Social Network Analysis and Mining (2022) 12:111
https://doi.org/10.1007/s13278-022-00943-3

ORIGINAL ARTICLE

HyperMan: detecting misbehavior in online forums based on hyperlink posting behavior

Risul Islam[1] · **Ben Treves**[1] · **Md Omar Faruk Rokon**[1] · **Michalis Faloutsos**[1]

Received: 19 December 2021 / Revised: 9 April 2022 / Accepted: 15 July 2022 / Published online: 12 August 2022

Abstract

How can we detect and analyze hyperlink-driven misbehavior in online forums? Online forums contain enormous amounts of user-generated contents, with threads and comments frequently supplemented by hyperlinks. These hyperlinks are often posted with malicious intention and we refer to this as 'hyperlink-driven misbehavior.' We present HyperMan, a systematic suite of capabilities, to detect and analyze hyperlink-driven misbehavior in online forums. We take a unique perspective focusing on hyperlink sharing practices of the users to spot misbehavior. HyperMan can categorize these hyperlinks as (a) phishing, (b) spamming, and (b) promoting malicious products. Our approach consists of three high-level phases: (a) extracting hyperlinks from the textual data, (b) identifying misbehaving hyperlinks, and (c) modeling the behavioral patterns of hyperlink sharing, where we identify key hyperlinks and analyze the collaboration dynamics of hyperlink sharing. In addition, we implement our approach as a powerful and easy-to-use open platform for practitioners. We apply HyperMan to spot misbehavior from three online security forums, where we expect the users to be more security-aware. We show that our approach works very well in terms of retrieving and classifying hyperlinks compared to previous solutions. Furthermore, we find non-trivial and often systematic misbehavior: (a) we find a total of 2703 misbehaving hyperlinks, and (b) we identify 94 colluding groups of users in terms of promoting hyperlinks. Our work is a significant step toward mining online forums and detecting misbehaving users comprehensively.

Keywords Security forums · URL extraction · Phishing detection · Misbehavior detection

1 Introduction

How pervasive is spamming and phishing in online forums? This burning question is the motivation behind our work. With the widespread adoption of email, spamming and phishing have emerged as key nuisances and threats. Recently, email filters have improved and the use of spam-aware services like Gmail have contained the reach of these activities. However, hackers are tenacious and are always on the search for new ways to accomplish their goals as new technologies emerge. One such new opportunity is presented by online forums, which have seen a tremendous increase in both number and user engagement. Currently, there are 1M+ forums with an estimated 550M+ registered users (Sidonce 2021).

Here, we take a more niche angle and we focus on security forums, where one would assume that users are more aware and thus less likely to fall victim to spamming and phishing. Online security forums bring together a wide variety of users generating enormous amounts of security-related content (Islam et al. 2020b) through their comments which are often supplemented by hyperlinks. Clearly, there are benign usages of these hyperlinks that can point to useful information. However, we find that sharing hyperlinks often accompanies malicious intentions which we call 'hyperlink-driven misbehavior.' This misbehavior can be broadly grouped into (a) phishing, (b) spamming, and (c) sharing of malicious products. Fully quantifying this misbehavior can reveal malicious hackers using security forums for malicious

✉ Risul Islam
risla002@ucr.edu

Ben Treves
btrev003@ucr.edu

Md Omar Faruk Rokon
mroko001@ucr.edu

Michalis Faloutsos
michalis@cs.ucr.edu

[1] UC Riverside, Riverside, USA

purposes. Throughout this paper, we use the terms *hyperlink*, *link*, and *URL* interchangeably.

The problem we address here is the following: *How can we detect and analyze hyperlink-driven misbehavior in online forums?* The question is motivated by the studies showing that malicious hackers have a strong presence in public forums: (a) they spread malware often masquerading as technical solutions or antivirus and create an online brand (Knot 2021), and (b) they promote malicious services (Gharibshah et al. 2020). We narrow down our focus to (a) security forums, and (b) misbehavior that is enabled via hyperlinks in posts. Therefore, the input to our problem is posted from online forums, which include the author and date. The outputs are (a) the misbehaving hyperlinks, (b) the type of misbehavior, and (c) the persistent offenders and their collaboration patterns.

There has been limited work focusing on the question as we frame it here. We are not aware of any work which identifies misbehavior from the perspective of hyperlink sharing in online forums. In fact, mining security forums in general has received relatively recent little attention. We can identify two main categories of related efforts: (a) security forum studies and (b) hyperlink classification studies. We discuss these efforts in our related work section.

As our key contribution, we propose HyperMan, a comprehensive methodology for detecting and analyzing hyperlink-driven misbehavior in online security forums. Our approach consists of the following key functions: (a) URL extraction, (b) systematic classification of URLs, and (c) modeling of emerging URL posting behaviors as shown in Fig. 1.

From an algorithmic point of view, HyperMan makes the following contributions. We develop *RPhish*, a novel machine learning-based approach to detect phishing websites. In this approach, we consider an exhaustive set of features along three dimensions: (a) name-related features, (b) network/reputation level information, and (c) web-page content. For the classification, we combine principal component analysis (PCA) for feature compression with a five-layer neural network. In addition, we develop *ExtLink*, a systematic approach to extract hyperlinks from raw text. In modeling user and group behaviors, we use a combination of tensor decomposition and DBSCAN for detecting outlier behaviors.

Our key results can be summarized in the following points.

(a) Our proposed methods perform well Our phishing detection algorithm, *RPhish*, demonstrates 98.2% accuracy, 97.01% True Positive Rate, and 1.3% False Positive Rate, beating the current best phishing detection models. In addition, our URL extraction method, *ExtLink*, exhibits a 26.7% increase in precision without any reduction in recall compared to baseline regular expression-based extraction.

(b) There is a non-trivial amount of malicious URLs in security forums We find a total of 637 misbehaving URLs including phishing, spamming, and malicious product sharing. Some of these activities are even aggressive: Nine URLs are aggressively promoted 1176 times by a group of 80 users.

(c) We identify significant collaborative groups which promote the same URLs Using tensor decomposition, we identify 30 tight-knit groups of users in terms of their URL posting over time. For example, we find a group of 17 users who jointly promoted decryption tools during a ransomware outbreak in December 2015. Interestingly, two of these users are associated with the creation of ransomware.

Our work in perspective The proposed work is part of an ambitious goal: We want to track down malicious hackers and understand their activities. Our initial results are promising: Malicious activities are prevalent in security forums. With appropriate follow-up work, achieving this goal can have a huge practical impact: Security analysts could prepare for emerging threats, anticipate malicious activity, and identify their perpetrators.

Open-sourcing for maximal impact We intend to make our tool and datasets public for research purposes. In our development, we use Python v3.6.2 and several related packages.

Lineage This paper is an extended version of our earlier 8-pages paper (Islam et al. 2021d). We outline the key

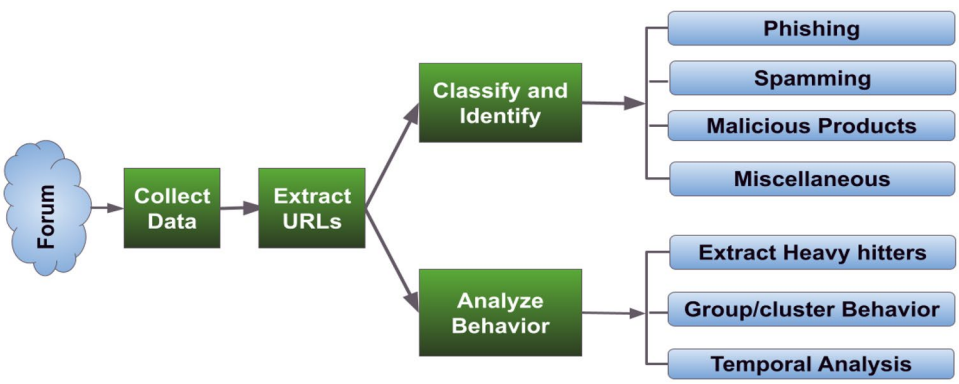

Fig. 1 Overview of the three-phase approach of HyperMan: **a** extract the URLs, **b** identify and classify URLs, and **c** model the misbehaving group behavior

additions and changes in this version. First, we describe our method more thoroughly and clarify the reason behind different algorithmic choices in Sect. 3. Second, we add two more datasets, a security forum and a gaming forum, in this journal version and we provide more detailed results for these additional datasets in Sect. 4, for example, Table 4. Third, we provide a temporal analysis of the URL sharing behavior and find the eventful days, for instance, Fig. 9. Fourth, we add additional results and comprehensive evaluation of our link extraction algorithm, *ExtLink*, in Tables 2 and 3. We also extensively evaluate our phishing detection algorithm, *RPhish*, in Table 6 where we compared our proposed method against different algorithmic choices and in Table 7 where we compared against three additional state-of-the-art methods. Fifth, we extend our discussion section, Sect. 5, where we discuss the scope, practical considerations, and limitations of our work. Finally, we update and improve the description of related works in Sect. 6.

