UNIVERSITY OF CALIFORNIA
RIVERSIDE

Extracting Actionable Information From Security Forums

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Joobin Gharibshah

December 2020

Dissertation Committee:

Professor Michalis Faloutsos, Chairperson
Professor Vassilis Tsotras
Professor Eamonn Keogh
Professor Vagelis Hristidis

The Dissertation of Joobin Gharibshah is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I would like to express my sincere gratitude to my advisor, Professor Michalis Faloutsos, for his patience, motivation ,and his continuous support throughout my graduate study. He has always supported me not only by providing a research assistantship but also academically and emotionally through the rough road to finish this thesis. He genuinely cares about me as his student and believes that I can succeed. Michalis was not only my academic advisor but also he was a great friend of mine who has been by my side to help in all the steps. My PhD life was a great opportunity for me to learn a lot from Michalis. I wish I could have graduated in a world without COVID19 so that I could tell Michalis all these in person. Kudos to Michalis the best advisor I could have had in my Ph.D.

I am thankful for the committee members, Prof. Vassilis Tsotras, Prof. Eamonn Keogh, and Prof. Vagelis Hristidis for their constructive comments and inputs. Their comments helped me shape up my dissertation and the research topic. Also, I would like deeply thank Prof. Vagelis Papalexakis for his insightful comments in my research.

I would like to thank my co-authors who have helped me in publications: Prof. Vagelis Papalexakis, Prof. Konstantinos Pelechrinis , Tai Ching Li, Andre Castro, Maria Solanas Vanrell.

I would like to further thank individuals and friends who helped me obtain the technical skills necessary for my research and for all the inspirational discussions which helped shape me into the person I am now. In particular, I am gratefully indebted to Mehdi Hosseini, Arsalan Mousavian, Ahmad Darki and Hossein Mostafavi.

I would like to thank my fellow group-mates, Ahmad Darki, Pravallika Devineni,

Tai Ching Li, Andre Castro, Jakapun Tachaiya, Risul Islam ,Sri Shaila G, Md Omar Faruk Rokon for the constructive discussions we had during our group meetings and for all fun we had during the last five years.

There are countless number of friends who supported me during this journey to the end of my Ph.D. If I could have the space to name all of them, that would lead to an additional 10 pages. However, I will try to name very few among them. I would like to start with my good friends: Arsalan Mousavian, Ahmad Darki, Farhad Eftekhari, Navid Abbaszadeh, Ramin Raziperchikolaei, Bahman Pedrood, Paniz Mozaffari, Ardavan Sassani, Mehdi Hosseini, Hossein Mostafavi , shabnam Omidvari, Abbas Roayaei, Amir Feghahati, Nami Davoodzadeh, Hadi Zamani Sabzi, Shahryar Afzal, Hodjat Asghari, Nasim Eslami, Aria Hosseini, Sina Davanian, Pravallika Devineni, Yue Cao, Nhat Li, Hoang Anh Dau, Jose Rodriguez, Kittipat Apicharttrisorn, Shailendra Singh, Jason Ott, and many more. Lastly, but most importantly, I would like to thank specially to Najmeh whose help and support was endless in my Ph.D. journey.

I cannot put it into words how grateful I am to my family who helped me become the person I am now. In particular, I would like to thank my lovely mother Sedigheh Hosseinpour who made education my first priority and kept me on the road to academic success up until now by reminding me of my abilities and values. My great brother Zhabiz Gharibshah for encouraging me to be the best I can and supporting me in my challenges. This PhD and dissertation were not possible without my family support!

To my mother for all her supports.

ABSTRACT OF THE DISSERTATION

Extracting Actionable Information From Security Forums

by

Joobin Gharibshah

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, December 2020
Professor Michalis Faloutsos, Chairperson

The goal of this thesis is to systematically extract information from security forums, whose information would be in general described as unstructured: the text of a post is not necessarily following any writing rules. By contrast, many security initiatives and commercial entities are harnessing the readily public information, but they seem to focus on structured sources of information. Here, we focus on analyzing text content in security forums to extract actionable information. Specifically, we search and find: IP addresses reported in the text, study keyword-based queries, and identify and classify threads that are of interest to the security analysts.

The power of our study lies in the following key novelties. First, we use a matrix decomposition method to extract latent features of the user behavioral information, which we combine with textual information from related posts. Second, we address the labeling difficulties by utilizing a cross-forum learning method that helps to transfer knowledge between models. Third, we develop a multi-step weighted embedding approach, more specifically, we project words, threads, and classes in appropriate embedding spaces and establish relevance and similarity there. These novel approaches enable us to extract and refine information which could not be obtained from security forums if only trivial analyses were used.

We collected a wealth of data from six different security forums. The contribution of our work is threefold: (a) we develop a method to automatically identify malicious IP addresses observed in the forums; (b) we propose a systematic method to identify and classify user-specified threads of interest into four different categories; and (c) we present an iterative approach to expand the initial keywords of interest which are essential feeds in searching and retrieving information.

We see our approaches as essential building blocks in developing useful methods for harnessing the wealth of information available in online forums.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this thesis, we study discussion forums, which have emerged as a widely-used but little-studied type of online social interaction. Users in such forums are sharing their ideas around specific topics or asking their questions and solving their issues with the help of each other. Approximately, there are more than 5 million active users in such social networks who contribute to more than 50 million posts per week. This volume of activity shows us a great opportunity in extracting the information in these discussions which is the main goal of this research. In this study, we focused on the specific type of discussion forums which are related to security and hacking activities. we will use the term “security forums” to describe online discussion forums with a focus on security, system administration, and general systems-related discussions. Security forums hide a wealth of information, but mining it requires novel methods and tools. The overarching goal of this thesis is to harness the user-generated content in security forums, and answer this question: ”How can we extract actionable information form security forums?” In these forums, security professionals, hobbyists, and hackers identify issues, discuss solutions, and in general exchange information. Here, we focus on the problem of analyzing text content in security forums. We consider the content of forums as the input and our approaches here could extract useful information that can be used by security analysis such as malicious IP addresses, different classes of threads of interest, and also keywords of interest to facilitate search action in the forums.

Studying the security forums comes with a unique set of challenges that we briefly outline here.

a. Data collection: The security forums are the types of social network which are not studied widely and the publicly available data for such social networks is limited. As the first challenge, it is needed to collect the data by crawling the web.

b. Establishing the ground-truth: There is no publicly available labeled data on security forums. It was a big challenge to prepare labeled data in the supervised task like identifying malicious IP addresses and classifying threads of interest.

c. Handling unstructured data: The text of posts in these forums is not essentially following any writing rules. Therefore, the information in such forums is described as unstructured. There is a challenge to clean and represent text data properly to be suitable as feeds into models.

d. Compensating for noisy data: Designing an efficient approach to solve the research questions on user generated data was another challenge, as it requires algorithms that can handle noisy data.

There is limited work on extracting information from security forums and even less work on extracting malicious activity and analyzing text with embedding techniques in such forums. We can group prior work in the following categories. First, there is limited work to focus on security forums and study them. In one recent work, they study the number of malicious IP addresses in forums, but without providing a comprehensive and systematic solution that we propose here [39]. Moreover, there are some efforts to detect malicious users [61, 63], and emerging threats on security forums. Second, there is a large body of work in analyzing the text with utilizing embedding techniques for understanding the context and improving text classification methods [65, 54, 32, 100, 96] but there is no previous work that uses such techniques to analyze and extract text-based information from security forums, specifically. Third, other works focus on analyzing structured sources, such as security reports and vulnerability databases [19, 52] and discussion forums in general without a security focus [109, 24]. We discuss related work extensively in each chapter of the dissertation.

In our work, we address a set of important questions regarding studying security forums. In particular, we want to extract as much useful information from security forums as possible in order to perform (possibly early) detection of potentially malicious content such as IP addresses and threads of interest to the security analysts. In this thesis, we are looking to analyze the forums to identify and classify the content of interests. Threefold of the interest here are IP addresses and user-specified security discussion threads and keywords for a search. We answer the following research questions in this thesis.

1) Is it possible to extract malicious IP addresses reported in security forums in an automatic way?

We propose InferIP, a systematic method to identify malicious IPs among the IP addresses which are mentioned in security forums. A key novelty is that we use the behavioral information of the users, in addition to the textual information from the related posts. We customize and use a Sparse Matrix Regression method on this expanded set of features. By design, our framework applies to forums in different languages as it relies on and the behavioral patterns and keywords and not a complex language-specific NLP technique. From a technical point of view, the challenge in designing a solution to our Key Question is most IPs mentioned in these forums are not malicious. We show that our system can add a significant number of previously unreported IP addresses to existing blacklist services.

2) How can we transfer knowledge between forums to identify and classify malicious IP addresses?

We propose RIPEx, a systematic approach to identify and label IP addresses in security forums by utilizing a cross-forum learning method. In more detail, the challenge is twofold: (a) identifying IP addresses from other numerical entities, such as software version numbers, and (b) classifying the IP address as benign or malicious. We propose an integrated solution that tackles both these problems. A novelty of our approach is that it does not require training data for each new forum. Our approach does knowledge transfer across forums: we use a classifier from our source forums to identify seed information for training a classifier on the target forum.

3) How can we extract threads of interest to security analysts from a security forum?

We propose REST, a systematic approach to identify and classify threads of interest based on an embedding approach. We consider two associated problems that together provide a complete solution to this problem. First, the input is all the data of a forum, and the user specifies its interest by providing one or more bag-of-words of interest. The goal is to return all the threads that are of interest to the user, and we use the term ”relevant” to indicate such threads. Second, we add one more layer of complexity to the problem. To further facilitate the user, we want to group the relevant threads into classes. We utilize the embedding domain which captures the similarity and the context of words to represent in multi-dimensional space. We refer to this step as the Characterization problem. Given a security forum, we want to extract threads of interest to a security analyst.

4) What are the relevant keywords to search in a forum so that we can maximize the amount of useful information that we can extract?

We propose IKEA, an iterative embedding-based approach to expand a set of keywords with a domain in mind. The novelty of our approach is three-fold: (a) we use two similarity expansions in the word-word and post-post spaces, (b) we use an iterative approach in each of these expansions, and (c) we provide a flexible ranking of the identified words to meet the user needs. A possible application is to use the expanded set of words to search for specific information within the domain of interest.

1.1 Road map

This dissertation consists of the following main chapters. Chapter 2 describes the matrix decomposition approach and latent feature extraction to identify malicious IP addresses (ASONAM 2017) [42]. Chapter 3 discusses cross seeding approach to identify and classifying IP addresses in security forums (PAKDD 2018)[40]. Chapter 4 discusses multi-step weighted embedding to identify and classify threads of interest to security analysts (ICWSM 2020)[44]. Finally, Chapter 5 discusses iterative keyword expansion approach to identify keyword of interest to analysts for searching in forums, and Chapter 6 concludes the work.

The overarching vision is to provide a powerful, and flexible method to extract useful information from security forums to help analysts study and analyze such forums better. We see our approach as a key capability within a practical tool-set for harnessing the wealth of information in online forums with the goal of informing a security analyst.

Chapter 2

Identifying Malicious IPs

How can we take the first-mover advantage away from hackers? We argue that hacker forums provide information earlier than other sources, and we should leverage these forums in our security intelligence. Here, we focus on a specific question. In particular, we want to extract as much useful information from hacker/security forums as possible in order to perform (possibly early) detection of malicious IP addresses, e.g., prior to their appearance on blacklists. The latter can exhibit large delays in their update and hence, new ways for labeling malicious IP addresses are needed [45]. In this study we will use the term "hacker forums" to describe online forums with a focus on security and system administration. Interestingly, we can classify these forums into categories: (a) main stream forums, like WildersSecurity, and (b) "fringe" forums, like OffensiveCommunity, where we find users with names like *satan911*. Some of these forums have been known to have hackers boast of attacks they have mounted, or sell tools for malicious purposes (think rent-a-botnet). For example, in our dataset there is a post that mentions "I give you a second server to have your fun with. Multiple websites on this server. So let's see if anyone can actually bring down the server". Right after that the hacker posted the IP, username and password for anyone to access the server. In fact, there is a *show-off* section in these forums for people to broadcast their hacking "skills".

The overarching goal of this work is to mine the unstructured, user-generated content in security forums. Specifically, we focus here on collecting malicious IP addresses, which are often reported at such forums. We use the term security forum to refer to discussion forum with a focus on security, system administration, and or more generally, systems-related discussions. The users in these forums include: security professionals, hobbyists, and hackers, who go on these forums to identify issues, discuss solutions, and in general exchange information.

Let us provide a few examples of how users report IP addresses, which may or may not be malicious. Posts could talk about a benign IP address, say in configuration files, as in the post:

"[T]his thing in my hosts file: 64.91.255.87 ... [is] it correct?".

At the same time, posts could also report compromised or malicious IP addresses, as in the post:

"My browser homepage has been hijacked to http://69.50.191.51/2484/".

The challenge is to automatically distinguish between the two. By doing so, we can provide a new source of information of malicious IP addresses directly from the affected individuals. Formally, we can state the problem as follows:

Key Question: Malicious IP Detection. Given a set of posts $\mathcal{P}_F$ that may contain IP addresses and users $\mathcal{U}_F$ of a security forum F, as well as, the features Φ_p, $\forall p \in \mathcal{P}_F$ and Φ_u, $\forall u \in \mathcal{U}_F$ for the posts and the users respectively, can we determine if a given IP address i is malicious or not?

The set of features $\mathcal{P}_F$ includes attributes such as the text of the post, the posting user, the time of post, etc., while $\mathcal{U}_F$ includes information such as the date of a user joining the forum, the number of posts the user has made etc. The above problem has two associated questions:

a. Exclusivity: How many IP Addresses can we find that are never reported by other reference sources?

b. Early warning: How much earlier are malicious IP Addresses reported in a forum compared to reference sources, for the IP Addresses reported by both?

Table 2.1: Extracting useful information; Number of malicious IP Addresses found by InferIP and not by VirusTotal.

		IP found by	
Dataset	Total IP	Virus Total	InferIP only
WildersSecurity	4338	216	**670**
OffensiveCommunity	7850	339	**617**
Ashiyane	8121	133	**806**

Most previous studies in this area have focused on structured information sources, such as security reports, or malware databases. In fact, many efforts focus on addressing security problems using knowledge obtained from the web, as well as social and information networks. These efforts are mainly focused on analyzing structured sources (e.g., [48]). However, studies assessing the usefulness of (unstructured) information in online forums have only recently emerged (e.g., [80]). These studies are rather exploratory and provide evidence of the usefulness of the data in the forums, but do not provide a systematic methodology or ready-to-use tools, which is the goal of our work. We discuss existing literature in more detail later in section 2.4.

The motivation of our work is to provide more information to security analysts and systems. We want to enhance and complement, but not replace, existing efforts for detecting malicious IP Addresses. For instance, many IP blacklists enlist an IP as malicious after a number of reports above a pre-defined threshold have been made for the specific address. Depending on the threshold and the reactivity of the affected users/systems, this might take several days, weeks or months. Therefore, a system, like the one proposed here, can identify and point to malicious IP address to blacklist services and firewalls.

We propose InferIP, a systematic approach for identifying malicious IP Addresses among the IP addresses, which are mentioned in security forums. A key novelty is that we use the behavioral information of the users, in addition to the textual information from the related posts. Specifically, we customize and use a Sparse Matrix Regression method on this expanded set of features.

This paper presents an extension of our previous work [42]. Here, we add some spatiotempral and behavioral analysis to extract the characteristics of the identified IP addresses and the users who used these IP address in their posts. Moreover, we investigate the ability of the proposed method to provide early warning regarding malicious IP addresses.

By design, our framework is applicable to forums in different languages as it relies only on the behavioral patterns of users and simple word counts, and not a complex language-specific Natural Language Processing technique. From a technical point of view the challenge in designing a solution to our Key Question is most IP Addresses mentioned in these forums are not malicious. We show that our system can add a significant number of previously unreported IP address to existing blacklist services. Finally, as an engineering contribution, we develop a customizable tool to facilitate the crawling of forums, which we discuss in the next section.

Our results can be summarized into the following points:

a. Our method exhibits precision and recall greater than 88% and 85% respectively, and an accuracy over *malicious* class above 86% in the 10-fold cross validation tests we conducted for the three different forums. In partially answering our Key Question, if our method labels a currently non-blacklisted IP as malicious, there is a high chance that it is malicious, given our high precision.

b. Our method identifies three times more malicious IP Addresses compared to VirusTotal [5] a widely used aggregator of 60 blacklists of IP addresses. Across our three forums, we find more than 2000 potential malicious IP Addresses that were never reported by VirusTotal.

c. Our method identifies more than half of the IP addresses at least 3 month earlier than VirusTotal. We study the malicious IP addresses that are identified by both VirusTotal and InferIP. We find 53%, 71% and 62% of these IP addreses in WildersSecurity, OffensiveCommunity and Ashiyane respectively at least 3 months earlier than they were reported in VirusTotal.

d. The number of reported malicious IP addresses has increased by a factor 8 in 4 years. We find that the number of malicious IP addresses has increased from roughly 100 in 2011 and 2012 to more than 800 in 2016. This could be attributed to either an increase in the user base, an increase in the number of attacks, or a combination of the two.

2.1 Data Collection and Basic Properties

We have collected data from three different forums relevant to our study; (i) WildersSecurity [6], (ii) OffensiveCommunity [4], (iii) Ashiyane [1]. The first two forums are mainly written in English, while the last forum is an Iranian forum, in Farsi[1].

