間慧, 朝比奈啓, 春田秀一郎, 笹瀬巌, “宛先までのホップ数解析によるTracerouteを用いたTarget Link Flooding Attack 検知手法, ” 電子情報通信学会通信方式研究会, CS2017-56, 2017 年 11 月 16 日.

[7] K. Arai, S. Haruta, H. Asahina, and I. Sasase, “Encounter Record Reduction Scheme based on Theoretical Contact Probability for Flooding Attack Mitigation in DTN, ” 電子情報通信学会通信方式研究会, 2018年7月.

[8] M. Yoshida, H. Asahina, S. Haruta, and I. Sasase, “False Density Information Attack Detection Scheme Using Overlap of Communication Range in VANET, ” 電子情報通信学会通信方式研究会, 2018 年 11月1日.

[9] K. Osuge, H. Kato, S. Haruta, and I. Sasase,“Feature Selection Scheme for Android ICC-related Features Based on the Gap of the Appearance Ratio, ” 電子情報通信学会通信方式研究会, 2018 年 11 月 1 日.

[10] 春田秀一郎, 山崎史貴, 朝比奈啓, 笹瀬巌, “色相を利用して自動的に検知範囲を拡大可能なフィッシングサイト検知法,” 電子情報通信学会通信方式研究会、CS2019-14, pp.7-12, 2019 年 7 月 4 日.

[11] 加藤広野, 春田秀一郎, 笹瀬巌, “SSLサーバ証明書の認証レベルに着目した悪性Android アプリ検知手法, ” 電子情報通信学会通信方式研究会、CS2019-15, pp.13-18, 2019 年 7 月 4 日.

[12] Y. An, S. Haruta, S. Choi, and I. Sasase, “Traffic Feature-based Botnet Detection Scheme Emphasizing the Importance of Long Patterns, ” 電子情報通信学会通信方式研究会、CS2019-18, pp.31-35, 2019 年 7 月4日.

[13] 中野紘典, 加藤広野, 春田秀一郎, 吉田匡志, 笹瀬巌, “Named Data Networking においてユーザのコンテンツ取得傾向に基づく信頼値により攻撃者の行動を制限する方式, ” 電子情報通信学会通信方式研究会、 CS2019-13, pp.1-6, 2019 年7月 4 日.