2 Dataset and terminology

We provide a brief description of our dataset and explain the terminology used throughout this paper.

A. Dataset We use data from four security forums: Offensive Community, Hack This Site, Wilder Security, and Ethical Hacker (Online Forums 2021) spanning 5 years from 2013 to 2017. We also utilize another data of gaming forum, Multi-Player Gaming and Hacking Cheats (MPGH), spanning 2018. The datasets were collected by our early efforts which can provide more details (Islam et al. 2020b, a). The data of a forum consist of the following: forum ID, thread ID, post ID, username, date, and post content. These forums are in English, and their users discuss a wide range of security-related topics. The users range from security professionals to hobbyists, but some are also malicious hackers. Some basic statistics of the dataset are shown in Table 1. We briefly describe the discussion scope of our dataset below.

(a) Security forum dataset We also utilize data that we collect from four security forums: Wilders Security, Offensive Community, Hack This Site, and Ethical Hackers. In these forums, users initiate discussion threads in which other interested users can post to share their security-related opinion.

Table 1 Basic statistics of our datasets

Dataset	User	Thread	Post	Active day
Offensive Com.	5412	3214	23,918	1239
Ethical Hacker	5482	3290	22,434	1175
Hack This Site	2970	2740	20,116	982
Wilder Security	3343	3741	15,121	777
MPGH	37,001	49,343	100,001	289

i. OffensiveCommunity (OC) As the name suggests, this forum contains "offensive security"-related threads, namely breaking into systems. Many posts consist of step-by-step instructions on how to compromise systems and advertise hacking tools and services.

ii. HackThisSite (HTS) As the name suggests, this forum has also an attacking orientation. There are threads that explain how to break into websites and systems, but there are also more general discussions on cyber-security.

iii. EthicalHackers (EH) This forum seems to consist mostly of "white-hat" hackers, as its name suggests. However, there are many threads with malicious intentions in this forum.

iv. WildersSecurity (WS) The threads in this forum fall in the gray area, discussing both "black-hat" and "white-hat" skills.

(b) Gaming forum dataset We consider an online gaming forum, Multi-Player Gaming and Hacking Cheats (MPGH) (Online Forums 2021). MPGH is one of the largest online gaming communities with millions of discussions regarding different insider tricks, cheats, strategy, and group formation for different online games. The dataset was collected for 2018 and contains 100K comments of 37K users (Pastrana et al. 2018).

B. Terminology A *thread* is started by its first *post*, and we refer to subsequent posts as *comments*. The term *entity* refers to either a *user, thread, post* or day. If a *user, thread, post*, or day contains at least one hyperlink in the contents, we refer to them as '*LinkUser,*' '*LinkThread,*' '*LinkPost,*' or '*LinkDay,*' respectively.

3 Methodology

As our key contribution, we develop HyperMan, a systematic suite of capabilities, to detect URL-driven misbehavior in security forums. Fig. 1 demonstrates the overview of our approach. It consists of three high-level phases: (a) extracting URLs, (b) identifying and classifying misbehaving URLs, and (c) modeling behavioral patterns.

3.1 Phase 1: extracting URLs

We describe the challenges in extracting URLs from text.

Challenges in URL extraction Extracting URL is a non-trivial task because (a) URLs can follow a relatively complex structure, with significant diversity in the types of protocols and Top Level Domains, e.g., https, .com, .co, .io, (b) sometimes URLs include IP addresses and port numbers instead of text, and (c) human errors such as typos and missing characters can introduce noise. Here is a list of legitimate URLs: http://213.32.103.5/cgi-sys/defaultwebpage.cgi, google.com, https://facebook.com, www.facebook.com.

Limitations of using regular expressions Most well-established approaches extract hyperlinks from the HTML code, but this is not possible here since we have unstructured text. For such text, most studies (Pandya et al. 2018; Ahmad et al. 2016) and popular online tools (ConvertCSV 2021) use methods that are based on regular expressions (**RegEx**). These methods do not overcome the limitations we mentioned above, and we quantify this in our 'Results' section.

Our URL extraction method We propose *ExtLink*, a method to extract URLs of a variety of formats, types, and structures efficiently from textual data. Note that, we define a string to constitute a legitimate website URL if it represents a registered Top Level Domain (domain suffix), e.g., facebook.com is legitimate, while facebook.aa is not.

We start by tokenizing the raw text. Even this operation is not trivial if we want to handle typos, missed or added blank spaces, common misspellings, etc.

The next challenge is to determine if a token (word) is a legitimate URL, which we do in three steps. First, we identify candidate URL structures containing at least one dot, such as facebook.aa/groups. Second, we parse the structure to identify the domain name and the domain suffix, e.g., domain name = "facebook" and domain suffix = "aa," assuming that the input is an URL. Third, we validate our assumption by checking if the domain name and the domain suffix make a legitimate URL. In our example, "facebook.aa" is not a legitimate URL. We present the logic in deciding whether a token is a URL in Algorithm 1. We discuss the functional modules presented in Algorithm 1 later in the next section.

Algorithm 1: Algorithm to decide whether a given token is a legitimate URL.

```
Input: Token
Output: URL = True or False
URL = False
if isCandidate(token)== True then
    domain, domain-suffix = FindParts(token)
    if isLegit(domain-suffix) == False then
        if isValidIP(domain) == True then
            URL = True
        else
            URL = False
    else
        if isLegit(domain) == False then
            URL = False
        else
            URL = True
return URL==True
```

3.2 Phase 2: identifying and classifying misbehaving URLs

We classify the websites that we find from the previous step into the following categories: (a) phishing, (b) spamming, (c) malicious products, and (d) miscellaneous. The miscellaneous category includes websites that are benign or that we cannot confidently label as malicious.

Part 1. Detecting phishing We propose a novel machine learning-based method to detect phishing URLs. Phishing websites try to steal user account passwords or other confidential information by tricking visitors into believing they are on a legitimate website. Naturally, attackers attempt to attract as many visitors as possible to such websites.

Our phishing detection algorithm Our phishing URL detection approach consists of three high-level steps: (a) feature engineering using compression, (b) training the model, and (c) classifying the unknown URLs. In brief details, our phishing detection algorithm follows three simple steps: (a) compress all the m features to n features ($n < m$) using PCA, (b) train the compressed dataset with n features by feeding them into the input layer of a multilayer perceptron, and (c) finally, use the trained model (actually the weights, W_i's) to classify the unknown URLs as phishing/legitimate. We discuss the steps in details below.

(a) Feature engineering using compression As a first step, we compress the dimensions of our training data. The features in the training data spread along three dimensions as mentioned in the Introduction. The details of our ground truth data are described in the 'Results' section. We use the traditional, well-established dimension reduction algorithm PCA (Wold et al. 1987) to compress the features. We find that feature compression enhances the separability between the target classes (phishing and legitimate) and yields better performance demonstrated in Fig. 2. Note that, we also experimented with other feature compression algorithm like Sparse Random Projection and Gaussian Random Projection, but PCA yields the best performance which we discuss in the next section.

(b) Training the model After the dimension (feature) reduction, we train a neural network to detect phishing websites.