Our data collection tool. We develop a customizable universal tool to make the crawling forums easier. The challenge here is that each forum has its own format and layout. Our tool requires only a custom configuration file, before crawling a new forum. In configuration file, we specify entities in the forum which are needed such as user ID, post's date, post's content and etc by XML Path Language known as *Xpath*. Leveraging our current configuration files, the task of crawling a new forum is simplified significantly. Using our crawler, we collect data from three forums, two English and one in Farsi for a total number of more than 30K users and 600K posts.

We use VirusTotal [5] as our reference blacklist IP addresses, since it is an aggregator, and combines the information from over 70 other blacklists and resources. VirusTotal is free to end users for non-commercial use and is a private API to query the services in the rate of more than 4000 IP addresses per minutes. It is provided upon requests for academic purposes.

[1] Our software and datasets will be made available at: https://github.com/hackerchater/

We provide some basic statistics for our three forums in Table 2.2. OffensiveCommunity and Ashiyane are two fringe forums in different languages. In these forums there is a section where people openly boast about their achievement in hacking. They share their ideas and *tutorials* on how to break into vulnerable networks. On the other hand, WilderssSecurity as a mainstream forum is mostly used to protect non-experts against attacks such as browser hijacking, and provide solutions for their security problems.

For completeness, we present some of the terms we use here. A user is defined by a login name registered with the site. The term post refers to a single unit of content generated by a user. A thread refers to a collection of posts that are replies to a given initiating post.

In Figures 2.1 and 2.2, we present the cumulative complementary distribution function of the number of posts per user and the number of threads per users respectively. As we can see in all the cases the distributions are skewed, that is, most of the users contribute few posts in the forums and engage with few threads. In WildersSecurity, 85% of users post less than 10 posts each, while 5.2% of the users post more than 50 posts. We find that 70% of the users post in only one thread and only 8% of the users are active in more than 10 threads. This skewed behavior is typical for online users and communities [31]. We will use features to capture aspects of both these user properties, as we will see in the next section.

In Figure 2.3, we present the cumulative complementary distribution function of the number of IP addresses that appear in each post. The skewed distribution shows that most of the posts contain a few number of IP address. We find that 84.2% of the

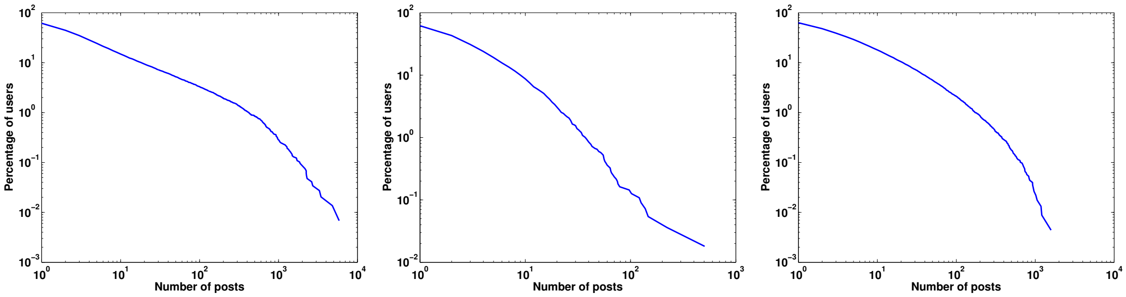
Table 2.2: The collected forums.

Forum	Threads	Posts	Users	Active days
WildersSecurity	28661	302710	14836	5227
OffensiveComm.	3542	25538	5549	1508
Ashiyane	67004	279309	22698	4978

posts with IP addresses in WildersSecurity and 84.1% in OffensiveCommunity have two or less IP addresses. In Ashiyane, 87.2% of these posts contain less than two IP addresses. Interestingly, in Ashiyane, we find 1% of the IP containing posts with more than 100 IP addresses. We investigated and we found that typically, these posts provide benign IP addresses of proxies servers to fellow administrators.

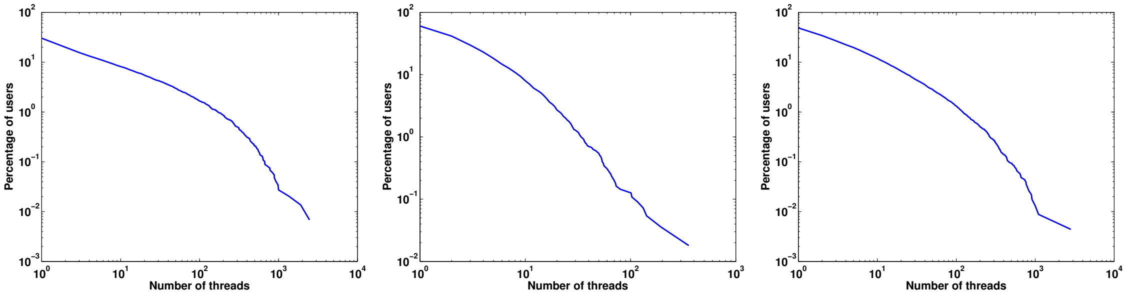
Groundtruth for training and testing. In order to build and evaluate, our model we need to obtain a reasonably labeled dataset from IP addresses that appear in the posts of the security forums. For that, we use the VirusTotal service and assign malicious labels to an IP that has been reported by this service. The number of malicious IP Addresses that we have used with the corresponding posts are shown in table 2.1 as the IP found by VirusTotal. Note that the absence of a report on VirusTotal does not necessarily mean that the IP is benign. However, a listed IP address is most likely malicious, since VirusTotal as most blacklist sites require a high threshold of confidence for blacklisting an address. This way, we find in total 688 malicious IP addresses for our forums as shown in Table 2.1.

Using this labeling process we have collected all the IP addresses that have appeared on our forums prior to their report on VirusTotal. For building our model, we also randomly select an equal number of IP addresses that have not been reported as malicious and via manual inspection further assess their status. Finally, for every security forum we have a different dataset and hence, we build a different model.



(a) WildersSecurity.
(b) OffensiveCommunity.
(c) Ashiyane

Figure 2.1: CCDF of the number of posts per user (log-log scale).

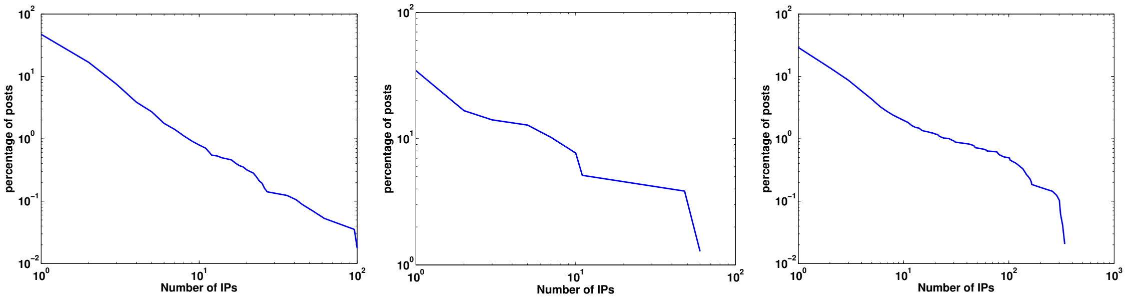


(a) WildersSecurity
(b) OffensiveCommunity
(c) Ashiyane

Figure 2.2: CCDF of the number of thread per user (log-log scale).

2.2 InferIP: Malicious IP Detection

We propose a method to identify whether an IP address within a post is malicious. For example, although many users report a malicious IP address, such as one that is attacking the user's network, there are also users that will mention a benign IP address when



(a) WildersSecurity

(b) OffensiveCommunity

(c) Ashiyane

Figure 2.3: CCDF of the number of IP addresses per post (log-log scale).

people discuss about network tutorials like setting up *Putty* or initiating a *SSH* connection.

While this task is simple for a human, it is non-trivial to automate. Adding to the challenge, different communities use different terminology and even different languages altogether (english and farsi in our case). In order to overcome these challenges, we use a diverse set of features and build a model to identify IP addresses that are potentially malicious.

Our approach consists of four steps that each hide non-trivial novelties:

Step 1: We consider the user behavior and extract features that profile users that post IP-reporting posts.

Step 2: We extract keywords from the posts and use information gain to identify the 100 most informative features.

Step 3: We identify meaningful *latent feature sets* using an unsupervised co-clustering approach [72].

Step 4: We train a classifier using these *latent feature sets* using 10-fold cross validation.

We describe each step in more detail.

Step 1: Behavioral Features. We associate each user of the forum with a set of 11 features that capture their behavior. In particular:

- Number of posts; the total number of posts made by the user
- Number of threads; the total number of threads the user has contributed to
- Number of threads initiated; the total number of threads initiated by the user
- Average thread entropy; the average entropy of the user distribution of the threads in which the user has contributed to
- Number of active days; the number of days that the user generates at least one post
- Average day entropy; the average entropy of the user distribution of the posts made on the days that the user is active
- Active lifetime; the number of days between the first and the last post of the user
- Wait time; the number of days passed between the day the user joined the forum and the day the user contributed their first post
- Average post length; the average number of characters in the user's posts
- Median post length; the median number of characters in the user's posts
- Maximum post length; the number of character's in the user's longest post

Step 2: Contextual Features. Apart from the aforementioned behavioral features we also include features related with the context in which an IP address appears

Table 2.3: Selecting a classifier: overall accuracy.

Forum	Naive Bayes	3NN	Logistic regression
WildersSecurity	91.9%	87.1 %	94.8%
OffensiveComm.	84.1%	83.2%	86.5%
Ashiyane	85.1%	82.3%	94%

Table 2.4: InferIP evaluation: 10-fold cross validation evaluation (using Logistic Regression).

Forum	Instances	Precision	Recall	ROC Area
WildersSecurity	362	0.9	0.94	0.96
OffensiveComm.	342	0.88	0.85	0.91
Ashiyane	446	0.9	0.92	0.92

within a post. In particular, we consider the frequency of the words (except stop-words) in the posts. Words that are frequent only in few documents (posts in our case) are more informative than those that appear frequently on a larger corpus [76]. To this end, we use TF-IDF to weight the various words/terms that appear in our data. After calculating the frequency and the corresponding weights of each word in the dataset we end up with more than 10,000 features/terms. Hence, in the next step we select discriminative features by extracting latent features.

We begin by performing feature selection in order to identify the most informative features by applying the information gain framework [106]. Furthermore, in order to avoid overfitting we pick a random subset of posts from the whole dataset and select the highest ranked features based on *Information Gain* score. In this way, a subset of discriminative keywords, 100 in our model, are selected. It turns out that each user uses only a small number of those words, resulting in a sparse dataset which we wish to exploit in our model.

Step 3: Identifying latent feature sets. We also like to leverage latent similarities of different posts in some of the dimensions spanned by post features and behavioral features for the writer of the post. Essentially, we seek to identify groups of highly similar posts under a small number of features, which does not necessarily span the full set of features. The reason why we wish to pinpoint a subset of the features instead of the entire set is because this way we are able to detect subtle patterns that may go undetected if we require post similarity across all the features. We call those sets of feastures *latent feature sets* . To this end, we apply a soft co-clustering method, Sparse Matrix Regression (SMR) [72], to exploit the sparsity and extract latent features of the post containing IP addresses. Given a matrix $\mathbf{X}$ of posts × features, its soft co-clustering via SMR can be posed as the following optimization problem:

$$\min_{\mathbf{a}_r \geq 0, \mathbf{b}_r \geq 0} \|\mathbf{X} - \sum_r^R \mathbf{a}_r \mathbf{b}_r^T\|_F^2 + \lambda \sum_{i,r} |\mathbf{a}_r(i)| + \lambda \sum_{j,r} |\mathbf{b}_r(j)|$$

where $\mathbf{a}_r$ and $\mathbf{b}_r$ are vectors that "describe" co-cluster r, which we explain below. Each $\mathbf{a}_r$ is a vector with as many dimensions as posts. Each value $\mathbf{a}_r(i)$ expresses whether post i is affiliated with co-cluster r. Similarly, $\mathbf{b}_r$ is a vector with as many dimensions as features, and $\mathbf{b}_r(j)$ expresses whether feature j is affiliated with with co-cluster r. Parameter λ controls how sparse the co-cluster assignments are, effectively controlling the co-cluster size. As we increase λ we get sparser results, hence cleaner co-clustering assignments. We tune λ via trial-and-error so that we obtain clean but non-empty co-clusters, and we select $\lambda = 0.01$ in our case.

Step 4: Training the model. We subsequently train a number of classifiers using the selected features based on **a** matrix. In particular, we examine (a) a Naive Bayes classifier, (b) a K-Nearest Neighbor classifier and (c) a logistic regression classifier. Our 10-fold cross validation indicates that the Logistic regression classifier outperforms kNN and Naive Bayse, achieving high accuracy, precision and recall (see Table 2.3).

Determining feature sets. We investigate the effect of selecting different feature sets in classifying IP addresses in forums. To this end, we investigate three subsets of the features discussed earlier.

a. Words-Frequency is the normalized frequency of the most informative words that appear in a post as discussed in Step 2.

b. Combined is the set of features which consists of the combination of the words frequency features, defined above, and user behaviour features, which are extracted in Step 1. In other words, it is the union of the features in Step 1 and Step 2.

c. Co-Clustered is the latent set of features extracted in Step 3 by applying the co-clustering approach on the Combined features set.

We evaluate these three sets of features on their ability to enable the classification. In more detail, we use these features with a classifier to assess their effectiveness by computing the accuracy of the classifier to identify malicious IP addresses. According to the results which are shown in Figure 2.4, the Co-Clustered features set exhibits higher accuracy by 4.1% compared to Words-Frequency. On the other hand, although the Combined features do not increase the accuracy compared to the Words-Frequency, the co-clustering method does. It extracts the latent features from the Combined features set and outperforms Words-Frequency and Combined in identifying malicious IP addresses.

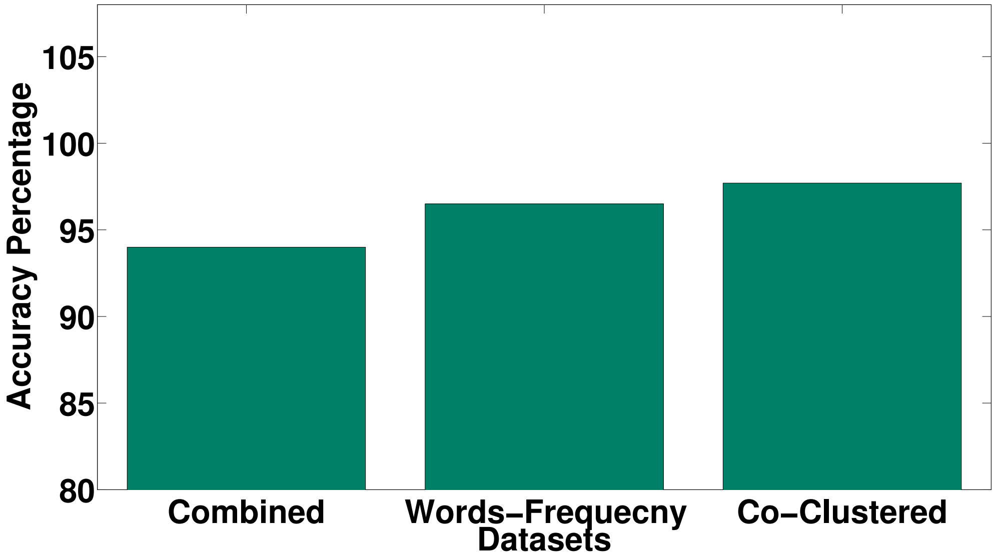


Figure 2.4: Accuracy of different feature sets in WildersSecurity forum to detect malicious IP

2.2.1 Applying InferIP on the forums

Having established the statistical confidence of our classifier, we apply it on the posts of the forums except the ones that we used in our groundtruth. We use the logistic regression classifier as it exhibits the best performance.

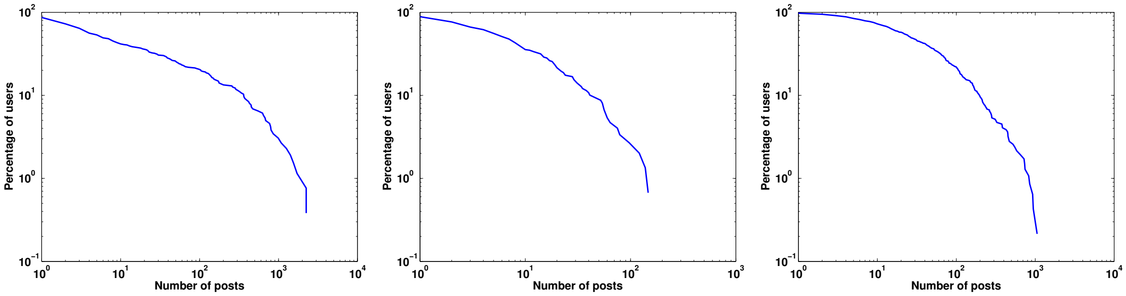
Applying InferIP on the forums shows that there is a wealth of information that we can extract from security forums in two aspects of the quantity and time of detecting malicious IP against VirusTotal.

a. Detecting more IP addresses. With InferIP, we find an additional 670 malicious IP addresses in WildersSecurity, and 617 in OffensiveCommunity 806 in Ashiyane (see Table 2.1). In other words, InferIP enables us to find three times additional malicious IP addresses in total compared to the IP addresses found on VirusTotal. It is interesting to observe that this factor varies among our three sites. For Ashiyane, our method finds roughly 6 times additional malicious IP addresses. With a precision of roughly around 90% and considering small amount of *False Positive* rate, our method can add a significant number of malicious IP addresses to a blacklist. Using the limited manual inspection, we confirm that the precision of the method on out of sample data is in the order of 88%.

b. Detecting malicious IP addresses earlier: more than half IPs, at least 3 months earlier Here we focus on the malicious IP addresses that are jointly identified by our method and VirusTotal and compare the time that they were reported in each source, and show the results in Table 2.5 for 3, 6 and 12 months difference in time. We compare jointly detected IP addresses with InferIP and VirusTotal in terms of time that the IP addresses were mentioned in posts and the time they were reported on VirusTotal. We see that on average 62% of the malicious IP addresses with InferIP could be identified at least 3 months earlier than VirusTotal. We can see that with InferIP, we find 53%, 71% and 62% of these IP addresses in WildersSecurity, OffensiveCommunity and Ashiyane respectively at least 3 months earlier than in VirusTotal. We also identify 39% and 24% of the malicious IP addresses respectively at least 6 and 12 months earlier with InferIP.