Neural Network Architecture. A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. It consists of some layers (input layer, hidden layers and output layer), each having some neurons in it which learns the weight of the parameters needed to classify. Each layer learns some specific patterns and the neurons adjust the parameters' weights using back-propagation algorithm and activation function (Hunt et al. 1992). That means, a neural network, also called multilayer

perceptron, is a set of connected input/output units where each connection has a weight associated with it. In summary, neural network is a multilayer perceptron where the input features, X_i"s are fed into the neurons of input layer followed by adjusting weights W_i's in hidden layers and finally presents its prediction decision, y_i's in the output layers.

Our neural network architecture consists of the input layer with three neurons, three hidden layers with 50 neurons in each, and one output layer with a single neuron (3-50-50-50-1). Out of all the architectures we tested, this architecture delivers the best performance, which we present in the 'Results' section.

(c) Identifying phishing URLs Finally, we use the trained model to classify the unknown URLs as either phishing or legitimate.

The details of the parameter choices of PCA and the neural network as well as the analysis of the model performance are discussed in the next section.

Part 2. Detecting spamming In general, the term spammer refers to a user who repetitively posts the same content. In our URL-centric study, the definition focuses on the repetitive posting of the same URL across many posts.

We follow the observations below to detect spamming URLs. First, spammers usually comment more than the general users. The frequency of the hyperlink contained in these *spam comments* generally outnumbers the frequency of hyperlinks contained in regular *comments*. That means the sole purpose of the spammers is to spam a hyperlink, which we call a '*dominant*' hyperlink, although they may rarely post other general hyperlinks as well. Therefore, a higher percentage of '*dominant*' hyperlinks in *comments* is a distinctive feature of spammers. Second, higher average similarity in all possible pairs of *comments* is also a distinctive feature of the spammers as they intentionally post the same content over and over again.

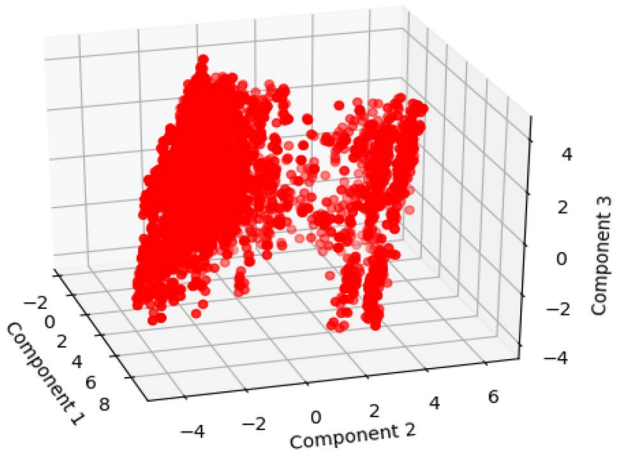

Fig. 2 An example that PCA reduces the dimension and enhances the separability. Here, number of compressed feature = 3)

In a nutshell, we consider a user to be a spammer if (i) the user posts a particular hyperlink more than a threshold, T_{freq}, (ii) his/her percentage of the *dominant* hyperlink is above a threshold, T_{dom}, and (iii) his/her average similarity in all possible pairs of *posts* is more than a threshold, T_{sim}. We then tag that user as 'spammer' and the corresponding *dominant* hyperlink as 'spamming hyperlink.' We present the choices of the thresholds and results from the security forums in the 'Results' section.

Part 3. Finding malicious products Users in security forums often advertise malicious hacking tool-selling websites which we refer to as 'malicious product' URLs. To identify URLs that share malicious products, we search for the URLs in a ranked list of malicious product-selling websites. First, there are several platforms, such as Alexa (Alexa 2021) and Hackerone (HackerOne 2021), which maintain extensive lists of well-known or popular websites of different categories. Second, an analysis of the landing page of a website can often indicate the type of services that it provides. This makes it possible to miss lesser-known websites, but as we will see later, we already find significant activities in this space. In this work, we use available lists of malicious product selling websites and will perform a content analysis of these websites in the future.

Part 4. Miscellaneous In this category, we place all the hyperlinks that we cannot confidently assign to any of the previous three categories. We attempt to classify them into the following three sub-categories: (a) technical security tutorials and information, (b) financial institutions and services, and (c) file and code-snippets sharing such as GitHub. Performing this classification is a challenging problem in its own right. Here, we leverage external resources that classify websites by their primary function. Specifically, we look for the URLs which are present in a list of Alexa top 500 (a) video sharing, (b) financial organization, and (c) file and code sharing websites. In the future, we intend to develop a more exhaustive solution for this category. We present our findings in the 'Results' section.

3.3 Phase 3: modeling behavioral patterns

HyperMan analyzes the behavioral patterns of users regarding the sharing of URLs. We answer the following three questions in this analysis.

(a) *What are the most popular URLs?* We detect the URLs that are promoted by a lot of individuals by finding outliers from a 2D scree plot where each point in the plot represents each URL's frequency (Y-axis) and the number of users that promoted the URL (X-axis) for each security forum. We use the DBSCAN algorithm to identify the outlier URLs. The findings are discussed in the 'Results' section.

(b) *How do the users collaborate to promote URLs?* We find groups of URLs that are promoted by groups of individuals over the time to understand the collaboration dynamics of URLs posting. We leverage tensor decomposition-based analysis, a very well-established approach to analyze group dynamics. To analyze the dynamics, we resort to the state-of-the-art tensor-based tool, TenFor (Islam et al. 2020b).

(a) Our tensor-based decomposition The input to TenFor is a 3D tensor T, where each 3D element, $T[i, j, k]$, captures the interaction of *LinkUser i* promoting URL *j* at weekly discretized time *k*. We provide the *LinkPosts* to TenFor as well. The output is a number of clusters where each cluster consists of a group of users, URLs, and weeks and captures a significant event or activity. For example, one such cluster in the OC forum represents a group of 29 users that are active in the first weekend of July 2016 and promotes a group of 43 video-sharing URLs for novice hackers. Figure 3 demonstrates another example cluster from OC forum. The Y-axis denotes the participation strength of the entities (user/ thread/ week) in that particular cluster.

(b) Identifying clusters of interest Our goal is to report surprising and interesting clusters. To do that, we perform the following three tasks. First, we create a 3D scree plot where each point denotes a cluster and each axis denotes the number of elements in each dimension (axis 1-number of users, axis 2-number of URLs, axis 3-number of week bins for that cluster) Second, we find the outlier clusters using DBSCAN outlier detection algorithm. Finally, we report the events hiding in the outlier clusters using "Storyline View" from TenFor, which captures a textual and visual summary of the essence of each cluster. We discuss the interesting findings from each forum in the next section.

(c) *When do the users promote URLs aggressively?* We answer the question by analyzing the trend of hyperlink sharing behavior. This analysis of URL posting over time can be a very important tool in finding interesting findings. In a particular eventful day, users might post a particular link or a group of links a lot of times. By analyzing the frequency of the links posted and *LinkPosts* in each day, we can extract the eventful days where a lot of links have been posted in different threads. We find the outlier *linkDates* where links have been posted a lot of times. We use *z*-score-based outlier detection algorithm to find the outlier/peculier *LinkDates*. Then, we report the links as well as their containing posts, threads, users along with their respective *LinkDates*. The "skylines" (tall bars) in Fig. 9 demonstrate that some *linkDates* really possess a lot of links posted by the users in different *LinkPosts*.

4 Results

We apply our method on the three security forums in our archive. HyperMan provides misbehaving URLs of different categories as well as other entities of interest (users, threads, posts, time intervals) associated with these. Table 4 shows the summarized output from HyperMan. We present the results, evaluations, and interesting findings from each category separately.