Additional stress-testing of our accuracy: In order to assess the performance of our approach, we randomly picked 10 percent of the labeled data with InferIP method and annotated them manually by human annotators. The calculated accuracy on the sampled data shows more than 85% accuracy on average over all datasets which is close but somewhat lower than the reported accuracy in the Table 2.3.

Contributing Users. Who are the users that report malicious IP addresses? We want to understand and ideally, develop a profile for these users, which we will refer to as



(a) WildersSecurity (b) OffensiveCommunity (c) Ashiyane

Figure 2.5: CCDF of the number of overall posts per Contributing users (who report malicious IPs) in log-log scale.

Table 2.5: Timely comparison between jointly detected malicious IP addresses in InferIP and VirusTotal. Reported percentage of malicious IP Addresses which InferIP detected earlier than VirusTotal

	At least X months earlier		
Dataset	3	6	12
WildersSecurity	53%	23%	14%
OffensiveCommunity	71%	46%	21%
Ashiyane	62%	49%	37%
Average (across forums)	62%	39%	24%

Contributing users. We start by considering the number of post these users post on the forums.

The majority of IP reporting is done by highly active (more than 10 posts overall) in Ashiyane. In Figure 2.5, we show the cumulative complementary distribution function for the number of posts per Contributing user for Ashiyane. More than 72% of the Contributing users post more than 10 posts overall, which we consider as high engagement given the distribution of posting that we saw in the previous section. Therefore, in Ashiyane, Contributing users are contributing significantly in reporting malicious IP addresses. Intrigued, we examined further and found that, among them, there are two users who have more than 1000 posts, 1058 and 2780 to be exact, and whose user-names are *"Classic"* and *"Crisis"*. On the other side of the spectrum, 2.4% of Contributing users have posted a single post in the forum, and in that post they reported a malicious IP address.

The majority of IP reporting is done by less active users (less than 10 posts overall) in OffensiveCommunity. In Figure 2.5, we show the cumulative complementary distribution function for the number of posts per Contributing user for OffensiveCommunity. Unlike Ashiyane, here 65% of the Contributing users have less than 10 posts overall. Going into more detail, roughly 12% of the Contributing users have a single post overall, while 26% of them have only two overall posts. The same behavior is observed in WildersSecurity which is shown in Figure 2.5.

Overall, there does not seem to be an obvious pattern between number of total posts and number of malicious IPs reported among Contributing users.

2.2.2 Case-study: from reported malicious IPs to a DDoS attack

We show that mining the forums could actually provide information about real events. We identify a link between a malicious IP address that our method detected with an actual DDoS attack.

We conducted the following analysis. We plot the time-series of the number of posts containing malicious IP addresses in WildersSecurity from 2012 to 2013 found by InferIP. We show the time-series in Figure 2.6. We observe some spikes on these time-series, which we further analyze. One of the spikes was in September 2012, and it reports a set of malicious IP addresses that were involved in an DDoS attack that month. That same thread continued being active, and in December of 2012, it was reported in that thread that attack was caused by *Nitol Botnet* due to a Microsoft's vulnerability [3].

We argue that this case-study points to additional layers of functionality that can be built upon our method, that can provide a semi-automated way to extract richer information beyond just reporting malicious IP addresses.

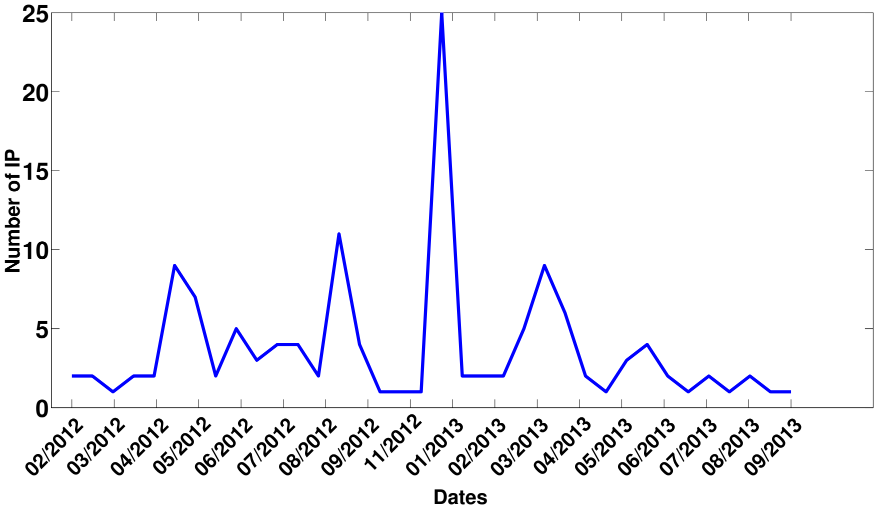


Figure 2.6: Time-series of the number of posts containing malicious IP reported in each month for WildersSecurity.

2.2.3 Discussion and limitations

Although our method exhibits pretty good accuracy overall, we attempt to understand its limitations and detect the source of misclassifications.

Limited text in the post: The words in the post provide significant evidence for the classification. In some cases, some posts are very sparse in their text, which makes the classification of the included IP address harder. We consider these kinds of posts a significant contributor to misclassifications.

Characterization at the post level: In our method, we classify an IP address by using features at the level of a post. Recall that roughly 86% of all posts across all forums has a single IP per post as shown in Figure 2.3. In other words, having more than one IP address per post is already not very common. Furthermore, even more rarely, we have seen a few cases, where a post contains both a benign and a malicious IP address. As our method is currently set-up, this will lead to errors in the classification. A straightforward solution is to consider examining the text surrounding each IP address within the post.

2.3 SpatioTemporal Analysis

In this section, we discuss the spatiotemporal features of the malicious IP addresses identified in security forums in Section 2.2.

2.3.1 Temporal analysis

The key question from a temporal point of view is if the number of reported malicious IP addresses increases or decreases over time.

The number of reported malicious IP addresses has increased by a factor 8 in 4 four years. In Figure 2.7, we plot the number of reported malicious IPs found by our method across all three forums between 2011-2016. We find that the number increased by a factor of 8: from roughly 100 to roughly 800. In spite of some decreases in years 2011, 2012 and 2015, it has a clear increasing trend.

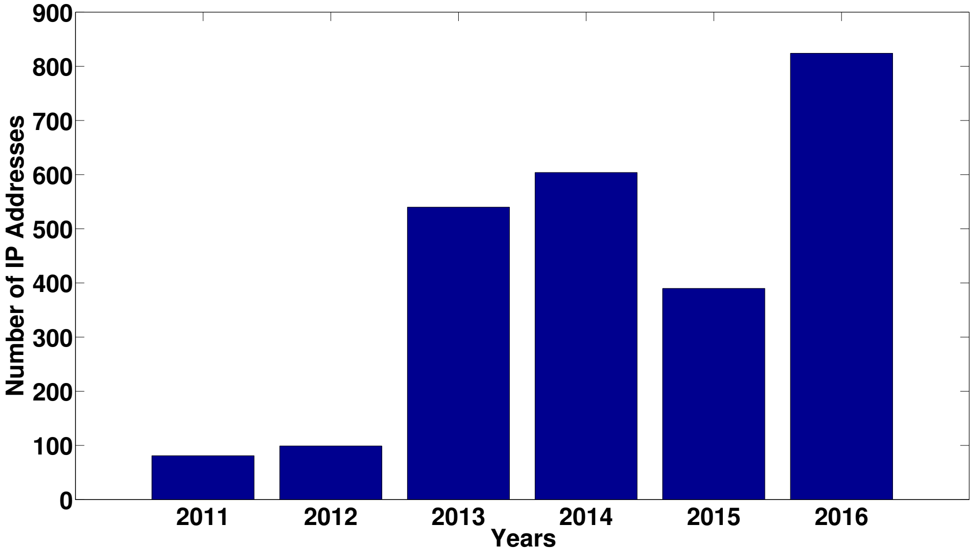


Figure 2.7: Increasing trend: Malicious IP addresses reported on the forums each year.

2.3.2 Spatial analysis

We study the geo-location of the identified IP addresses from Section 2.2. We utilize *GeoLite* database [2], which can show us the country and continent of an IP address. Here we focus on continents of the IP addresses location.

A natural question to ask is whether the geographical distribution of the malicious addresses differs between VirusTotal and InferIP. We investigate this in detail below.

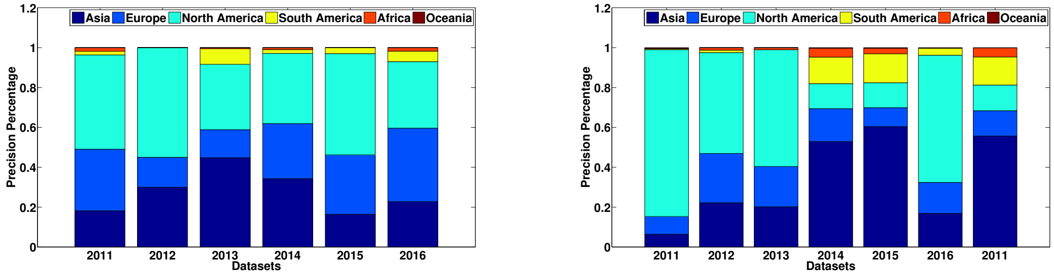
VirusTotal: North America hosts the majority of the reported malicious IP addresses. We plot the percentage of the distribution of the IP addresses extracted from VirusTotal across continents in Figure 2.8 (a) between 2011-2016. We observe that the majority of the malicious IP addresses are located in the North America continent. There are two exception in 2013 and 2016 when Asia and Europe respectively contain most of the

Table 2.6: Percentage of distribution of IP addresses across continents over all the years.

	North America	Asia	Europe	South America	Africa	Oceania
InferIP	**46.7**	32.5	13.5	5.2	1.6	0.5
VirusTotal	**50**	26.5	20.4	2.4	0.6	0.17

malicious IP addresses. Overall, Table 2.6 shows the geo-graphical distribution over all the years: North America, Asia and Europe are the three most active continents in that order.

InferIP: North America dominates again, but South America and Africa have non-trivial contributions. We plot the percentage of the distribution of the IP addresses extracted form InferIP across continents in Figure 2.8 (b) between 2011-2016. We observe that North America hosts the majority of the reported malicious IP addresses again, but we find a more diverse global activity compared to what we observed in VirusTotal. For example, we can see that in years 2013, 2014, and 2016: (a) Asia has the majority of the malicious IP addresses, and (b) South America and Africa have a considerable percentage of malicious IP addresses. However, when seen across all years, the geographical distributions of the IPs in InferIP and VirusTotal quite similar: North America, Asia and Europe have the majority of the malicious IPs detected by InferIP similarly to those of VirusTotal. In Figure 2.9, we plot the geographical distribution of malicious IPs per continent across all years and all forums for InferIP and VirusTotal, while the exact numbers are shown in Table 2.6. Qualitatively the distributions look relatively similar, especially in the order of significance of the continents, but at the same, we can see that South America and Africa have a larger percentage of IP addresses in InferIP compared to those in VirusTotal.



(a) VirusTotal

(b)InferIP

Figure 2.8: SpatioTemporal distribution of malicious IP addresses detected by InferIP and VT .

2.4 Related Work

We briefly discuss three categories of relevant research.

a. Analyzing structured security sources. There is a long line of research studying the ontology of cyber security and the automatic extraction of information from structured security documents. Iannacone *et al.*[48] developed a schema for extracting relevant concepts from various types of structured data sources. In another work, Blanco *et al.* [17] proposed methods to detect anomalies on the extracted ontology and network flow graph. Moreover, Bridges *et al.*[20] proposed a method to do entity labeling on structured data by utilizing neural networks. These work are complementary to ours as we focus on unstructured data, which poses different challenges.

b. Analyzing online security forums. Recently security forums have been the focus of various studies that showcase the usefulness of the information present in security forums. For example, Motoyama *et al.* [67] present a comprehensive statistical analysis in underground forums. Others studies focus on the users' classification or the discovery of the relationships between the forum's members [110, 7]. Extracting different discussion topic

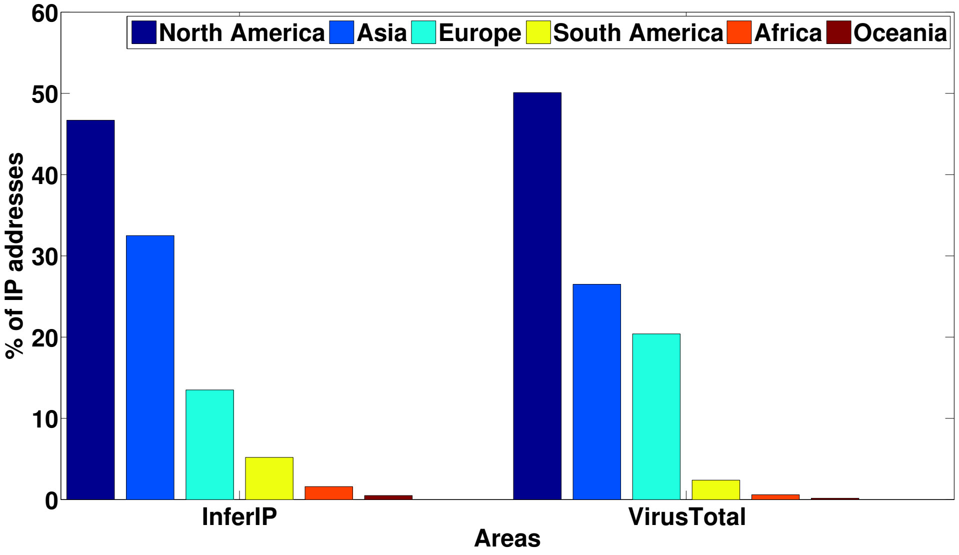


Figure 2.9: The percentage distribution of malicious IP addresses in each continent across all three forums for InferIP and VirusTotal.

in the forums and classifying the language of the codes posted in the forum has been done in [80]. Contrary to these studies, our work emphasizes on the development of automated systems that actually exploit the wealth of information in these security forums in order to enhance security. Similar to detecting malicious users on commenting platforms has been done on [60]. A recent work analyzes security forums to identify and geo-locate Canadian IP addresses focusing on spam and phishing [39] and in another work, Portnoff *et al.* [74] studies the exchange of malicious services and tools and studies their prices on the security forums.

c. Analyzing blogs and social networks. There has been a plethora of studies on blogs and social media, but their goals are typical not related to extracting security information. [23, 11, 98]. The studies range from modeling user behavior [31, 77] to inferring information about the user (demographics, preferences, mental state), and to modeling the information propagation on online forums. Although interesting, the focus of these studies are significantly different from our goal here.

2.5 Conclusion

The take away message from our work is that there seems to be a wealth of useful information in security forums. The challenge is that the information is unstructured and we need novel methods to extract it. In this direction, a key insight of our work is that using behavioral and text-based features can provide promising results.

In support of this assertion, we develop a systematic method to extract malicious IP addresses reported in security forums. We utilize both behavioral, as well as textual features and show that we can detect malicious IP addresses with high accuracy, precision and recall. Our results in Table 2.1 are promising.

We then apply InferIP to all the posts we have collected. Although are classification is not perfect, our relatively high precision (hovering around 90% in Table 2.4) provides sufficient confidence in our results. We find three times as many additional malicious IP addresses as the original malicious IP addresses identified by VirusTotal. Furthermore, even for the jointly discovered IP addresses, at least 53% of the IP addresses detected at least 3 months earlier than VirusTotal. The key message from our spatiotemporal analysis is that the number of reported malicious IP addresses is increasing over time.

In the future, we plan to extend our work by extracting other types of security information. Our first goal is to detect malicious URLs mentioned in the forums. Our second and more ambitious goal is to identify the emergence of new malware, threats, and possibly attacks, which we expect to see associated with large numbers of panic-filled or help-requesting posts. Our final goal is to identify malicious users, since interestingly, some users seem to be promoting and selling hacking tools in these forums.

Chapter 3

Cross-Seeding to Identify Malicious Reported IPs

The overarching goal of this work is to harness the user generated content in forums, especially security forums. More specifically, we focus here on collecting malicious IP addresses, which are often reported at such forums. We use the term security forums to refer to discussion forums with a focus on security, system administration, and in general systems-related discussions. In these forums, security professionals, hobbyists, and hackers identify issues, discuss solutions, and in general exchange information.

We provide a few examples of the types of discussions that take place in these forums that could involve IP addresses, which is our focus. Posts could talk about a benign IP address, say in configuration files, as in the post: *"[T]his thing in my hosts file: 64.91.255.87 ... [is] it correct?"*. At the same time, posts could also report compromised or malicious IP addresses, as in the post: *"My browser homepage has been hijacked to*

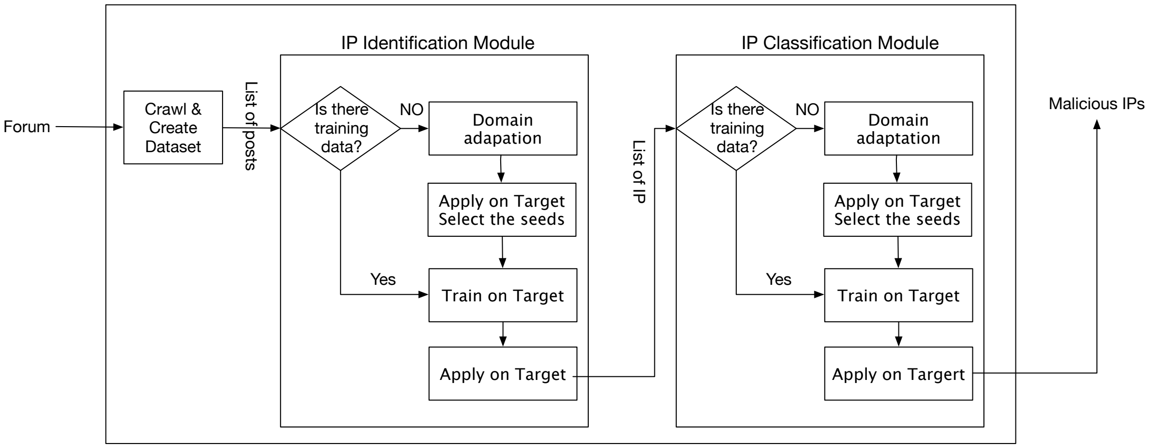


Figure 3.1: The overview of key modules of our approach (RIPEx): (a) collecting data, (b) IP Identification, and (c) IP Characterization. In both classification stages, we use our Cross-Seeding approach that in order to generate seed information for training a classifier for a new forum.

http://69.50.191.51/2484/". Our goal is to automatically distinguish between the two and provide a new source of information for malicious IP addresses directly from the affected individuals.

The problem that we address here is to find all the IP addresses that are being reported as malicious in a forum. In other words, the input is all the posts in a forum and the expected output is a list of malicious IP addresses.

As with any classification problem, one would like to achieve both high precision and recall. Precision represents the percentage of the correctly labeled over all addresses labeled malicious. Recall is the percentage of malicious addresses that we find among all malicious addresses reported in forums. It turns out that this is a two-step problem. First, we need to solve the **IP Identification** problem: distinguishing IP addresses from other numerical entities, such as a software version. Second, we need to solve the **IP Characterization** problem: characterizing IP address as malicious or benign. The extent of the Identification problem caught us by surprise: we find 1820 non-address dot-decimals, as we show in table 3.1.

There is limited work on extracting information from security forums, and even less work on extracting malicious IP addresses. We can group prior work in the following categories. First, recent works study the number of malicious IP addresses in forums, but without providing the comprehensive and systematic solution that we propose here [39]. Second, there are recent efforts that extract other types of information from security forums, related to the black market of hacking services and tools [74], or the behavior and roles of their users [47, 7]. Third, other works focus on analyzing structured sources, such as security reports and vulnerability databases [19, 52]. We discuss related work in section 3.4.

There is a wealth of information that can be extracted from security forums, which motivates this research direction. Earlier work suggests that there is close to four times more malicious IP addresses in forums compared to established databases of such IP addresses [42]. At the same time, there are tens of thousands of IP addresses in the forums, as we will see later. Interestingly, not all of the reported IP addresses are malicious, which makes the classification necessary.

We propose RIPEx[1], a comprehensive, automated solution that can detect malicious IP addresses reported in security forums. As its key novelty, our approach minimizes the need for human intervention. First, once initialized with a small number of security forums, it does not require additional training data to mine new forums. Second, it addresses both the Identification and Characterization problems. Third, our approach is systematic and readily deployable. We are not aware of prior work claiming these three properties, as

[1] RIPEx stands for **R**iverside's **IP** **Ex**tractor.

we discuss in section 3.4. The overview of our approach is shown in figure 3.1.

The key technical novelty is that we propose **Cross-Seeding**, a method to conduct a multi-step knowledge transfer across forums. We use this approach for both classification problems, when we have no training data for a new forum. With Cross-Seeding, we create training data for the new forum in the process depicted in figure 3.1. We use a classifier based on the current forums to identify seed information in the new forum. We then use this seed information to train a classifier for the new forum. This forum-specific classifier performs much better than if we have used the classifier of the current forums on the new forum. We refer to this latter knowledge transfer approach as **Basic**.

We evaluate our approach using five security forums with a total of 31K users and 542K posts spanning a period of roughly six years. Our results can be summarized into the following points.

a. Identification: 98% precision with training data per forum. We develop a supervised learning algorithm for solving the Identification problem in the case where we have training data for the target forum. Our approach exhibits 98% precision and 96% recall on average across all our sites.

b. Identification: 95% precision with Cross-Seeding. We show that our Cross-Seeding approach is effective in transferring the knowledge between forums. Using the WildersSecurity forum as source, we observe an average of 95% precision and 93% recall in the other forums.

c. Characterization: 93% precision with training data per forum. We develop a supervised learning algorithm for solving the Characterization problem assuming we have training data for the target forum. Our classifier achieves 93% precision and 92% recall on average across our forums.

d. Characterization: 88% precision on average with Cross-Seeding data. We show that our Cross-Seeding approach by using OffensiveCommunity forum as source can provide 88% precision and 82% recall on average.

e. Cross-Seeding outperforms Basic. We show that Cross-Seeding is important, as it increases the precision by 28% and recall by 16% on average in the Characterization problem, and the precision by 8% and recall by 7% on average in the Identification problem.

f. Using more source forums improves the Cross-Seeding performance. We show that, by adding a second source forum, we can improve the precision by 13% on average over the remaining three forums.

Our work suggests that there is a wealth of information that we find in security forums and offers a systematic approach to do so.

3.1 Our Forums and Datasets

We have collected data from five different forums, which cover a wide spectrum of interests and intended audiences. We present basic statistics of our forums in Table 3.1 and we highlight the differences of their respective communities.

Our semi-automated crawling tool. We have developed an efficient and customizable python-based crawler, which can be used to crawl online forums, and it could be of independent interest. To crawl a new forum, our tool requires a configuration file that describes the structure of the forum. Leveraging our current configuration files, the task of crawling a new forum is simplified significantly. Due to space limitations, we do not provide further details. Following are the descriptions of collected forums.

- **WildersSecurity (WS)** seems to attract system administrator types and focuses on defensive security: how one can manage and protect one's system. Its topics include anti-virus software, best practices, and new vulnerabilities and its users seem professional and eloquent.

- **OffensiveCommunity (OC)** seems to be on the fringes of legality. As the name suggests, the forum focuses on breaking into systems: it provides step by step instructions, and advertises hacking tools and services.

- **HackThisSite (HT)** seems to be in between these extremes represented by the first two forums. For example, there discussions and competitions on hacking challenges, but it does not act as openly as a black market of illegal services and tools compared to OffensiveCommunity.

- **EthicalHackers (EH)** seems to consist mostly of "white hat" hackers, as its name suggests. The users discuss hacking techniques, but they seem to have a strict moral code.

- **Darkode (DK)** is a forum on the dark web that has been taken down by the FBI in July 2015. The site was a black market for malicious tools and services similar to OffensiveCommunity.

Our goal is to identify and report IP addresses that the forum readers report as

	WildersSec.	OffensiveComm.	HackThisSite	EthicalHackers	Darkode
Posts	302710	25538	84125	54176	75491
Threads	28661	3542	8504	8745	7563
Users	14836	5549	5904	2970	2400
Dot-decimal	4325	7850	1486	1591	1097
IP found	3891	6734	1231	1330	1082

Table 3.1: The basic statistics of our forums

malicious. We currently do not assess whether the author of the post is right, though the partial overlap with blacklisted IPs indicates so. We leave for future work to detect misguided reports of IP addresses.

Determining the ground-truth. For both of the problems we address here, there are no well-established benchmarks and labeled datasets. To train and validate our approach, we had to rely on external databases and some manual labelling. For the Identification problem, we could not find any external sources of information and benchmarks. To establish our ground-truth, we selected dot-decimal expressions uniformly randomly, and we used four different individuals for the labelling. To ensure testing fairness, we opted for balanced datasets, which led us to a corpus of 3200 labeled entries across all our forums.

For the Characterization problem, we make use of the VirusTotal site which maintains a database of malicious IP addresses by aggregating information from many other such databases. We also provide a second level of validation via manual inspection.

We create the ground truth by uniformly randomly selecting and assessing IP addresses from our forums. If VirusTotal and the manual inspection give it the same label, we add the addresses into our ground-truth. Finally, we again ensure that we create balanced sets for training and testing to ensure proper training and testing.

3.2 Overview of RIPEx

We represent the key components of our approach in addressing the Identification and Characterization problems. To avoid repetitions, we present at the end the Cross-Seeding approach, which we use in our solution to both problems.

3.2.1 The IP Identification module

We describe our proposed method to identify IP addresses in the forum.

The IP address format. The vast majority of IP addresses in the forums follow the *IPv4* dot-decimal format, which consists of 4 decimal numbers in the range [0-255] separated by dots. We can formally represent the dot-decimal notation as follows: *IPv4* $[x_1.x_2.x_3.x_4]$ with $x_i \in [0 - 225]$, for $i = 1, 2, 3, 4$. Note that the newer *IPv6* addresses consists of eight groups of four hexadecimal digits, and our algorithms could easily extend to this format as well. Interestingly, we found a negligible number of *IPv6* addresses, and we opted to not focus on *IPv6* addresses here. For example, in WildersSecurity forum, we find 3891 *IPv4* addresses and only 56 *IPv6* addresses. At such small numbers, it is difficult to train and test a classifier. Thus, for the rest of this paper, IP address refers to *IPv4* addresses.

The challenge: the dot-decimal format is not enough. If IP addresses were the only numerical expressions in the forums with this format, the Identification problem could have been easily solved with straightforward text processing and Named-Entity Recognition (NER) tools, such as the Stanford NER models [37]. However, there is a non-trivial number of other numerical expressions, which can be misclassified as addresses. For example, we quote a real post: *"factory reset brings me to the Clockworkmod 2.25.100.15 recovery menu"*. where the structure *2.25.100.15* refers to the version of Android app *"Clockworkmod"*.

To this end, we propose a method to solve the IP Identification problem, a supervised learning algorithm. We first identify the features of interest as we discuss below. We then train a classifier using the Logistic Regression method gives the best results among the several methods using 10-fold cross validation on our ground-truth as we decribed in the previous section.

Feature selection. We use three sets of features in our classification.

a. Contextual information: *TextInfo*. Inspired by how a human would determine the answer, we focus on the words surrounding the dot-decimal structure. For example, the words *"server"* or *"address"* suggests that the dot-decimal is an address, while the words *"version"* or a software name, like *"Firefox"* suggests the opposite. At the same time, we wanted to focus on words close to the dot-decimal structure. Therefore, we introduce **Word-Range, W,** to determine the number of surrounding words before and after the dot-decimal structure that we want to consider in our classification. We use TF-IDF [76] to normalize the frequency of a word to better estimate its discriminatory value.

b. The numerical values of the dot-decimal: *DecimalVal*. We use the numerical value of the four numbers in the the dot-decimal structure as features. The rationale is that non-addresses, such as software versions, tend to have lower numerical values. This insight was based on our close interaction with the data.

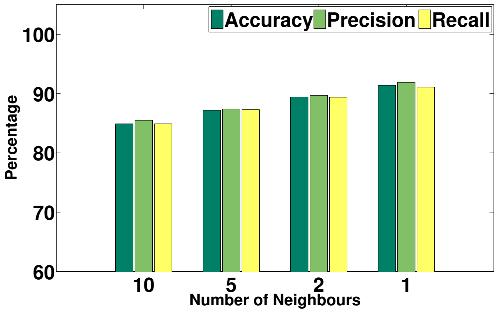


Figure 3.2: Classification performance versus the number of words Word-Range, W, in WildersSecurity.

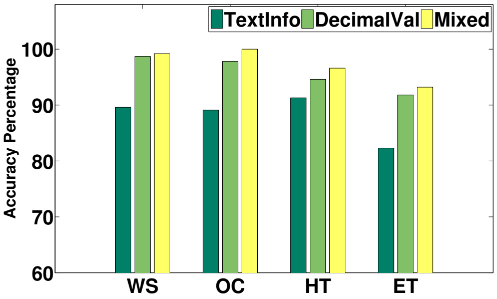


Figure 3.3: Classification accuracy for different features sets in 10-fold cross validation in four forums.

c. The combined set: *Mixed*. We combine the two feature sets to create in order to leverage their discriminating power.

Determining the right number of context words, Word-Range. We wanted to identify the best value of parameter Word-Range for our classification. In figure 3.2, we plot the classification accuracy, precision and recall, as we vary Word-Range, $W = 1, 2, 5$ *and* 10, for the WildersSecurity forum and using only the *TextInfo*. We see that using one to two words gives better results compared to using five and ten words. The explanation to this counter-intuitive result is that considering more words includes text that is not relevant for inferring the nature of a dot-decimal, which we verified manually.

Using numerical values *DecimalVal* improves the performance significantly. In Figure 3.3, we plot the classification accuracy of different features sets. Recall that we are not able to include Darkode forum due to its limited number of non-IP dot-decimal expressions, as we saw in 3.2.1. We see that using *DecimalVal* features alone, we can get 94% overall accuracy and using both *DecimalVal* and *TextInfo*, we get 98% overall accuracy across our forums. Focusing on the IP address class, we see a an average precision

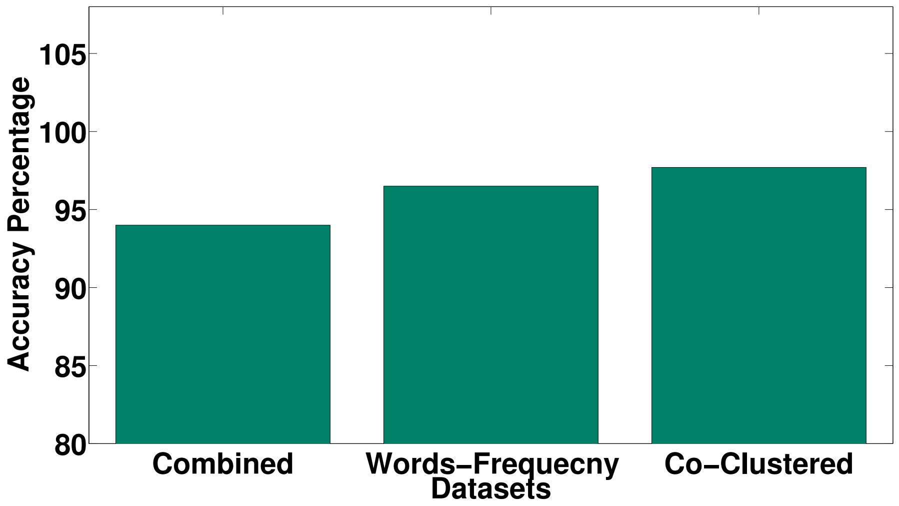


Figure 3.4: Characterization: The effect of the features set on the classification accuracy with balanced testing data.

of 95% using only *DecimalVal* and, 98% using both *DecimalVal* and *TextInfo*.

3.2.2 The IP Characterization module

We develop a supervised learning algorithm to characterize IP addresses. Here, we assume that we have labeled data, and we discuss how we handle the absence of ground truth in section 3.2.3. We first identify the appropriate set of features which we discuss below. We then train a classifier and find that the Logistic Regression method gives the best results among several methods that we evaluated. Due to space limitations, we show a subset of our results.

Features sets for the Characterization problem. We consider and evaluate three sets of features in our classification.

a. Text information of the post: *PostText*. We use the words and their frequency of appearance in the post. Here, we use the TF-IDF technique [76] again to better estimate the discriminatory value of a word by considering its overall frequency. In the future, we intend to experiment with sophisticated Natural Language Processing models for analyzing the intent of a post.

b. The Contextual Information set: *ContextInfo*. We consider an extended feature set that includes both the *PostText* features, but also features of the author of the post. These features capture the behaviour of the author, including frequency of posting, average post length etc. These features were introduced by earlier work [42], with the rationale that profiling the author of a post can help us infer their intention and role and thus, improve the classification.

Characterization: 93% precision with training data. We assess the performance of the Characterization classifier using the set of features above and by using the labeled data of each forum. We evaluate the performance using 10-fold cross validation. In figure 3.4, we show the accuracy of classification.

We can achieve 93% precision and 92% recall on average across all the forums. The results are shown in figure 3.4, where we report the results using the accuracy across both classes, given that we have balanced training datasets.

Selecting the *PostText* feature set. We see that, by using *PostText* features on their own, we obtain slightly better results. *PostText* feature achieves 94% accuracy on average, while using the *ContextInfo* results in 92% accuracy on average across all forums. Furthermore, text-based only features have one more key advantage: they can transfer between domains in a straightforward way. Therefore, we use the *PostText* features in the rest of the paper.

3.2.3 Transfer Learning with Cross-Seeding

In both classification problems, we face the following conundrum:

a. the classification efficiency is better when the classifier is trained with forum-specific ground-truth, but,

b. requiring ground-truth for a new forum will introduce manual intervention, which will limit the practical value of the approach.

We propose to do cross-forum learning by leveraging transfer learning approaches [29, 71]. We use the terms *source* and *target* domain to indicate the two forums with the target forum not having ground-truth available. For both classification problems, we consider two solutions for classifying the target forum:

a. Basic: We use the classifier from the source forum on the target forum.

b. Cross-Seeding: We propose an algorithm that will help us develop a new classifier for the target forum by using the old classifier to create training data as we explain below.