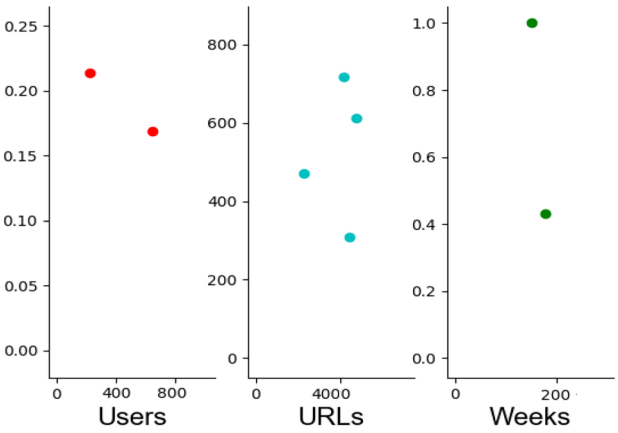

Fig. 3 An example of a 3D Tensor Decomposed cluster from OC forum (2 users, 4 URLs, 2 weeks)

4.1 Phase 1: extracting URLs

Our approach, *ExtLink*, extracts hyperlinks from the textual data of a forum. Specifically, upon tokenizing the posts using 'space' as the delimiter, we follow the steps presented in Algorithm1 to decide whether each token is a URL.

The *FindParts* function is responsible for factorizing the candidate token into domain name and domain suffix (TLD). In our current implementation, we use the 'tldextract' package of Python for factorization. If the domain suffix is not in '*Public Suffix List (PSL)*', a periodically updated list of private and public domain suffices available on Internet is maintained by Mozilla, it sets *domain suffix* = *empty* and the domain name accordingly (TLDExtract 2021). We find that 'tldextract' package is the most suitable tool for our purpose since other tools, such as 'purse_url' package of R language, do not utilize the PSL as 'tldextract' does.

We found that it was necessary to develop our own URL extraction method because traditional RegEx-based URL extraction methods have drawbacks that lead to poor performance. First, while generating the format of the URLs is easier using RegEx, manually configuring different types of protocols, TLDs, IP addresses(v4 and v6), optional port numbers, etc., becomes extremely hard and tedious to generalize. Second, using automated tools to generate RegEx by providing examples requires numerous handpicked examples which is a tough task. An alternative method of URL extraction is to use online tools such as convertCSV, browserlink, and miniwebtool, but they also use RegEx internally and can report URLs that adhere to a few specific formats only.

Table 2 The precision and recall of each method. Our *ExtLink* provides the highest precision without sacrificing recall

Method	Precision	Recall
RegEx	72.8	98.7
ConvertCSV	84.7	51.5
Browserlink	88.3	50.5
ExtLink	**99.5**	**98.7**

Table 3 The URLs extracted from each forum by each approach

Forum	*ExtLink*	RegEx	ConvertCSV	Browserlink
OC	22,599	24,577	8990	4573
EH	5458	6306	2450	1221
HTS	13,880	14,350	4687	2917
WS	5880	6359	3677	2902
MPGH	11,026	12,350	5681	4323

We evaluate *ExtLink* and report the findings below.

Evaluation Our evaluation suggests that *ExtLink* outperforms existing methods in terms of both precision and recall. We compare the performance with (i) **RegEx**-based approach, which we explain below, and (ii) popular online tools: **ConvertCSV** and **Browserlink** (ConvertCSV 2021; Browserlink 2021). For consistency, we use the same RegEx-based method that was used in previous works (Pandya et al. 2018; Ahmad et al. 2016). Our method is able to recognize many standard and commonly used URL formats: [scheme:][//authority]path[?query][#fragment] where scheme can be http, https, file, etc., and authority = [userinfo@]host[:port] (Regex 2021).

***ExtLink* outperforms RegEx: better precision with the same recall** We find that *ExtLink* outperforms the RegEx-based approach in terms of precision due to the RegEx results containing more false positives. For example, many RegEx-returned URLs are typos where the user failed to add a space at the end of a sentence. As a result, phrases like 'quickly.we' and 'go.I' are returned as URLs.

In the absence of established ground truth, we conducted the following study to quantify both precision and recall. We generated a set of 500 randomly selected posts, D_{post}, from the OC forum and manually extracted a total of 237 URLs from it. We applied all approaches on D_{post} and compared the outcomes. Table 2 summarizes the comparative results. We found that *ExtLink* extracted 234 valid URLs out of its 235 reported URLs yielding a precision of 99.5%. *ExtLink* missed only three URLs yielding a recall of 98.7%. RegEx yielded the same recall as *ExtLink* but a lower precision of 72.8% (234 valid out of 321 extracted URLs).

Table 3 shows the number of URLs from each security forum with *ExtLink* and reference methods. *ExtLink* identifies significantly more URLs than ConvertCSV and Browserlink. These online tools report fewer URLs because they use very stringent URL formats. RegEx returns the largest number of URLs from all three methods, but as we previously discussed, this comes at the cost of poor precision.

We utilize another systematic approach to assess the precision. We generate a set of 500 randomly sampled URLs, D_{url}, from the URLs detected by both *ExtLink* and RegEx and manually cross-check the validity of the URLs in D_{url} using another domain expert. *ExtLink* identified 498 valid URLs (precision 99.6%) from D_{url}, whereas RegEx found only 391 valid URLs (precision 78.2%).

Results from applying *ExtLink* on real data We find that our algorithm extracts a total of 58843 URLs (OC 22599, HTS 13880, EH 5458, WS 5880, MPGH 11026) from the security forums. We find a total of 7022 *LinkUsers*,

Table 4 Summary of the output of HyperMan. We showcase the aggregated numbers while reporting the results. Inside the cell, D. = Distinct

Output type	OC	HTS	EH	WS	MPGH	Total
Phishing URLs	27	21	9	7	15	79
Spamming URLs	4	3	7	14	9	37
Malicious Product URLs	178	168	220	1566	455	2587
Tutorial URLs	263	1447	699	409	300	3118
Financial URLs	32	5	3	7	1577	1624
File and Code URLs	201	456	137	794	2902	4490
Highly promoted URLs using scree plot analysis	9 D. URLs, 1176 Posts, 80 Users	17 D. URLs, 2756 Posts, 360 Users	5 D. URLs, 264 Posts, 125 Users	11 D. URLs, 196 Posts, 65 Users	20 D. URLs, 2007 Posts, 109 Users	62 D. URLs, 6399 Posts, 739 Users
Colluding clusters Tensor Decomposition	11 Clusters, 237 Users, 197 Links, 15 Weeks	9 Clusters, 145 Users, 154 Links, 13 Weeks	10 Clusters, 102 Users, 101 Links, 11 Weeks	30 Clusters, 484 Users, 452 Links, 39 Weeks	34 Clusters, 544 Users, 654 Links, 23 Weeks	94 Clusters, 1812 Users, 1558 Links, 101 Weeks
Eventful Days	43	32	32	29	45	181

of which 820 are from OC, 2264 are from HTS, 634 are from EH, 1003 from Ws, and 2301 from MPGH forum. In total, we report a sum of 8236 *LinkThreads*, 6702 *LinkPosts*, and 2986 *LinkDays* from the forums.

We provide some basic statistics that help us understand the URL posting behavior of the users in security forums. For instance, we find that 10% of the users are responsible for posting 77% of the URLs. The members of the OC community tend to share almost three times more URLs (4.17 URLs/user) than members of EH (1.83/user) and HTS (1.47/user). For OC forum, 82% of *LinkUsers* post at most six URLs, whereas only 0.1% post more than 700 URLs. Similar trends are observed for HTS, EH, WS, and MPGH forums as well.

4.2 Phase 2: identifying and classifying misbehaving URLs

After extracting the URLs, we classify them and report the types of misbehavior that could be of interest to a security analyst. Table 4 summarizes the results. Below, we evaluate our classification and highlight some key observations.

Part 1. Detecting phishing We show that our proposed phishing detection algorithm, *RPhish*, performs well. We describe the ground truth data, parameter and hyper-parameter choices, the performance of our model, and some interesting findings below.

Ground truth dataset We use the ground truth dataset from (Dua and Graff 2017) which contains 30 features including eight URL name-based features (e.g., 'containing IP address?,' 'has HTTPS?'), 11 network/ reputation-based features (e.g., 'HTTPS issuer trusted?,' 'using non-standard port?'), and 11 content-based features (e.g.,'has website forwarding?,' 'use pop-up windows?'). We present the list of the features in Table 5. Note that, the network-/reputation-based features are a feature set category where which based on whether the feature/s denote security metric/s in broader sense. For example, if an URL's HTTP issuer is trusted, that means its reputation is good. The details of these features can be found in (Dua and Graff 2017).