Our Cross-Seeding approach. We propose to create training data for the target forum following the four steps below, which are illustrated in figure 3.1 and outlined in algorithm 1.

a. Domain adaptation. The main role of this step is to ensure that the source classifier can be applied to the target forum. The main issue in our case is that the feature

Algorithm 1 Cross-Seeding: transfer learning between forums

```
CrossForum ($\mathcal{X}, \mathcal{Y}$) :
  Take the union of the features in forum $\mathcal{X}$ and $\mathcal{Y}$
  Apply classifier from $\mathcal{X}$ on $\mathcal{Y}$
  Select the high-confidence instances to create seed for $\mathcal{Y}$
  Train a new classifier on $\mathcal{Y}$ based on the new seeds.
  Apply the new classifier on $\mathcal{Y}$
```

sets can vary among forums. Recall that, for both classification problems, we use the frequency of words and these words can vary among forums. We adopt an established approach that works well for text classification [29]: we take the union of the feature sets of the source and target forums. The approach seems to work sufficiently well in our case, as we see later.

b. Creating seed information for the target forum. Having resolved any potential feature disparities, we can now apply the classifier from the source forum to the target forum. We create the seeding data by selecting instances of the target domain, for which the classification confidence is high. Most classification methods provide a measure of confidence for each classified instance and we revisit this issue in section 3.3.

c. Training a new classifier for the target forum. Having the seed information, this is now a straightforward step of training a classifier.

d. Applying the new classifier on the target forum. In this final step, we apply our newly-trained forum-specific classifier on the target forum.

3.3 Evaluation of our Approach

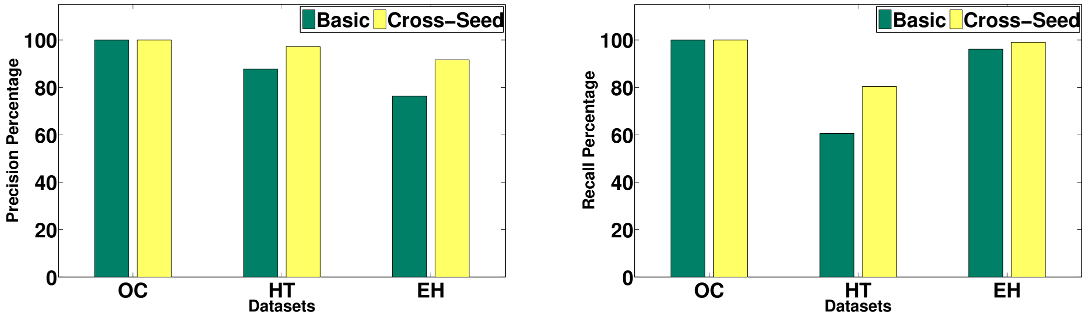
We evaluate our approach focusing on the performance of Cross-Seeding for both the Identification and the Characterization problems.

Our classifier. We use Logistic Regression as our classification engine, which performed better than several others, including SVM, Bayesian networks, and K-nearest-neighbors. In Cross-Seeding, we use the Logistic Regression's prediction probability with a threshold of 0.85 to strike a balance between sufficient confidence level and adequate number of instances above that threshold. We found this value to provide better performance than 0.8 and 0.9, which we also considered.

A. The IP Identification problem. As we saw in section 3.2.1, our classification approach exhibits 98% precision and 96% recall on average across all our sites, when we train with ground-truth for each forum.

a. Identification: 95% precision with Cross-Seeding. We show that our cross-training approach is effective in transferring the knowledge between domains. We use the classifier from WildersSecurity and we use it to classify three of the other forums, namely, OffensiveCommunity, EthicalHackers, and HackThisSite. Note that we do not include Darkode in this part of the evaluation as it did not have sufficient data for testing (less than 15 non-address expressions in all its posts).

In figure 3.5, we show the results for precision and recall of cross-training using Basic and Cross-Seeding. We see that Cross-Seeding improves *both* precision and recall significantly. For example, for HackThisSite, Cross-Seeding increases the precision from 57% to 79% and the recall from 60% to 78%.



(a)Precision (b)Recall

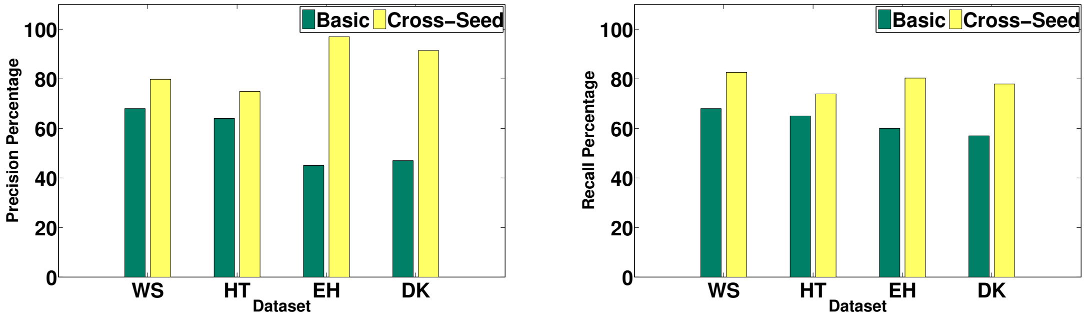
Figure 3.5: Identification: Cross-Seeding improves both Precision and Recall. Using WildersSecurity to classify OffensiveCommunity, HackThisSite, and EthicalHackers.

b. Identification: Cross-Seeding outperforms Basic. Cross-Seeding improves the precision by 8% and recall by 7% on average for the experiment shown in figure 3.5. The average precision increased from 88% to 95% and the average recall increased from 85% to 97%.

B. The IP Characterization problem. We evaluate our approach for solving the Characterization problem without per-forum training data. As we saw in section 3.2.2, we can achieve 93% precision and 92% recall on average across all the forums, when we train with ground-truth for each forum.

a. Characterization: 88% precision on average with Cross-Seeding. Using OffensiveCommunity as source, and we classify WildersSecurity, HackThisSite, EthicalHackers and Darkode as shown in figure 3.6. Our Cross-Seeding approach can provide 88% precision and 82% recall on average.

b. Characterization: Cross-Seeding outperforms Basic. We show that Cross-Seeding improves the classification compared to just reusing the classifier from an-



(a)Precision (b)Recall

Figure 3.6: Characterization: Cross-Seeding improves both Precision and Recall. Using OffensiveCommunity as source, we classify WildersSecurity, HackThisSite, EthicalHackers and Darkode.

other forum. In figure 3.6, we show the precision and recall of the two approaches. Using OffensiveCommunity as our source, we see that Cross-Seeding improves the precision by 28% and recall by 16% on average across the forums compare to the Basic approach. We also observe that the improvement is substantial: Cross-Seeding improves both precision and recall in all cases.

	OffensiveComm.	HackThisSite	Darkode	Average
Precision	3.3	20.5	17.8	13.2
Recall	8.3	6.4	38.8	17.8

Table 3.2: Characterization: Using two instead of one source forums improves precision and recall on average: Average improvement of using EthicalHackers and WildersSecurity as sources together compared to each of them individually.

c. Using more source forums improves the Cross-Seeding performance significantly. We quantify the effect of having more than one source forums in the classification accuracy of a new forum. We use EthicalHackers and WildersSecurity as our training forums, and we use Cross-Seeding for OffensiveCommunity, HackThisSite, and Darkode. First, we use the source forums one at a time and then both of them together. In table 3.2, we show the average improvement of having two source forums over having one for each target website. Using two source forums increases the classification precision by 13% and the recall by 17% on average.

Discussion: Source forums and training. How would we handle a new forum? Given the above observations, we would currently use all our five forums as sources for a new forum. Overall, we can argue that the more forums we have, the more we can improve our accuracy. However, we would like to point out that some forums are more "similar" and thus more suitable for cross-training. We will investigate how to best leverage a large group of source forums once we collect 20-25 more forums.

3.4 Related Work

We summarize related work clustered into areas of relevance.

a. Extracting IP addresses from security forums. There two main efforts that focus on IP addresses and security forums [39, 42] and neither provides the comprehensive solution that we propose here. The most relevant work [42] does not address the Identification problem, and sidesteps the problem of cross-forum training by assuming training data for each forum. The earlier work [39] focuses on the spatiotemporal properties of Canadian IP addresses in forums, but assumes that all identified addresses are suspicious and therefore they did not employ a classification method, which is the focus of our work.

b. Extracting other information from security forums. Various efforts have attempted to extract other types of information from security forums. A few recent studies identify malicious services and products in security forums by focusing on their availability and price [74, 68].

c. Studying the users and posts in security forums. Other efforts study the users of security forums, group them into different classes, and identify their roles and social interactions [7, 111, 47, 81, 88].

d. Analyzing structured security-related sources. There are several studies that automate the extraction of information from structured security documents, extracting ontology and comparing the reported information, such as databases of vulnerabilities, and security reports from the industry [53, 19, 48].

Transfer learning methods and applications. There is extensive literature on transfer learning [29, 27, 25] and several good surveys [71, 103], which inspired our approach. However, to the best of our knowledge, we have not found any work that address the same domain-specific challenges or uses all the steps of our approach, which we described in 3.2.3.

3.5 Conclusion

We propose a comprehensive solution for mining malicious IP addresses from security forums. A novelty of our approach is it minimizes the need for human intervention. First, once it is initialized with a small number of security forums, it does not require additional training data for each new forum. To achieve this, we use Cross-Seeding, which uses initialization via domain adaptation: we use a classifier from current forums to create seed information for the new forum. Second, it addresses both the Identification and Characterization problems, unlike all prior work that we are aware of. We evaluate our method real data and we show that: (a) our Cross-Seeding approach works fairly well reaching precision above 85% on average for both classification problems, (b) Cross-Seeding outperforms the Basic approach, and (c) using more source forums increases the performance as one would expect.

Our future plans include: (a) collecting a large number of security forums, (b) exploring the limits of the classification accuracy by using more source forums, and (c) exploring additional transfer learning methods.

Chapter 4

Identifying and Classifying Thread of Interest

Security forums hide a wealth of information, but mining it requires novel methods and tools. The problem is driven by practical forces: there is useful information that could help improve security, but the volume of the data requires an automated method. The challenge is that there is a lot of "noise", there is lack of structure, and an abundance of informal and hastily written text. At the same time, security analysts need receive focused and categorized information, which can help their task of shifting through it further. We define the problem more specifically below.

Given a security forum, we want to extract threads of interest to a security analyst. We consider two associated problems that together provide a complete solution. First, the input is all the data of a forum, and the user specifies its interest by providing one or more bag-of-words of interest. Arguably, providing keywords is a relatively easy task for the user.

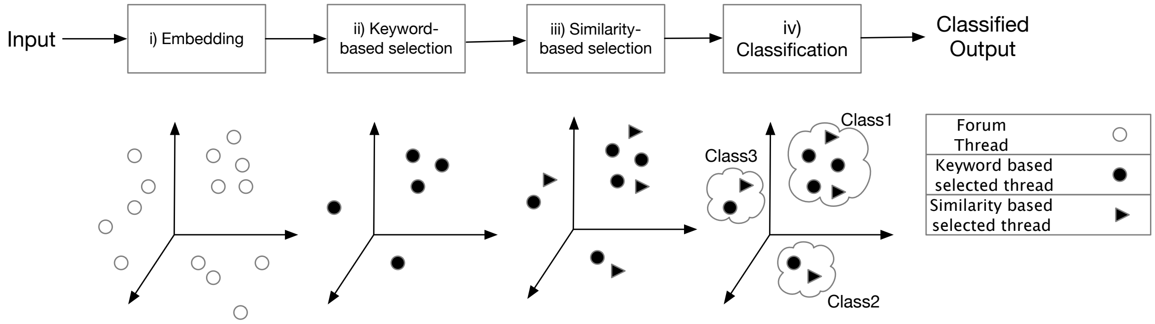

Input
i) Embedding
ii) Keyword-based selection
iii) Similarity-based selection
iv) Classification
Classified Output
Class1
Class2
Class3
Forum Thread ○
Keyword based selected thread ●
Similarity based selected thread ▶

Figure 4.1: An overly-simplified overview of analyzing a forum using the REST approach: i) **project** all threads to embedding space, ii) **select** relevant threads using keyword-based selection, iii) **expand** by adding similar threads, iv) **classify** the threads into classes using supervised learning. We illustrate the embedding space as a three dimensional space.

The goal is to return all the threads that are of interest to the user, and we use the term **relevant** to indicate such threads. A key challenge here is how to create a robust solution that is not overly sensitive to the omission of potentially important keywords. We use the term **identification** to refer to this problem.

Second, we add one more layer of complexity to the problem. To further facilitate the user, we want to group the relevant threads into classes. Again, the user defines these classes by providing keywords for each class. We refer to this step as the **classification** problem. Note that the user can specify the classes of interest fairly arbitrarily, as long as there is training data for the supervised-learning classification.

There is relatively limited work on extracting information from security forums, and even less work on using embedding techniques in analyzing online forum data. We can group prior work in the following categories. First, there is work that analyzes security forums to identify malicious activity [74, 97, 40]. Moreover, there are some efforts to detect malicious users [61, 63] and emerging threats on forums and other social networks [83, 84].

Second, there are several studies on analyzing online forums without a security focus [109, 24]. Third, there is a large body of work in embedding techniques for: (a) analyzing text in general [65, 54], and (b) improving text classification [32, 100, 96]. Also, note that there exist techniques that can do transfer learning between forums and thus, eliminate the need to have training data for every forum [40]. We discuss related work in more detail in our related work section.

We propose a systematic approach to identify and classify threads of interest based on a multi-step weighted embedding approach. Our approach consists of two parts: (a) we propose a similarity-based approach with thread embedding to extract relevant threads reliably, and b) we propose a weighted-embedding based classification method to group relevant threads into user-defined classes.

The key technical foundation of our approach relies on: (a) building on a word embedding to define thread embedding, and (b) conducting similarity and classification at the thread embedding level. Figure 4.1 depicts a high-level visualization of the key steps of our approach: (a) we start with a word embedding space and we define a thread embedding where we project the threads of the forum, (b) we identify relevant threads to the user-provided keywords, (c) we expand this initial set of relevant threads using thread similarity in the thread embedding, (d) we develop a novel weighted embedding approach to classify threads into the four classes of interest using ensemble learning. In particular, we use similarity between each word in the the forums and representing keywords of each class in order to up-weight the word embedding vectors. Then we use weighted embeddings to train an ensemble classifier using supervised learning.

We evaluate the proposed method with three security forums with 163k posts and 21k unique threads. The users in these forums seem to have a wide range of goals and intentions. For the evaluation, we created a labelled dataset of 1350 labeled threads across three forums, which we intend to make available to the research community. We provide more information on our datasets in the next section. section 4.1.

Our results can be summarized into the following points:

a. Providing robustness to initial keyword selection. We show that our similarity-based expansion of the user-defined keywords provides significant improvement and stability compared to simple keyword-based matching. First, the effect of the initial keyword set is minimized: by going from 240 to 300 keywords, the keyword-based method identifies 25% more threads, while the similarity based method increases by only 7%. Second, our approach increases the number of relevant threads by 73-309% depending on the number of keywords. This suggests that our approach is less sensitive to omissions of important keywords.

b. The relevant threads are 22-25% of the total threads. Our approach reduces the amount of threads to 22-25% of the initial threads. Clearly, these results will vary depending on the keywords given by the user and the type of the forum.

c. Improved classification accuracy. Our approach classifies threads of interest in four different classes with an accuracy of 63.3-76.9% and weighted average F1 score 76.8% and 74.9% consistently outperforming five other approaches.

Our work in perspective. Our work is building block towards a systematic, easy to use, and effective mining tool for online forums in general. Although here we focused on security forums, it could easily apply to other forums, and provide the users with the ability to define topics of interest by providing one or more set of keywords. We argue that our approach is easy to use since it is robust and forgiving w.r.t. the initial keyword set.

4.1 Definitions and Datasets

	OffensiveComm.	HackThisSite	EthicalHackers
Posts	25538	84,745	54176
Users	5549	5904	2970
Threads	3542	8504	8745

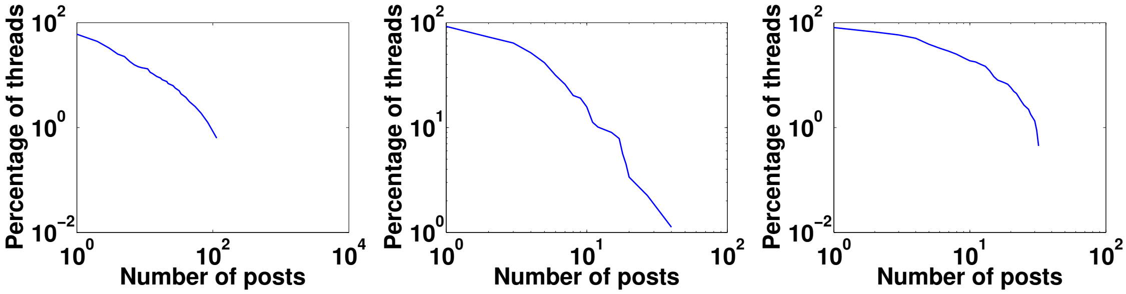
Table 4.1: The basic statistics of our forums.

We have collected data from three different forums: OffensiveCommunity, HackThisSite and EthicalHackers. These forums seem to bring together a wide range of users: system administrators, white-hat hackers, black-hat hackers, and users with variable skills, goals and intentions. We briefly describe our three forums below.

a. OffensiveCommunity (OC): This forum seems to be on the fringes of legality. As the name suggests, the forum focuses on “offensive security”, namely, breaking into systems. Indeed, many posts provide step by step instructions on how to compromise systems, and advertise hacking tools and services.

b. HackThisSite (HT): As the name suggests, this forum has also an attacking orientation. There are threads that describe how to break into websites and systems, but there are also more general discussions about the users’ experiences in cyber-security.