Parameter and hyper-parameter tuning As mentioned earlier, we use PCA followed by a Neural Network (NN) for the prediction. Among all compression algorithms such as PCA, Gaussian Random Projection (GRP), and Sparse Random Projection (SRP) that we try, PCA gives the best results in our study (Table 6). Furthermore, we show the effect of varying the compressed feature dimension, n, on accuracy and False Positive Rate in Fig. 4. We reach the best performance (accuracy 98.2%, FPR 1.3%) when we set $n = 3$ in PCA.

From other NN architectures that we experimented with, 3-50-50-50-1 architecture yields the best performance. We use the sigmoid and categorical cross-entropy as activation and loss functions, respectively. The training-to-test dataset ratio is 0.6:0.4. A comparison against other architectures with a different set of parameters that we tried is presented in Table 6. Finally, we use a drop-out ratio of 0.2 and tenfold cross-validation to ensure that our model is not suffering from over-fitting. We also manually verified all the phishing websites predicted from the security forums to verify that our model does not suffer from over-fitting.

***RPhish* outperforms the baseline approaches** We identified five methods, CANTINA (Zhang et al. 2007), Stacking Model (Li et al. 2019), Prasad (Prasad and Rao 2021), Deepa (Deepa et al. 2021), and Phishdef model (Le et al. 2011) as reference points for our approach. These five methods demonstrated consistently strong performance in their respective work. Table 7 summarizes the comparison results. The performance of *RPhish* (accuracy of 98.20%, TPR of 97.01%, and FPR of 1.3%) outperforms the existing best heuristic-based method CANTINA (accuracy of 95%,

Table 5 List of the features used for Phishing websites detection

Category A (Name Based)	Category B (network/reputation based)	Category C (content based)
Has IP Address?	HTTPS Issuer Trusted & Age of Certification	Favicon
Long URL?	Domain Registration Length	Request URL by Image/Video
Short URL?	Using Non-standard Port?	URL of anchor
Has @ Symbol?	Abnormal URL (WHOIS Search)?	# of Links in < *meta* >, < *script* > and < *link* >
"\\" Redirection	Age of Domain (WHOIS)	SFS Handler
Has "-" Symbol?	DNS Record	Submitting Information to email
Subdomain & Multi-domain status	Pagerank	Status Bar Customization
Has HTTPS?	Website Traffic Rank	Website Forwarding
	Google Index	Disabled Right Click?
	Number of Links pointing to that page	IFrame Redirection
	Statistical Report	Use Pop-Up Window?

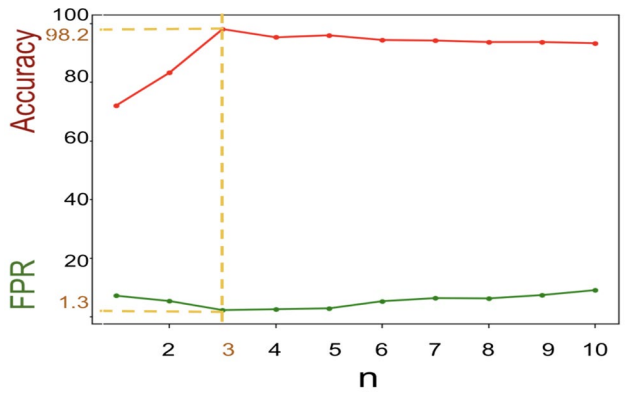

Fig. 4 The effect of the number of features, n, in our dimension-reduction on the performance on Accuracy and False Positive Rate results of PCA + NN. Selecting $n = 3$ yields the best performance

TPR of 95%, and FPR of 3%) and DNN-based work Stacking Model (accuracy of 97.2%, TPR of 95.3%, and FPR of 1.61%).

Applying our phishing detection method on the real dataset reveals interesting findings We find a total of 79 phishing URLs from all three security forums. Table 4 shows the breakdown of phishing URLs from each forum.

Some interesting findings about these phishing websites are (i) 41% (33) of these phishing sites are mimicking Facebook with the obvious intent of stealing user login credentials, (ii) 20% (15) are trying to mimic financial or e-commerce websites (Chase: 3, Discover: 3, Bank of America: 1, Amazon: 2, eBay: 1), and (iii) 37% (29) of these phishing websites URL names are typosquatting versions of the original domain name. Typosquatting is a form of cyber-crime where hackers create fake websites which names are a slight variation in the well-known websites (Banerjee et al. 2011).

To facilitate the future research work, we also demonstrate the posting frequency of the phishing websites available in our dataset in Fig. 5. We plot the frequency of the top 20 phishing websites posting frequency in non-increasing order. We find that the highest frequency corresponds to imitating facebook.

Part 2. Detecting spamming The intuition behind our spamming detection is to make use of the nature of spamming. We identify spammers and spamming hyperlinks based on high activity in terms of (a) hyperlink frequency, (b) percentage of *dominant* hyperlink, and (c) average post similarity. Figure 6 shows the percentage of *dominant* hyperlinks by users who post more than twenty hyperlinks in the OC forum. We can observe that four users, manually verified as spammers, demonstrate a higher percentage of *dominant* hyperlink. Similarly, Fig. 7 demonstrates that the same spammers exhibit significantly higher average post-similarity than others.

For the spamming detection algorithm, we use the following threshold values: $T_{freq} = 2*$ average hyperlink per *LinkUser* for the respective forum, $T_{dom} = 50\%$, and $T_{sim} = 50\%$.

Spamming in security forums We identify a total of 37 spammers of which four are from OC, three are from HTS, seven are from EH, 14 are from WS, and nine are from MPGH forum. These spammers usually try to advertise new websites. For example, in OC, user *Ghost_lite* promotes an adult site and user *Guru* aggressively advertises a domain 'mtgoox.com' which is for sale. In HTS, user *cyberdrain* tried to promote a new security forum, 'viphackforums.

Table 7 Comparison results: *RPhish* outperforms the prior approaches on our ground truth data

Approach	Accuracy	TPR	FPR	*F*-score
CANTINA (Zhang et al. 2007)	95.0	95.0	3.0	91.0
Stacking Model (Li et al. 2019)	97.2	95.8	1.6	95.0
Prasad (Prasad and Rao 2021)	92.9	91.4	6.5	92.3
Deepa (Deepa et al. 2021)	94.3	94.3	4.3	93.9
Phishdef (Le et al. 2011)	93	90.12	8.1	89.3
RPhish	**98.2**	**97.0**	**1.3**	**97.1**

Table 6 Phishing detection accuracy: Our phishing detection approach outperforms other approaches. (NN epochs=150, batch size= 128, SVM kernel=RBF). Here, n is the number of features to be compressed to in PCA

Architecture/model	Parameters/sub-algorithms	Layers	Accuracy
SVM (30 features)	Cache size- 200, probability- false, shrinking- True, kernel- rbf	N/A	92.57
SVM + PCA	PCA: n_component- 4, copy -True, svd solver- auto. SVM: Cache size- 200, probability- false, shrinking- True, kernel- rbf	N/A	96.27
NN(30 features)	sigmoid, sgd	3-50-50-50-1	92.20
NN + SRP	sigmoid, sgd PCA: n_component- 5	5-100-50-1	90.45
NN + GRP	sigmoid, sgd, n_component- 3	3-50-50-50-1	89.47
CNN (30 features)	softmax, maxpool, adam, poolsize- 2*2	Convolution layer: 64,128,128,128, 256,256,256 Fully connected: 1024,1	90.43
LSTM (30 features)	Drop out rate- 0.2, rmsprop, output activation- softmax	30-100-100-2	90.14
RPhish	sigmoid, $n = 3$	3-50-50-50-1	**98.20**