EthicalHackers (EH): This forum seems to consist mostly of “white hat” hackers, as its name suggests. Many threads are about making systems more secure. However, there are many discussions with malicious intents are going on in this forum. Moreover, there are some notification discussions to alert about emerging threats.



(a)OffensiveComm. (b)HackThisSite (c)EthicalHackers

Figure 4.2: CCDF of the number of Post per thread (log-log scale).

Basic forum statistics. We present basic statistics of our forums in Table 4.1. We also study some of their properties and make the following two observations.

Observation 1: More than half of the threads have one post! In Figure 4.2, we plot the complementary cumulative distribution function of the number of post per thread for our forums. We observe the skewed distribution that describes the behavior of large systems. In addition, the distribution shows that more than half of thread has one single post in the threads and 73% of the threads has one or two posts in threads.

Observation 2: The first post defines the thread. Prior research [109] seems to confirm something that we intuitively expect: the first post of the thread pretty much defines the thread. Intrigued, we sampled and manually verified that this seems to be the case. Specifically, we inspected a random sample of 10% of the relevant threads (found by our approach), and we found that more than 97% of the follow up posts fall in line with the topic of the thread: while a majority of them, express appreciation, agreement etc. For example, the follow up posts to a malicious tutorial in OffensiveCommunity were: “Great Tut”, “Thank you for sharing”, “Nice post”, “Work[s] great for me!”

Defining the classes of interest. As we explained in the introduction, we want to further help a security analyst by giving them the ability to define classes of interest among the threads of interest. These are user-defined classes. To ground our study, we focus on the following classes, which we argue could be of interest to a security analyst.

a. Alerts: These are threads where users are reporting about being attacked by a hackers or notifying about exploits and vulnerabilities. An example from EthicalHackers is a thread with the title “Worm Hits Unsecured Space Station Laptops” and the first line of the first post is “NASA spokesman Kelly Humphries said in a statement that this was not the first time that the ISS had been affected by malware, merely calling it a “nuisance.””

b. Services: These are threads where users are offering or requesting malicious hacking services or products. An example from OffensiveCommunity is a thread with the title “Need hacking services” and this first line “Im new to this website. Im not a hacker. Would like to hire hacking services to hack email account, Facebook account and if possible iPhone.”

c. Hacks: These are threads where users post detailed instructions for performing malicious activities. The difference with the above category is that the information is offered for free here. An example from OffensiveCommunity is a thread titled “Hack admin account in XP, Vista, Windows 7 and Mac - Complete beginners guide!!” with a first line: “Hack administrator account in XP OS – Just by using command prompt is one of the easiest ways (without installation of any programs).....”. As expected, these posts are often lengthy as

	OffensiveComm.		HackThisSite		EthicalHackers	
Labeled	450		450		450	
	#	%	#	%	#	%
Hacks	202	45%	31	7%	42	9%
Services	204	45%	286	64%	166	37%
Alerts	27	6%	20	4%	78	18%
Experiences	17	4%	113	25%	164	36%

Table 4.2: Our groundtruth data for each forum and the breakdown per class.

they convey detailed information.

d. Experiences: These are threads where users share their experience related to general security topics. Often users provide a personal story, a review or an article on a cyber-security concept or event. For example, in HackThisSite a thread titled “Stupid people stories”, the author explains cyber-security mistakes that he made.

The sets of keywords which “define” each class are shown in Table 4.5. Clearly, these sets will be provided by the user depending on classes of interest. Note that these keywords are also provided to our annotators as hints for labeling process.

4.1.1 Establishing the Groundtruth

For validating our classification method, we need groundtruth to do both the training and the validation. We randomly selected 450 threads among the relevant threads from each forum as selected by the identification part. The labelling involves five annotators that manually label each thread to a category based on the definitions and examples of the four classes which we listed above. The annotators were selected from a educated and technically savvy group of individuals to improve the quality of the labels. We then combine

Label	Hacks	Services	Alerts	Experiences
OffensiveComm.	0.778	0.702	0.816	0.732
HackThisSite	0.953	0.966	0.793	0.875
EthicalHackers	0.682	0.733	0.766	0.620

Table 4.3: Assessing the annotator agreement using the Fleiss-Kappa coefficient for each class for our three datasets.

the “votes”, and assign the class selected by the majority.

We assess the annotators’ agreement based on the Fleiss-Kappa coefficient and we show the results in Table 4.3. We see that there is a high annotator agreement across all forums as the Fleiss-Kappa coefficient is 78.6. 92.6, 70.3 for OffensiveCommunity, HackThisSite and EthicalHackers respectively.

With this process, we labelled 1350 posts in three forums and we present our labeled data in Table 4.2. We make our groundtruth available to the researchers in the community in order to foster follow up research. [1]

4.1.2 Challenges of simple keyword-based filtering

Given a set of keywords, the most straightforward approach in identifying relevant documents (or threads here) is to count the combined frequency with which these keywords appear in the document. A user needs to identify the keywords that best describe the topics and concepts of interest, which can be challenging for non-trivial scenarios [102]. We outline some of the challenges below.

- The user may not be able to provide all keywords of interest. In some cases, the user is not aware of a term, and in some cases, this not even possible: consider the case

[1] Data is provided at the following link: https://github.com/icwsmREST2019/RESTDATA.

where we want to find the name of a new malware that has not yet emerged.

- Stemming, variations and compound words is a concern. The root of a word can appear in many different versions: e.g. hackers, hacking, hacked hackable, etc. There exist partial solutions for stem but challenges still remain [51].
- Spelling errors and modifications and linguistic variations. Especially for an international forum, different languages and backgrounds can add noise.

The above challenges motivated us to consider a new approach that uses a small number of indicative keywords to create a seed set of threads, and then use similarity in the embedding space to find more similar threads, as we describe in the next section.

4.2 Identifying threads of interest

We present our approach for selecting relevant threads starting from sets of keywords provided by the user. Our approach consists of the following phases: (a) a keyword matching step, where we use the user-defined keywords to identify relevant threads that contain these keywords, and (b) a similarity-based phase, where we identify threads that are "similar" to the ones identified above. The similarity is established at the word embedding space as we describe later.

4.2.1 Phase 1: Keyword-based selection

Given a set or sets of keywords, we identify the threads where these keywords appear. A simple text matching approach can distinguish all occurrence of such keywords

Symbol	Description
v_i	Word i in a forum
$\vec{v_i}$	Embedded vector for word i
$v_{i,k}$	Value of dimension k in embedded vector for word i
t_r	Thread r
m	The dimensions of the word embedding space
n	Number of words in a thread
d	Number of words in a forum
$W(t_r)$	Set of words in thread r
D	Set of words in a forum
$\vec{P}(e)$	Embedding projection of entity e (word, thread etc)
$Sim(w, c)$	Similarity of vectors w and c
w_k	The "center of gravity" word for class k
$\vec{\beta}_k$	Affinity vector of class k
$\vec{\beta}_k[i]$	Value of Affinity vector of class k at index i
WS_l	Keyword set l for identifying relevant threads
T_{key}	Keyword threshold in identifying relevant threads
T_{sim}	Similarity threshold in identifying relevant threads

Table 4.4: The symbol table with the key notations.

in the forum threads. In more detail, we follow the steps below:

Step 1: The user provide a set or sets of keywords WS_l, which capture the user's topics of interest. Having sets of keywords enables the user to specify combinations of concepts. For example, in our case we use, the following sets: (a) hacking related, (b) exhibiting concern and agitation, and (c) searching and questioning.

Step 2: We count the frequency of each keyword in all the threads. This can be done easily with elastic search or any other straightforward implementation.

Step 3: We identify the relevant threads, as the threads that contain a sufficient number of keywords from each set of keywords $WildersSecurity_l$. This can be defined by a threshold, T_{key_l}, for each set of keywords.

Going beyond simple thresholds in this space, we envision a flexible system, where the user can specify complex queries that involve combinations of several different keyword sets $WildersSecurity_l$. For example, the user may want to find threads with: (a) at least 5 keywords from set $WildersSecurity_1$ and 3 keywords from $WildersSecurity_2$, or (b) at least 10 keywords from $WildersSecurity_3$.

4.2.2 Phase 2: Similarity-based selection

We propose an approach to extract additional relevant threads based on their similarity to existing relevant threads. Our approach is inspired by and combines elements from earlier approachs [65, 90], which we discuss and contrast with our work in the related work section.

Overview of our approach. Our approach follows the steps below, which are also depicted visually in Figure 4.1. The input is a forum, a set of keywords, and set of relevant threads, as identified by the keyword-based phase above.

Step 1. Determining the embedding space. We project every word as a point in a m-dimensional space using a word embedding approach. Therefore, every word is represented by a vector of m dimensions.

Step 2. Projecting threads. We project all the threads in an appropriately constructed multi-dimensional space: both the relevant threads selected from the keyword-based selection and the non-selected ones. The thread projection is derived from the projections of its words, as we describe below.

Step 3. Identifying relevant threads. We identify more relevant threads among the non-selected threads that are "sufficiently-close" to the relevant threads in the thread embedding space.

The advantage of using similarity at the level of threads is that thread similarity can detect high-order levels of similarity, beyond keyword-matching. Thus, we can identify threads that do not necessary exhibit the keywords, but use other words for the same "concept". We show examples of that in Tables 4.9 and 4.10.

Our similarity-base selection in depth. We provide some details in several aspects of our approach.

Step 1: in depth. We train a skip-gram word embedding model to project every word as a vector in a multi-dimensional space [65]. Note that we could not use pre-trained embedding models, since there are many words in our corpus that do not exist in the dictionary of previous models.

The number of dimensions of the word embedding can be specified by the user: NLP studies usually opt for a few hundred dimensions. We discuss how we selected our dimensions in our experiments section.

At the end of this step, every word v_i is projected to $\vec{P}(v_i)$ or $\vec{v_i}$, a real-value m-dimensional vector, $(v_i[1], v_i[2], ..., v_i[m])$. A good embedding ensures that two words are similar, if they are close in the embedding space.

Step 2: in depth. We project the threads in an $2m$-space, by "doubling" the m-dimensional space that we used for words as we will show below. The thread projection is a function of the vectors of its words and captures both the average and the maximum values of the vectors of its words.

a. Capturing the average: $P_{avg}(t_r)$. Here, we want to capture the average "values" of the vectors of the words in the thread. For thread t_r, the average projection,

$P_{avg}(t_r)$ is calculated as follows for each dimension l in the m-dimensional word space:

$$\vec{P}_{avg}(t_r)[l] = \frac{1}{|W(t_r)|} \cdot \sum_{v_i \in W(t_r)} \vec{v_i}[l], \tag{4.1}$$

Recall that $W(t_r)$ is the set of words of the thread. For simplicity, we refer to projection of word v_i as $\vec{v_i}$ instead of the more complex $\vec{P}(v_i)$.

b. Capturing the high values: $P_{max}(t_r)$. Averaging can fail to represent adequately the "outlier" values, and to overcome this, we calculate a vector of maximum values, $\vec{P}_{max}(t_r)$, for each thread. For each dimension l in the word embedding, $P_{max}[l]$ is the maximum value of that dimension over all existing l-dimension values among all the words in the thread, which we can state more formally below:

$$\vec{P}_{max}(t_r)[l] = \max_{v_i \in W(t_r)} \vec{v_i}[l] \tag{4.2}$$

Finally, we create the projection of thread t_r by using both these vectors, $\vec{P}_{avg}(t_r)$ and $\vec{P}_{max}(t_r)$, as this combination has been shown to provide good results [90]. Specifically, we concatenate the vectors and we create the thread representations in an $2m$-dimensional space.

$$\vec{P}(t_r) = (\vec{P}_{avg}(t_r), \vec{P}_{max}(t_r)) \tag{4.3}$$

Step 3: in depth. We identify similar threads at the $2m$-space-dimensional space of thread embedding from step 2. We propose to use the cosine-similarity determine the similarity among threads, which seems to give good results in practice. Most importantly, we can control what constitutes a *sufficiently-similar* thread using a threshold T_{sim}. The threshold needs to strike a balance between being too selective and too loose in its definition of similarity. Furthermore, note the right threshold value also depends on the nature of the problem and the user preferences. For example, a user may want to be very selective, if the resources for further analyzing relevant threads is limited or if the starting number of threads is large.

4.3 Classifying threads of interest

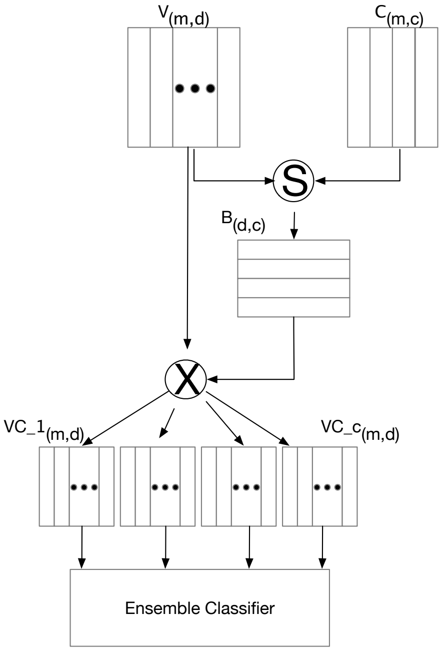


Figure 4.3: A visual overview of our classifier

We present our approach for classifying relevant threads into user defined classes. To ground the discussion, we presented four classes on which we focus here, but our approach can be applied for any number and type of classes as long as there is training data for the supervised learning.

Defining Affinity. We use the term **affinity**, $\vec{\beta}_k[i]$, to refer to the "contribution" of word v_i in a thread towards our belief that the thread belongs to class k.

Recall also that each class k is characterized by a group of words that we denote as $WordClass_k$. These sets of words are an input to our algorithm and in practice they will be provided by the user.

High-level overview creating our classifier. Our approach consists of the following steps, which are visually represented in Figure 4.3.

Step 1. We create a representation of every class k into the word embedding space by using the words that define the class, $WordClass_k$.

Step 2. For all the words in the forum, we calculate the affinity of the word v_i for each class k, $\vec{\beta}_k[i]$.

Step 3. For each class, we create a weighted embedding by using the affinity to adjust the embedding projection of each word for each class.

Step 4. We use weighted embedding to train an ensemble classifier using supervised learning.

Using the classifier. Given a thread, we calculate its projection in the embedding space, and then we pass it to the classifier to determine its class.

Our algorithm in more detail. In the remainder of this section, we provide a more in depth description of the algorithm.

Step 1: in depth. For each class k, we use the set of words $WordClass(k)$, and to define a representation, $\vec{w}_k$, for that class in the word embedding space. We project each word in $WordClass(k)$ to the embedding space by using the same word embedding model, which we trained in the previous section. Then, we define the class vector w_k to be the average of the word embeddings of the words in $WordClass(k)$ similarly to equation 4.1. Note that these class embedding vectors correspond to each column of the matrix $C_{(m,c)}$ in Figure 4.3, where m in the dimension of the embedding and c is the number of classes.

Step 2: in depth. The affinity of each word v_i in the forum for each class is calculated by the similarity of the word v_i to $\vec{w}_k$, which represents the class in this space. We calculate the proximity using the cosine similarity, as follows:

$$Sim(\vec{v_i}, \vec{w}_k) = \frac{\vec{v_i} \cdot \vec{w}_k}{||\vec{v_i}|| \cdot ||\vec{w}_k||} \tag{4.4}$$

Then, for each class k, we create vector $\vec{\beta_k}$ whose element $[i]$ corresponds to the affinity of word v_i of the forum D. Specifically, we normalize the values by using *Softmax* of the similarity vector $Sim(v_i, w_k)$ as follows:

$$\vec{\beta_k}[i] = \frac{exp(Sim(\vec{v}_i, \vec{w}_k))}{\sum_{y_j \in D} exp(Sim(\vec{y}_j, \vec{w}_k))}, \tag{4.5}$$

where $y_j \in D$ iterates through all the words in the forum. Note that $\vec{\beta_k}$ corresponds to a row k in matrix $B_{d,c}$ in figure 4.3, where c is the number of classes and d is the total number of words in the forum.

Step 3: in depth. For each class k, we create a “custom” word embedding, $VC_k(m, d)$ in Figure 4.3. Each such matrix that is focused on detecting threads of k and it will be used in our ensemble classification.

For each class, we create, $VC_k(m, d)$, a class-specific word embedding by modifying the word projections, $\vec{v_i}$ using the affinity of the word $\vec{\beta_k}[i]$ for class. Formally, we calculate

VC_k by calculating column $VC_k[*, i]$ as follows:

$$VC_k[*, i] = \vec{\beta_k}[i] \cdot \vec{v_i} \tag{4.6}$$

where $\vec{\beta_k}[i]$ is the affinity value of word v_i for class k.

For each thread t_r, we calculate the projection of the thread by calculating $\vec{P}_{avg}(t_r)$ and $\vec{P}_{max}(t_r)$ using the modified word projections, $\vec{\beta_k}[i] \cdot \vec{v_i}$, captured in the $VC_k(m, d)$ matrix and using equations 4.1 and 4.2. Finally, we create the projection of each thread in the $2m$-space, using equation 4.3.

Step 4: in depth: We use weighted embeddings of threads to train an ensemble classifier using supervised learning.

For each class k, we train the classifier by using the weighted representation vector in a supervised learning. Each VC_k in Figure 4.3 becomes the basis for a classifier with weighted penalty in favor of class k. The ensemble classifier combines the classification results from each VC_k classifier using the max-voting approach [55].