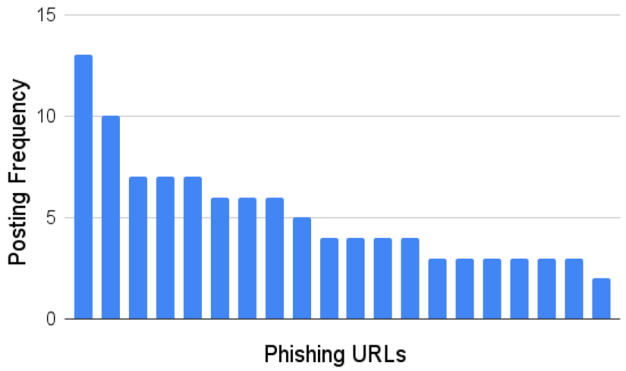

Fig. 5 The posting frequency of top 20 phishing website. X-axis denotes the phishing URLs

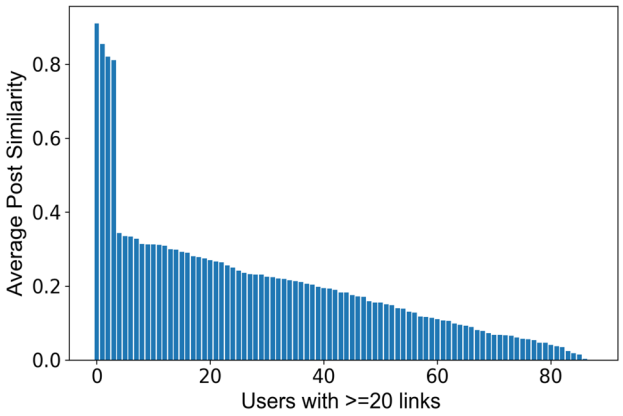

Fig. 7 Non-increasing average post similarity of the users who post more than twenty hyperlinks in their comments in OC. Four users have significantly higher average post similarity

net,' saying that the forum may go down if they can not find more users. In EH, user *Don* tried to disseminate the news of 'WannaCry' ransomware outbreak sharing his own post from 'bitshacking' forum and tried to sell an anti-ransomware tool in 2017. In MPGH, user *Bob1177* tried to promote a crypto-currency aggressively. We manually check these *spamming posts* and verify them as spam.

Part 3. Finding malicious products In our current implementation, we characterize URLs by using databases that classify websites. Here, we use the Alexa list of the top 500 websites that sell malicious products. The marketing of such products is dynamic. We find that the number of URLs for both malicious and defensive tools increases during major events of malware outbreaks as we discuss below.

Malicious products in security forums We detect 2587 URLs pointing to websites that sell malicious products across our three security forums. In OC, a group of seven users tried to promote a WiFi hacking tool from 'virustotal' in January 2015. In HTS, another group of 23 people recommended a decryption tool from 'sourceforge' in the event of spreading 'Locky' ransomware in February 2016. We observe a surge of malicious product related URLs in WS forum where users basically promoted decryption tools for ransomware attack. Thus, a peak in the posting of these URLs suggests the outbreak of the major security events (Gharibshah et al. 2020), which serve as interesting indirect information for a security analyst.

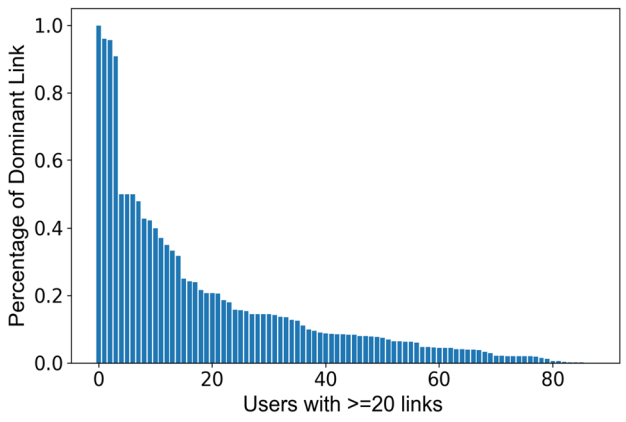

Fig. 6 Non-increasing percentage of *dominant* hyperlink of the users who post more than twenty hyperlinks in their comments in OC. Four users have significantly higher *dominant* hyperlink percentage

Part 4. Miscellaneous We place here all URLs that do not belong to any of the above-mentioned misbehavior categories. We further sub-categorize these URLs to understand more about them. We find a total of 3118 URLs of technical security tutorials and information, 1624 financial institutions and services-related URLs, and 4490 file and code-snippet sharing related URLs. Table 4 shows the breakdown of these sub-categorized URLs.

(a) Technical security tutorials and information We detect 3118 security tutorial URLs from all five forums. We report some of the interesting findings here. Among the five forums, users of HTS are dominant in tutorials link (1447) sharing. In OC, user *Dragunman* shared tutorials on hacking into banks throughout June 2015. User *-Ninjex-, andm-Shred* in HTS shared youtube tutorials for building hacking tools throughout the month of August 2014. *VandaDGod*, an expert Linux hacker, shared a popular tutorial series on Hacking in Kali Linux in November 2017 in EH.

(b) Financial institutions and services Posting URLs to financial institutions seems to be related to the selling of security or hacking tools taking advantage in the event of malware outbreaks. MPGH forums are ways ahead in terms of financial institutional link sharing because a lot of gaming coins and online ready-made accounts have been bought and sold in this forum. In security forums, some users take advantage of malware outbreaks to sell their products and point to the financial services of their choice. For example,

one interesting finding is that, among the 32 mentions of financial URLs in OC, 21 of them occurred in the month of December 2015, and February 2016. At that time, ransomware named 'SimpleLocker' outbroke and some users tried to sell decryption tools for it. These sellers' post followed almost the exact same pattern: '*Decryption key for SimpleLocker. Payment only in chase.com*'. In WS and EH, two groups mainly talked about the security strength of 'Commbank' website as they try to hack into the system.

(c) File and code-snippet sharing Forum users frequently share different files and code-snippets in data storage platforms such as GitHub, Bitbucket, and MediaFire. We report 4490 URLs and associated entities from the file and code sharing sub-category. Interestingly, we find one URL pointing to 'MediaFire.com' in 2015 claiming that it allows the hosting of many malware and viruses. As expected, MPGH forum users also shared many Google doc link which contains many gaming hacks and cheats.

4.3 Phase 3: modeling behavioral patterns

Although there are many interesting behaviors that we can consider, due to space restrictions, we are forced to narrow our discussion down to the following three questions.

(a) *What are the most popular URLs?* We detect outlier URLs from a 2D scree plot for each security forum. Figure 8 presents the scree plot from OC where each point represents each URL's frequency (Y-axis) and the number of users promoting that URL (X-axis).

Identifying heavy-hitting URLs using scree plots A total of 62 URLs are found from all three forums. The breakdown is presented in Table 4. We highlight three observations. First, in OC, a group of 18 users promoted 'crackcommunity.c0.pl' 577 times to help them have some traffic because someone reported against this forum. Second, two users solely tried to promote 'vn5socks.net' in different *posts* 607 times in WS. Third, in EH, four users announced some rule change in 'hackthissite.com,' because they were administrators of HTS but regular users of EH (Fig. 9).

(b) *Is there any collaborative behavior in promoting URLs?* We identify groups of users which seem to work synergistically to promote groups of URLs. To capture this behavior systematically, we use TenFor (Islam et al. 2020b), which utilizes the temporal dimension.

Tensor analysis findings from security forums We find a total of 94 clusters including 19 surprising and interesting clusters (four from OC, three from HTS, three from EH, four from WS, and five from MPGH). All of the entities in these clusters are reported in HyperMan.