Using contextual features. Apart from the words in the forum, we can also consider other types of features, which we refer to as contextual features of the threads. One could think of various such features, but here we list the features that we use in our evaluation: (1) number of newlines, (2) length of the text, (3) number of replies in the thread (following posts after the first post), (4) average number of newlines in replies, (5) average length of replies, and (6) the aggregated frequency of the words of each bag-of-words set provided by the user.

These features capture contextual properties of the posts in the threads, and pro-

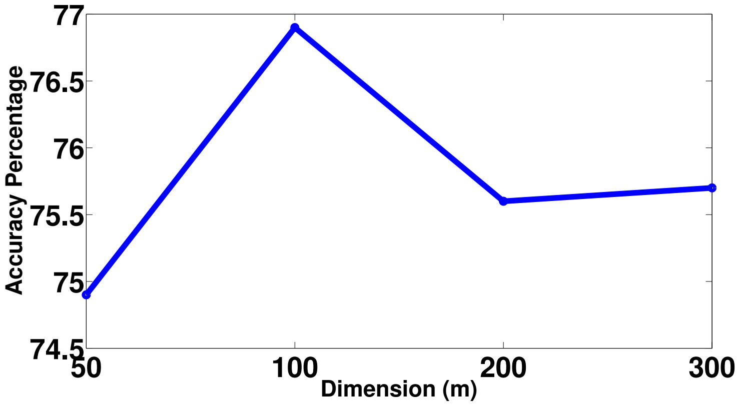


Figure 4.4: Selecting the number of dimensions of word embedding: the accuracy of REST for different dimensions in OffensiveCommunity.

Hacks	Services	Alerts	Experiences
Tutorial	tool	announced	article
guide	price	reported	story
steps	pay	hacked	challenge

Table 4.5: $WordClass$, the set of words which ”define” each class.

vide additional information not necessarily captured by the words in the thread. Empirically, we find that these features improve the classification accuracy significantly. The inspiration to introduce such features came from manually inspection of posts and threads. For example, we observed that Hacks and Experiences usually have longer posts than other. Moreover, Hacks threads contain a larger number of newline characters. An interesting question is to assess the value of such metrics when used in conjuction with word-based features.

4.4 Experimental Results

We present our experimental results and evaluation of our approach.

4.4.1 Conducting our study

We use the three forums that presented in Table 4.1 and the groundtruth, which we created as we explained in section definitions.

Keywords sets: We considered three keyword sets to capture relevant threads. These keywords set are: (a) hacking related, (b) exhibiting concern and agitation, and (c) searching and questioning. We collected a set of more than 290 keywords in three sets. We started with a small core group of keywords, which we expanded by adding their synonyms using thesaurus.com and Google's dictionary. We ended up with 68, 207 and 17 keywords for the three groups respectively.

These keyword sets are used in extracting relevant threads with the keyword-based selection. We select a thread, if it contains at least one word from each keyword set: $T_{key_1}, T_{key_2}, T_{key_3} >= 1$. As we discussed earlier, there are many different ways to perform this selection in the presence of multiple groups of words and depending on the needs of the problem.

Pre-processing text: As with any NLP method, we do several pre-processing steps in order to extract an appropriate set of words from the target document. First we tokenize the documents in to bigrams, then we remove the stopwords, numbers and IP addresses based on a recent work [40]. In addition, here we opt to focus on the title and the first post of a threads instead of using all the posts. Our rationale is based on the two observations regarding the nature of the threads: (a) most of them have one post anyway, and (b) the title and the post typically define their essence. In the future, we will examine the effect of using the text from more posts from each thread.

Identification: Implementation choices. The identification algorithm requires some implementation choices, which we describe below.

Embedding parameters: We set the window size to 10 and we tried several different values as the dimension of the embedding between 50-300, and we found that $m = 100$ with the highest accuracy as depicted in Figure 4.4 and m is in the range of choice of other studies in this space.

Similarity threshold: $T_{sim} = 0.96$. The similarity threshold T_{sim} determines the "selectiveness" in identifying similar threads, as we described in a previous section. We find that a value of 0.96 worked best among all the different values we tried. It strikes the balance between being: sufficiently selective to filter out non-relevant threads, but sufficiently flexible to identify similar threads.

Classification: Implementation choices. We present the implementation choices for our classification study.

Evaluation Metrics: We used the accuracy of the classification along with the average weighted F1 score, which is designed to take into consideration the size of the different classes in the data.

Our classifier. We use random forest as our classification engine, which performed better than several others that we examined, including SVM, Neural Networks, and K-nearest-neighbors. Results are not shown due to space limitations.

Class defining words: The set of keywords we have used for each class are as shown in Table 4.5.

Baseline methods. We evaluate our approach against five other state of the arts methods, which we briefly describe below.

- **Bag of Words (BOW)**: This methods uses the word frequency (more accurately the TFIDF value) as its main feature [64, 42, 50].
- **Non-negative Matrix Factorization (NMF)**: This method uses linear-algebra to represent high-dimensional data into low-dimensional space, in an effort to capture latent features of the data [59].
- **Simple Word Embedding Method (SWEM)**: There is a family of methods that use the word2vec as their basis, and use a recently proposed method [90].
- **FastText (FT)**: Similar to NMF and SWEM, FastText represents words and text in a low dimensional space [54].
- **Label Embedding Attentive Model (LEAM)**: This is the most recent approach [100] claims to outperform other state of art methods including PTE [96]. We used their provided linear implementation of their attentive model.
- **Bidirectional Encoder Representations from Transformers (BERT)** : This is a new pre-trained Deep Bidirectional Transformer for Language Understanding introduced by Google [32]. BERT provides contextual representation for text, which can be used for a wide range of NLP tasks. As we discuss later, BERT did not provide

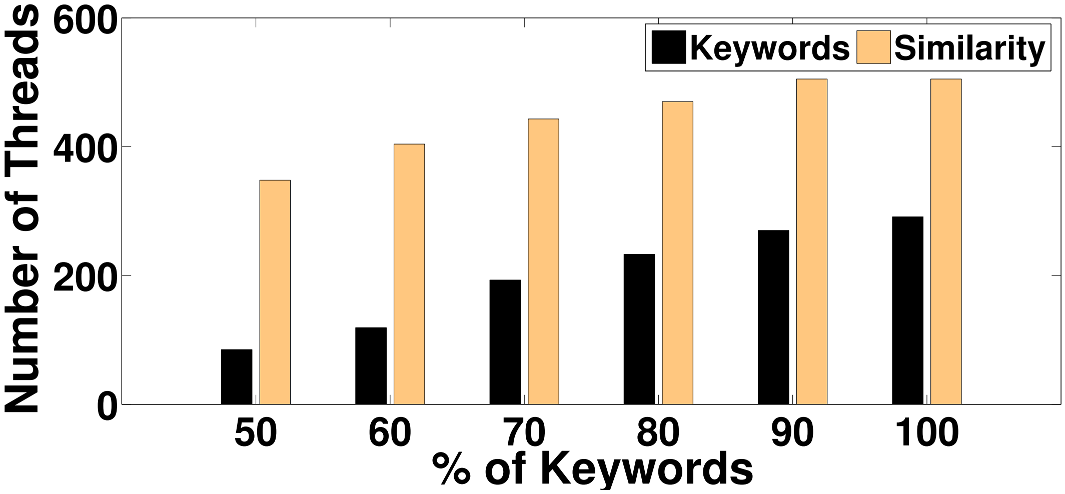


Figure 4.5: The robustness of the similarity approach to the initial keywords: number of relevant threads as a function of the number of keywords for OffensiveCommunity.

Relev. Threads	OffensiveComm.	HackThisSite	EthicHack
Keyword	291	840	893
Similarity	505	1121	1360
Total	796	1961	2753
Total(%)	22%	23%	25%

Table 4.6: The relevant threads and their identification method: keywords and similarity. The total percentage refers to the selected threads over all the threads in the forum.

good results initially, and we created a tuned version to make the comparison more meaningful.

4.4.2 Results 1: Thread Identification

We present the results from the identification part of our approach.

Our similarity-based method is robust to the number of initial keywords. We want to evaluate the impact of the number of keywords to the similarity based method. In Figure 4.5, we show the robustness of each identification methods to the initial set of keywords for OffensiveCommunity. By adding 60 keywords, from 240 to 300, the keyword-based method identifies 25% more threads, while the similarity based method has only 7% increment. Similarly, doubling the initial size of the keywords results in 242% increase for the keyword-based method but only 45% in the similarity-based method.

We argue that our approach is more robust to the initial number of keywords. First, with less number of keywords, we retrieve more threads. Second, an increase in the number of keywords has less relative increase in the number of threads. This is an initial indication that our approach can achieve good results, even with a small initial set of keywords.

Evaluation of our approach: High precision and reasonable recall. We show that our approach is effective in identifying relevant threads. Evaluating precision and recall would have been easy if all the threads in a forum were labelled. Instead, we use an indirect method to gauge recall and precision as we describe below.

Indirect estimation of recall. We consider as "groundtruth" the relevant threads that we find with set of keywords in keyword-based selection method and report how many of those threads that our method finds with only 50% of the keywords in similarity-based selection. The experiment is shown in Figure 4.5. We use only 50% of the keywords to extract the relevant threads with the similarity selection approach, and then compare it with the relevant threads identified with larger set of keywords [60-100]%. We show in Table 4.7 that with 50% of the keywords we can identify more than 60-70% of the relevant threads, which we identify if we have more keywords available.

Estimating precision. To evaluate precision, we want to identify what percent-

keywords %	60	70	80	90	100	Avg.
OffensiveComm.	78.2	76.9	72.9	70.8	70.1	70.94
HackThisSite	74.82	72.01	70.68	69.92	69.74	71.43
EthicHack	68.41	60.4	60.8	57.2	56.51	60.67

Table 4.7: Identification: Indirect ”gauge” of Recall: We report how many threads our method finds with 50% keywords compared to the keyword based selection with larger sets of keywords [60-100]% .

	OffensiveComm.	HackThisSite	EthicHack	Avg.
Precision	98.2	97.5	97.0	97.5

Table 4.8: Identification Precision: the precision of the identified thread of interest with the similarity-based method.

age of the retrieved threads are relevant. To this end, we resort to manual evaluation. We have labeled 300 threads from each dataset retrieved with 50% of the keywords and we get our annotators to identify if they are relevant. We show the results in Table 4.8. We understand that on average more than 97.5% of the threads identified with the similarity based method are relevant with an inter-annotator agreement Fleiss-Kappa coefficient of 0.952.

The power of the embedding in determining similarity. We find that the similarity step identifies threads that are deemed relevant to a human reader, but are not “obviously similar”, if you examine the threads word for word. We provide a few examples of threads that were identified by the keyword-based selection, and the related similar threads that our approach identified. Table 4.9 and 4.10 illustrate how the retrieved thread are similar to the target thread conceptually, without matching linguistically.

A four-fold thread reduction. Our approach reduces the amount of threads to only 22-25% of the initial threads as shown in Table 4.6. Figure 4.6 depicts the same data

Selection Method	Title	Post
Keyword selected	[ULTIMATE] How to SPREAD your viruses successfully [TUTORIAL]	Educational Purposes NOT MINE In this tutorial I will show you how to spread your trojans/viruses etc. I will show you many methods, and later you choose which one ....
Similarity selected	Botnet QA!	Just something I compiled quickly. Im also posting my bot setup guide soon. If you want any questions or links added to the Q&A, please ask and Ill add them.
	The COMPLETE beginners guide to hacking	another great guide i found :D Sections: 1) Introduction 2) The hacker manifesto 3) What is hacking? 4) Choosing your path 5) Where should I start? 6) Basic terminology 7) Keylogging...
	[TUT]DDoS Attack - A life lesson	Introduction I know their are a lot more ways to DoS than are shown here, but ill let you figure them out yourself. If you find any mistake in this tutorial please tell me‘What is D́DoS?́

Table 4.9: Examples of similar threads for class Hacks: threads offering hacking tutorials.

visually. Clearly, these results will vary depending on the keywords given by the user and the type of the forum.

4.4.3 Results 2: Thread Classification

We present the results of our classification study.

REST compared to the state-of-the-art. Our approach compares favourably against the competition. Table 4.11 summarizes the results of the baseline methods and our REST for three forums. REST outperforms other baseline methods with at least 1.4 percentage point in accuracy and 0.7 percentage point in F1 score, except BERT. First,

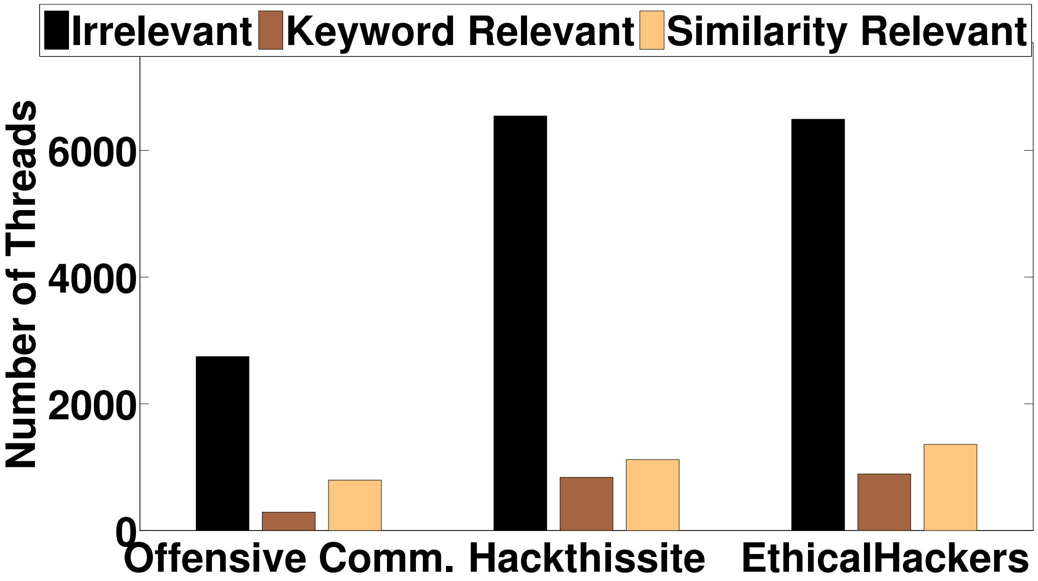


Figure 4.6: Number of relevant thread in each forums identified by our approach: (a) irrelevant (not selected), (b) selected via keyword matching and (c) selected via similarity.

using BERT "right out of the box" did not give good results initially. However, we fine-tuned BERT for this domain. BERT performs poorly on two sites, HackThisSite and EthicalHackers, while it performs well for OffensiveCommunity. We attribute this to the limited training data in terms of text size and also the nature of the language users use in such forums. For example, we found that the titles of two misclassified threads contained typos and used unconventional slang and writing structure " Hw 2 gt st4rtd with r3v3r53 3ngin33ring 4 n00bs!!", "metaXploit 3xplained fa b3ginners!!!". We intend to investigate BERT and how it can be tuned further in future work. Note that methods BOW and NMF did not assign any instances to the minority classes correctly, therefore the value of F1 score in Table 4.11 is reported as NA.

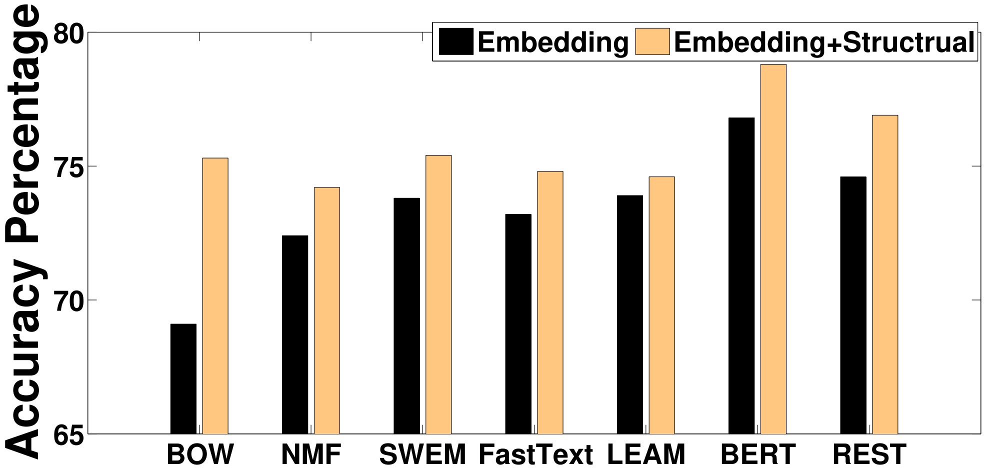


Figure 4.7: Classification accuracy for two different features sets in 10-fold cross validation in OffensiveCommunity forum.

The contextual features improves classification for all approaches. We briefly discussed contextual features in our classification section. We conduced experiments with and without these features for all six algorithms and we show the results in Figure 4.7 for OffensiveCommunity. Including the contextual features in our classification improves the accuracy for all approaches (on average by 2.4%). The greatest beneficiary is the Bag-of-Words method whose accuracy improves by roughly 6%.

4.5 Related Work

We summarize related work group into areas of relevance.

a. Identifying entities of interest in security forums. Recently there have been a few efforts focused on extracting entities of interest in security forums. A very interesting study focuses on the dynamics of the black-market of hacking goods and services and their pricing [74], which for us is one of the categories of interest. Some other recent efforts focus on identifying malicious IP addresses that are reported in the forum [40, 42], which is relatively different task, as there, the entity of interest has a well-defined format. Another interesting work [97] uses a word embedding technique focusing identifying vulnerabilities and exploits.

b. Identifying malicious users and events. Several studies focus on identifying key actors and malicious users in security forums by utilizing their social and linguistics behavior. [61, 63, 7]. Another work [41, 83] identifies emerging threats by monitoring threads activities and the behavior of malicious users and correlating it with information from security experts on Twitter. Another study [84] detects emerging security concerns by monitoring the keywords used in forums and other online platforms, such as blogs.

c. Analyzing other online forums. Researchers have analyzed a wide range of online forums such as blogs, commenting platforms, reddit etc. Indicatively, we refer to a few recent studies. Google [109] analyzed *question-answer* type of forums and they also published the large dataset that they collected. Another study focusing on detecting question-answer threads within a discussion forum using linguistic features [24].