We manually investigate 10 peculiar clusters and found them interesting. For instance, one cluster in OC is a group of 13 users promoting 43 hacking tutorial URLs in June 2016, and again in August 2016. Another outlier cluster in

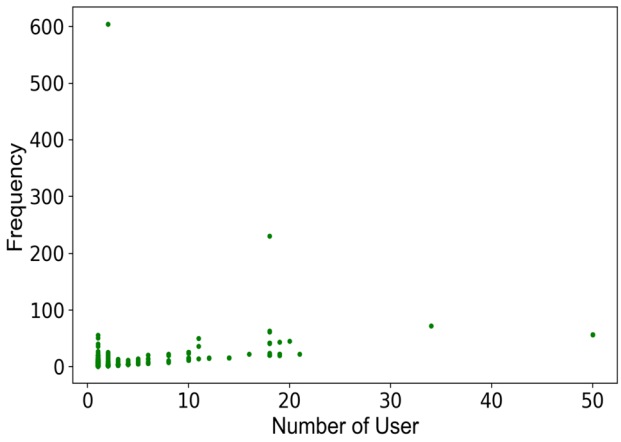

Fig. 8 Scree plot for OC forum. Each point represents each URL's frequency (Y-axis) and the number of user promoted that URL (X-axis). We find nine URLs are aggressively promoted 1176 times by a group of 80 users

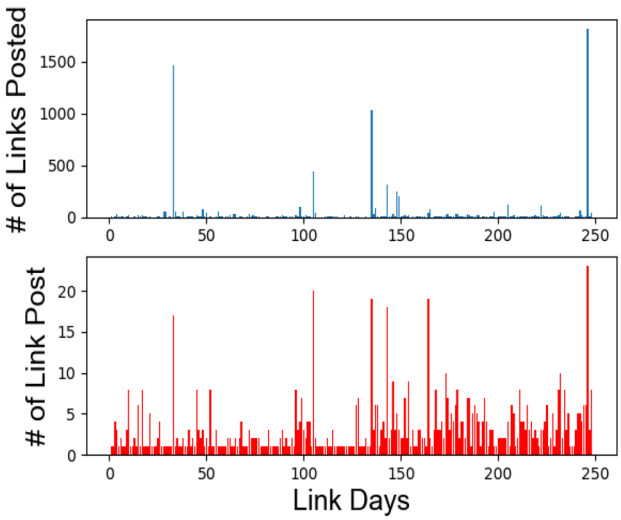

Fig. 9 [Upper] # of links posted in *LinkDates*. [Lower] # of posts that contain at least one link in *LinkDates*

HTS consists of only two users (administrators) who are promoting a URL pointing to the posting rules of HTS in 10 different weeks again and again when anything severe happens in the security world. Another outlier cluster in EH consists of 17 black market decryption tool sellers posting 19 hyperlinks to their respective selling websites in December 2015, and February 2016, correlating with the 'SimpleLocker' ransomware outbreak.

(c) *When do the users promote URLs aggressively?* Our temporal analysis basically detects the abnormal peaks in number of URL posting. We find 181 eventful days from all forums when a huge number of links have been posted using z-score-based anomaly detection algorithm ($z = -3, 3$). We

go through every *post* that incorporates at least one link in that eventful days. We find each of the eventful day is really important to investigate through. For example, in OC, user '*Montana*' posted a list of 2503 HTTP proxy servers in March, 2016 which was really praised by the group members. User '*AN0N.PH03N1X*' posted over 2700+ premium adult site accounts hacked in March, 2016. Analyzing the posts of both '*Montana*' and '*AN0N.PH03N1X*,' we find that they are from same region and there was a ban on certain websites imposed by their government which triggered them to look for alternative ways to access those banned sites. Another user '*Drax00*' published a list of 1500+ vulnerable websites to be hacked which sparked a discussion among the users in September, 2015. We find similar types of events in HTS and EH as well. In conclusion, our analysis successfully identifies these eventful days as well as the links, *LinkUsers*, *LinkThreads* and *LinkPosts* associated with these days.

5 Discussion

We discuss the practical considerations and limitations of our approach.

(a) How do we handle misspelled hyperlinks? In our current implementation, *ExtLink* extracts URLs with the correct format and spelling. Our rationale is that we detect URLs that would lead to a website if they were copy-pasted to a browser. In this work, we do not report URLs with erroneous formats or typos such as '*google.co*', '*http:/www.google.com*', and '*ww.127.0.0.1*'. Considering erroneous URLs introduces a trade-off where we could end up reporting URLs that are (a) plain wrong or non-functional or (b) not intended to be used, e.g., *www[dot]malware[dot]com*. In this instance, saying that the user is sharing malware would be incorrect.

(b) Is our tool generalizable to forums other than security forums? Our approach can work with any online forum, not just security forums. We have tested HyperMan on different types of forums, including a gaming forum and observed interesting findings. Focusing on security forums is interesting in its own right: (i) These forums are hardly explored, and (ii) previous works suggested that they hide a wealth of information including malicious activities (Rokon et al. 2020; Islam et al. 2021b; Gharibshah et al. 2020; Portnoff et al. 2017).

(c) Can we extract all kinds of entities of interest in a forum? Our tool is capable of extracting not only the URLs of interest but also other entities, for example, the users, thread, posts, and times of interest. Moreover, our behavioral analysis extracts the group dynamics of the entities of interest that a security analyst can easily gauge through.

(d) Are the entities that we report interesting and investigation-worthy? Our systematic and manual evaluation via experts clearly suggests that the findings are very much investigation worthy.

(e) Does our spamming detection mechanism detect only malicious spamming hyperlinks? We only detect the URLs as spamming if a certain URL's posting frequency as well as the encapsulating same post surpasses a tolerance level. In that sense, even a legitimate URL can fall in spamming category if the post containing the URL is duplicated and posted with only a minor change over and over again at an annoying level. Also, note that since our method counts frequency for spamming detection, we consider 'similar' domain names or URLs as separate URLs.

(f) Can our tool identify misinformation? Among the misbehavior that we detect in this work, phishing is a kind of misinformation that we report. The definition of misinformation is very broad and, hence, exact focus on misinformation detection is out of the scope of this work.

(g) Do our datasets represent actual online forum data? We utilize the data of security forums spanning five years from previous studies. We focused primarily on proposing novel methods for URL extraction, misbehavior detection and, overall, a systematic and comprehensive tool to detect and report hyperlink-driven misbehavior in online forums. In our future study, we opt to collect most recent data and apply clustering or network link analysis methods to further dig into the dynamics of the reported misbehavior.

(h) How can a practitioner use our tool? Our approach is ready for use in a straightforward way. Our plan is to share our code and datasets. Therefore, a practitioner needs to only provide the forum dataset: F:=(thread ID, post ID, username, date, and post content). HyperMan takes care of the rest and generates the desired output.

6 Related work

Studying hyperlink-driven misbehavior in security forums has received very little attention from the research community. Most studies differ from our work in that either: (a) they do not focus on detecting misbehavior online security forums or (b) they focus on identifying key players, threads, or events without an analysis from the perspective of hyperlink posting.

We briefly discuss the related works below.

(a) Online security forum studies This is a recent and less studied area of research. Some recent studies focus on identifying key actors and emerging concerns in security forums using supervised techniques and NLP by analyzing social and linguistic behavior (Marin et al. 2018; Rokon et al. 2021). Some of these works are empirical studies that do not develop a systematic methodology. Recent efforts

include analyzing the dynamics of black market hacking services (Portnoff et al. 2017). (Gharibshah et al. 2018) extracts the malicious IP addresses reported by users from security forums. (Gharibshah et al. 2020) classifies the threads from online forums given keywords of interest. (Islam et al. 2020b, 2021c, b) finds the cluster of interest and identifies the events of interest from online platforms. Moreover, in another work (Islam et al. 2020a, 2021a; Jonas et al. 2019), they analyzed the hacker dynamics from GitHub and threats from online platforms.

Our work is different from these efforts in the sense that we focus on the identification of hyperlink-based misbehavior in a comprehensive way.

(b) Hyperlink classification studies There are only a few works that extract URLs from raw text (Pandya et al. 2018; Ahmad et al. 2016). They focus on the age prediction of Twitter users by analyzing the contents in the URLs (Pandya et al. 2018) and on extracting information from scientific paper PDF files (Ahmad et al. 2016). These works place no emphasis on detecting misbehavior.

Most of the misbehavior detection works do not develop a systematic suite of capabilities. Instead, they focus on detecting only single type of misbehavior such as phishing. Many of these works are heuristic-based (Zhang et al. 2007) and machine learning-based (Li et al. 2019; Prasad and Rao 2021; Deepa et al. 2021). They exhibit good performance in terms of some selective performance metrics but tend to suffer from overfitting and/or low TPR and/or high FPR. CANTINA (Zhang et al. 2007) utilizes a bunch of heuristics and uses TF-IDF to gain the highest accuracy of 95%. Machine learning-based work (Li et al. 2019) utilizes a DNN Stacking Model to gain an accuracy of 97.2%. (Prasad and Rao 2021) proposed a hybrid model to detect phishing weblinks to gain an accuracy of 9%, while (Deepa et al. 2021) proposed a machine learning-based model to gain an accuracy of 94.3%. Phishdef (Le et al. 2011), on the other hand, utilizes only URL name-based features showing an average accuracy of 93%. Our proposed method, *RPhish*, is unique because it first compresses the features and then uses a NN to gain the best performance.

Very few works focus on detecting misbehavior from online platforms, and those that do not take the URL posting perspective like ours. These works focus on binary classification of misbehaving or benign user detection (Li et al. 2017), deteriorating and non-deteriorating behavior prediction (Tshimula et al. 2020), or medical misbehavior of drug non-compliance detection (Bigeard and Grabar 2019). None of the above-mentioned studies focus on systematic classification and identification of misbehavior from a hyperlink sharing perspective in online forums.

7 Conclusion

We propose and develop HyperMan, a comprehensive suite of capabilities to systematically identify URLs from text and URL-driven misbehavior from online security forums.

Our approach has the following main advantages: (a) it extracts URLs from raw text effectively with high precision and recall, (b) it classifies hyperlinks in an accurate and comprehensive way, and (c) it explores the entities and the collaborative behavior of users utilizing the power of tensor decomposition.

The current work is a building block to mine the wealth of information that exists in online forums. Follow-up efforts can use our approach to (a) monitor hacker activity, (b) detect emerging trends, and (c) identify influential hackers toward safeguarding the Internet.

Acknowledgements This work was supported by the UC Multicampus-National Lab Collaborative Research and Training (UCNLCRT) award #LFR18548554.

References

Ahmad R et al (2016) Information extraction from pdf sources based on rule-based system using integrated formats. In: Semantic web evaluation challenge. Springer, pp 293–308

Alexa (2021) Alexa web ranking. https://www.alexa.com/siteinfo/. Accessed 2-June-2021

Banerjee A et al (2011) Sut: quantifying and mitigating url typosquatting. Comput Netw 55(13):3001–3014

Bigeard É, Grabar N (2019) Detection and analysis of medical misbehavior in online forums. In: SNAMS, IEEE, pp 7–12

Browserlink (2021) Popular online url extractor. https://www.browserling.com/tools/extract-urls. Accessed 2-June-2021

ConvertCSV (2021) Popular online url extractor. https://convertcsv.com/url-extractor.htm,browserling.com/tools/extract-urls. Accessed 2-June-2021

Deepa S et al (2021) Phishing website detection using novel features and machine learning approach. TURCOMAT 12(7):2648–2653

Dua D, Graff C (2017) UCI machine learning repository. https://archive.ics.uci.edu/ml/datasets/phishing+websites

Gharibshah J, Papalexakis EE, Faloutsos M (2020) REST: a thread embedding approach for identifying and classifying user-specified information in security forums. ICWSM

Gharibshah J et al (2018) RIPEx: Extracting malicious ip addresses from security forums using cross-forum learning. In: PAKDD. Springer

HackerOne (2021) Hackerone: top 100 haking tools. https://www.hackerone.com/blog/100-hacking-tools-and-resources. Accessed 2-June-2021

Hunt KJ, Sbarbaro D, Zbikowski R, Gawthrop PJ (1992) Neural networks for control systems-a survey. Automatica 28(6):1083–1112

Islam R, Rokon MOF, Darki A, Faloutsos M (2020a) Hackerscope: the dynamics of a massive hacker online ecosystem. In: 2020 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM), pp 361–368

Islam R, Rokon MOF, Papalexakis EE, Faloutsos M (2020b) Tenfor: a tensor-based tool to extract interesting events from security forums. In: 2020 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM), pp 515–522

Islam R, Rokon MOF, Darki A, Faloutsos M (2021) Hackerscope: the dynamics of a massive hacker online ecosystem. Soc Netw Anal Min 11(1):1–12

Islam R, Rokon MOF, Papalexakis EE, Faloutsos M (2021b) Recten: a recursive hierarchical low rank tensor factorization method to discover hierarchical patterns in multi-modal data. In: Proceedings of the international AAAI conference on web and social media

Islam R, Rokon MOF, Papalexakis EE, Faloutsos M (2021c) Tenfor: Tool to mine interesting events from security forums leveraging tensor decomposition. Lecture Notes in Social Networks

Islam R, Treves B, Rokon MOF, Faloutsos M (2021d) Linkman: hyperlink-driven misbehavior detection in online security forums. In: Proceedings of international conference on advances in social network analysis and mining (ASONAM). IEEE/ACM

Jonas MA, Hossain MS, Islam R, Narman HS, Atiquzzaman M (2019) An intelligent system for preventing ssl stripping-based session hijacking attacks. In: MILCOM 2019-2019 IEEE military communications conference (MILCOM). IEEE, pp 1–6

Knot A (2021) Hackers posing as mcafee antivirus. shorturl.at/mtuAS. Accessed 2-June-2021

Le A, Markopoulou A, Faloutsos M (2011) Phishdef: Url names say it all. In: 2011 Proceedings IEEE INFOCOM, IEEE, pp 191–195

Li TC et al (2017) Trollspot: Detecting misbehavior in commenting platforms. IEEE/ACM ASONAM 2017:171–175

Li Y, Yang Z, Chen X, Yuan H, Liu W (2019) A stacking model using url and html features for phishing webpage detection. Future Gener Comput Syst 94:27–39

Marin E et al (2018) Community finding of malware and exploit vendors on darkweb marketplaces. In: ICDIS, IEEE, pp 81–84

Online Forums (2021) Ethical hacker, hack this site, offensive community, wilders security. https://www.ethicalhacker.net/, https://www.hackthissite.org/, http://offensivecommunity.net/, https://www.wilderssecurity.com/, https://mpgh.net/

Pandya A et al (2018) On the use of urls and hashtags in age prediction of twitter users. In: IEEE IRI, pp 62–69

Pastrana S, Thomas DR, Hutchings A, Clayton R (2018) Crimebb: Enabling cybercrime research on underground forums at scale. In: WWW, pp 1845–1854

Portnoff RS, Afroz S, Durrett G, Kummerfeld JK, Berg-Kirkpatrick T, McCoy D, Levchenko K, Paxson V (2017) Tools for automated analysis of cybercriminal markets. In: WWW, p 657

Prasad SDV, Rao KR (2021) A novel framework for malicious url detection using hybrid model. TURCOMAT 68–76

Regex (2021) Regular expression format. https://en.wikipedia.org/wiki/URL/. Accessed 2-June-2021

Rokon MOF, Islam R, Darki A, Papalexakis EE, Faloutsos M (2020) Sourcefinder: Finding malware source-code from publicly available repositories in github. In: 23rd International symposium on research in attacks, intrusions and defenses ({RAID} 2020), pp 149–163

Rokon MOF, Yan P, Islam R, Faloutsos M (2021) Repo2vec: a comprehensive embedding approach for determining repository similarity. In: 2021 IEEE international conference on software maintenance and evolution (ICSME). IEEE

Sidonce J (2021) Estimated number of online forums and users. https://quora.com/How-many-online-forums-are-in-existence. Accessed 2-June-2021

TLDExtract (2021) Tldextract package. https://github.com/john-kurkowski/tldextract. Accessed 2-June-2021

Tshimula JM et al (2020) On predicting behavioral deterioration in online discussion forums. In: IEEE/ACM ASONAM, pp 190–195

Wold S et al (1987) Principal component analysis. Chemom Intell Lab Syst 2(1–3):37–52

Zhang Y et al (2007) Cantina: a content-based approach to detecting phishing web sites. In: 16th international conference on World Wide Web, ACM, pp 639–648