Despite many common algorithmic approaches, we argue that each type of forum and different focus questions necessitate novel algorithms.

d. NLP, Bag-of-Words, and Word Embedding techniques. Natural Language Processing is a vast field, and even the more recent approaches, such as query transformation and word embedding have benefited from significant numbers of studies [85, 64, 65, 58, 61, 50, 100, 108, 90, 59]. Most recently, several methods focus on combining word embedding and deep learning approaches for text classification [100, 108, 90, 32].

We now discuss the most relevant previous efforts. These efforts use word embedding representation and they use it for classification for text, but: (a) neither of those focuses on forums, (b) there are some other technical differences with our work. The first work, predictive text embedding (PTE) [96], uses a network-based approach, where each thread is described by a network of interconnected entities (title, body, author etc). The second study, LEAM [100], uses a word embedding and a Neural Network classifier to create a thread embedding. LEAM argues that it outperforms PTE, and as we show here, we outperform LEAM. Recently Google introduced BERT [32], a deep pre-trained bidirectional transformers for language understanding which uses a pre-trained unsupervised language model on large corpus of data. Although the power of large data set for training is indisputable, at the same time, we saw first hand the need for some customization for each domain space.

Finally, there are some efforts that use Doc2Vec to identify the embedding of a document (equivalently threads in our case). However, these techniques would not work well here due to the small size of the datasets [58]. This technique could be applied in much larger forums, and we will consider it in such a scenario in the future.

4.6 Conclusion

There is a wealth of information in security forums, but still, the analysis of security forums is in its infancy, despite several promising recent works.

We propose a novel approach to identify and classify threads of interest based on a multi-step weighted word embedding approach. As we saw, our approach consists of two parts: (a) a similarity-based approach to extract relevant threads reliably, and b) weighted embedding-based classification method to classify threads of interest into user-defined classes. The key novelty of the work is a multi-step weighted embedding approach: we project words, threads and classes in the embedding space and establish relevance and similarity there.

Our work is a first step towards developing an easy-to-use methodology that can harness some of the information in security forums. The easy-of-use stems from the ability of our method to operate with an initial set of bag-of-words, which our system uses to seeds to identify threads that the user is interested in.

Selection Method	Title	Post
Keyword selected	Blackmailed! How to hack twitter?	Hey, everyone. Im new on this website and I need help. Im trying to hack a twitter account because theyve been harassing me and reporting isnt helping at all.
Similariry selected	Need hacking services	IIm new to this website. Im not a hacker. Would like to hire hacking services to hack email account, Facebook account and if possible iPhone. Drop me a pm if you can do it. Fee is negotiable. Thanks
	Hello hacker members	My name is XXXX and im looking for someone to help me crack a WordPress password from a site that has stolen all our copyrighted content. Weve reported to google but is taking forever. I have the username of the site, just need help to crack the password so i can remove our content. Please message me with details if you can help
	finding a person with his email	Hello guys! I need to find out how I can find a person b́ehindán email! Let me explain please ...
	Hi	hello everyone im new here and i want to learn how to hack an account any account in fact fb twitter even credit card hope you code help me out who knows maybe i can help you in the future right give and take

Table 4.10: Examples of similar threads for class Services: threads looking for hacking services.

Datasets	Metrics	BOW	NMF	SWEM	FastText	LEAM	BERT	**REST**
OffensiveComm.	Accuracy	75.33±0.1	74.31±0.1	75.55±0.21	74.64±0.15	74.88±0.22	**78.58± 0.08**	77.1±0.18
	F1 Score	NA	NA	74.15±0.23	72.5±0.15	72.91±0.18	**78.47±0.01**	75.10±0.14
HackThisSite	Accuracy	65.3±0.41	69.46±0.12	73.27±0.10	69.92±0.08	74.6±0.04	68.99±0.4	**76.8± 0.1**
	F1 Score	NA	70.23±0.13	71.89±0.14	65.81±0.4	71.41±0.09	63.61±0.41	**74.47±0.24**
EthicalHackers	Accuracy	59.74± 0.21	58.3± 0.15	61.3± 0.17	59.73± 0.21	61.80 ±0.13	54.91± 0.32	**63.3± 0.09**
	F1 Score	NA	57.83±0.16	59.6±0.23	59.5±0.13	60.9±0.17	51.78±0.15	**61.7±0.21**

Table 4.11: Classification: the performance of the five different methods in classifying threads in 10-fold cross validation.

Chapter 5

IKEA: Unsupervised Domain-Specific Keyword-Expansion

What are the relevant keywords to search a forum so that we can maximize the amount of useful information that we can extract? This is the overarching question that motivates this work. First, we argue that there is a wealth of information in online forums. These forums aggregate the collective wisdom of millions of people around the world, and they capture useful information, signals and trends. Second, we focus on security forums to ground our work. Thus, our goal is to help a security analyst by making our approach: (a) easy to use, by asking few initial keywords, and (b) flexible to cater to a wide range of types of investigations.

We define the problem in more detail. The user provides: (a) an initial set of

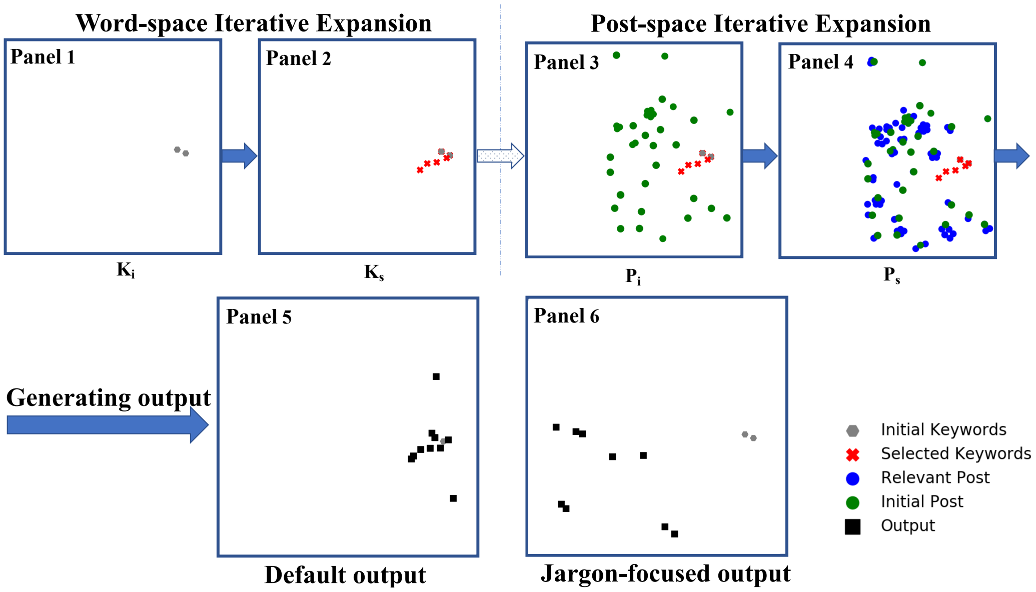


Figure 5.1: A visual overview of our approach using samples from a real query : (a) iterative expansion in the word-space from the initial keywords (Panel 1, gray circles) to the expanded keyword set K_s (Panel 2, red crosses); (b) iterative expansion in the post-space from the posts P_i obtained using the K_s keywords (Panel 3, green circles) to the extended post set P_s (Panel 4, blue circles). Output: the top 10 highest ranked words based on user preference for default in Panel 5, and jargon-focused in Panel 6.

keywords, (b) a sample forum, and (c) her expansion preference. The output is an expanded set of **ranked** keywords from the sample forum that best relates to the initial keywords. We consider the following requirements. First, we want our approach to work even with a really small set of initial keywords. Second, we want to enable the user to get the answer that best matches their intention, which we explain below.

We provide an example to clarify the problem and the concept of user preference. The user could provide two keywords *virus* and *attack* or a name of a virus. The goal is to retrieve the most relevant and important keywords leveraging the sample forum. We consider this as the **default** type of user intention, where a response could include words like *malware, ransomware*, or *antivirus*. However, a user may have a preference towards identifying specific names of malware, tools or technical jargon, in which case they would prefer an answer that would include words like *kraken* and *rustock*. Namely, the goal is to return the proper names and jargon, akin to the system learning by example in an unsupervised fashion. We use the term **jargon-focused** for this type of user preference.

The above problem formulation has received relatively little attention, and usually in a tangential way. We can group prior work as follows. First, there have been several embedding-based techniques for query expansion [33, 56, 78], which we compare with our approach in sections *Experimental Results* and *Related Work*. Second, some efforts focus on topics and keyword extraction from a document [16, 38], without an initial keyword set. Third, other studies apply NLP-based techniques to identify specific information and user interactions, such as malicious IP addresses, and selling of services in security forums [40, 74, 97]. We elaborate on previous work in section *Related Work*.

We propose a systematic approach, IKEA[1], to expand an initial limited set of keywords focusing on a specific domain in an unsupervised learning fashion. The novelty of our approach is three-fold: (a) we use and combine two similarity expansions in the word-word and post-post spaces, in an appropriately constructed embedding space, (b) we use an iterative approach for each of the aforementioned expansions, and (c) we provide a flexible processing of the identified words. The flexibility in the last step refers to our ability to rank the retrieved words in the order that best suits the needs of the user query as in the example mentioned above [2].

We provide a high-level view of our approach in Figure 5.1. It shows how we

[1] IKEA stands for **I**terative **K**eyword **E**xpansion **A**pproach.

[2] The jargon-focused case can be seen as a variation of the named entity identification. Here, we address a *targeted* named-entity identification, a less-explored variation, and our goal is to showcase the flexibility introduced by the last step in our approach.

start from an initial set of keywords, which we project in appropriately-defined embedding space. Second, we use an iterative process to identify similar keywords, set K_s, in the word embedding space (Panel 2). Third, we identify the posts, P_i, that contain keywords from the set K_s. Fourth, we use an iterative process to identify a set of similar posts, P_s, in the post embedding space (Panel 4). Finally, we extract keywords from the set of posts P_s and present them to the user ranked according to their preference.

We evaluate our method using three security forums over a five-year period. For the evaluation, we created a labeled dataset using both: (a) the Mechanical Turk service, and (b) security experts. We intend to make available our annotated dataset to the community in order to facilitate further research in this space.

We summarize our key results below:

– **IKEA outperforms other state-of-the-art methods**. Specifically, IKEA exhibits more than 0.82 Mean Average Precision (**MAP**) and 0.85 Normalized Discounted Cumulative Gain (**NDCG**) in the top 50 retrieved keywords on average across three forums.

– **IKEA finds relevant jargon-focused keywords with up to 0.94 precision.** The flexible ranking empowers IKEA to exhibit relatively good precision. Interestingly, we find that 35% of these keywords are names of malware and virus, as we see in section *Experimental Results.*

– **IKEA works well as a query expansion method for documents.** Stepping away from forums, we use IKEA as a query expansion technique on the Fire 2011 document-query benchmark. We find that IKEA outperforms other state-of-the-art

Table 5.1: The basic statistics of our forums and Fire 2011 dataset. Fire consists of documents instead of posts.

	OffensiveComm.	HackThisSite	EthicalHackers	Fire
Posts	25,538	84,745	54,176	89,286
Threads	3,542	8,504	8,745	N.A
Words	45,119	47,810	48,157	551,075

methods with a MAP of 0.33-0.41.

The overarching vision is to provide a powerful, easy-to-use, and flexible method to provide domain-specific keyword expansion in an unsupervised way. We see our approach as a key capability within a practical tool-set for harnessing the wealth of information in online forums.

5.1 Background and Datasets

Our work focuses on security forums, but we also consider a document-based benchmark. We discuss our datasets below, and present their basic statistics in Table 5.1.

1. Security Forums. We have collected data from three different forums: OffensiveCommunity (*OC*), HackThisSite (*HT*) and EthicalHackers (*EH*). These forums bring together a wide range of users: system administrators, white-hat hackers, black-hat hackers, and users with variable skills, goals and intentions.

We briefly describe our three forums below.

a. OffensiveCommunity (OC): As the name suggests, this forum seems to contain information, which could enable hacking activities. It focuses on “offensive security”, which implies offensive activities, like detecting vulnerabilities, and breaking into systems.

Table 5.2: The list of our initial keyword sets.

Category	Identifier	List of keywords
Security	W_{MV}	Malware, Virus
	W_{HA}	Hack, Account
	W_{AV}	Attack, Vulnerability
Financial	W_{CC}	Credit, Card, Bank
	W_{SB}	Buy, Sell
Video Tutorial	W_{VG}	Video Game
	W_{TG}	Tutorial Guide
Jargon	J_{OC}	Darkcomet, Gingerbread
	J_{HT}	Morris, Slowloris
	J_{EH}	Chernobyl

There are many posts with instructions on how to compromise systems and advertisements of hacking tools and services.

b. HackThisSite (HT): The orientation of this forum appears to focus attacking techniques: how to hack vulnerable websites and systems and also related tutorials and services. At the same time, there are also general discussions around security practices and evaluation of methods and tools.

c. EthicalHackers (EH): As the name suggests, the forum seems to be a place for “white hat” hackers. There are many posts which are providing guidelines and tutorials for making systems more secure, and discussions about emerging threats. At the same time, there are many other discussions that seem to enable and nurture malicious activities.

2. Document benchmark: Fire 2011 (English). This is an annotated benchmark dataset for information retrieval purposes. It consists of documents from an English news agency, and 51 queries with the relevant documents.

Initial keywords: We evaluate the performance of our approach on the forums dataset with initial indicative sets of keywords as shown in Table 5.2. In practice, the keywords will be determined by the interests of the person using our approach, such as a security analyst. To ensure breadth, we use keyword sets that relate to different categories of queries as shown in Table 5.2. To stress test our approach, we focus on keyword sets with less than three words, which arguable makes the life of the user easier. Note, that we did experiment with three or more keywords, and the results were qualitatively similar with our approach performing well.

Table 5.3: Assessing the annotator agreement using the Fleiss-Kappa coefficient for each initial keyword set experiment.

Identifier	MTurk	Experts
W_{AV}	-	0.569
W_{HA}	-	0.535
W_{MV}	0.436	0.652
W_{CC}	0.444	0.511
W_{VG}	0.399	-
W_{TG}	0.677	-
W_{SB}	0.672	-
J_{Avg}	-	0.626

Establishing the groundtruth. Despite some recent efforts [40, 73], we were not able to find any benchmarks for online forums.

To establish the groundtruth, we use two group of annotators to evaluate the relevancy: (a) five experts in the security domain, and (b) five annotators from Amazon's Mechanical Turk platform (www.mturk.com). The annotators labeled the keywords based on the relevancy to the initial keyword set. In more detail, each word is labelled as relevant (*"a synonym, or a potential companion of the initial keywords in an English technical text"*). The final label is produced by using the majority vote approach. As expected, the Mechanical Turk annotations were of poor quality on the security related keyword sets as we discuss below. This happened despite setting high criteria to get only skilled annotators.

We also need groundtruth for assessing the ability of our approach in finding jargon-focused keywords. Here, we use security experts to label the retrieved keywords, since the subject and the context in this type of queries require even more technical expertise. In more detail, we provide our experts with the following context: (a) the keyword, (b) a post snippet that contains the keyword, and (c) top-ten google search results on the given keyword.

We assess our annotated data by using the Fleiss-Kappa coefficient on the two groups of annotators and we show the coefficient on average across three forums in Table 5.3. We see that there is good agreement as the Fleiss-Kappa coefficient is in the range of 0.399 and 0.677. Two queries, W_{MV} and W_{CC} , have been labeled by both groups of annotators and the expert annotators show the higher inter-agreement coefficient. We see a coefficient of 0.652 for experts versus 0.436 for MTurks in the case of W_{MV}. This suggests the need to use experts as the annotation tasks become more technical.

5.2 Overview of IKEA

Our approach provides a domain-specific keyword expansion consisting of four major steps, which we outline below.

Step 1: Domain representation. We represent words and posts of forums in an m-dimensional embedding space.

Step 2: Word-space expansion. We expand the initial set of keywords by adding relevant words iteratively.

Table 5.4: The symbol table with the key notations.

Symbol	Description
K_i	Initial set of keywords
K_s	Selected set of keywords in the word-based iterative process
K_f	Keywords appearing in a forum
P_i	Initial set of posts containing K_s
P_s	Selected set of posts after the iteration
P_f	Posts appearing in a forum
Z_k	Keywords similarity threshold
Z_p	Posts similarity threshold
K_e	Keywords extracted from P_s
Sim_K	Keywords similarity function
Sim_P	Posts similarity function
α_i	Ranking score weight for parameter i

Step 3: Post-space expansion. We identify posts that are similar to the set of posts, which contain the relevant words from the previous step.

Step 4: Result Processing. We extract and rank the keywords from the posts of the previous step, based on several metrics of importance and relevancy.

In the rest of the section, we discuss algorithmic aspects of the above steps, and we highlight their novelty. Note that an additional novelty is the combination of all these elements in an effective framework.

5.2.1 Step 1: Domain representation

We project words and posts in an appropriately-constructed m-dimensional space. Here, we use the Word2Vec approach [65] as a building block for doing this projection. We project every post on the same m-dimensional space by using the average of the projections of its words. There are other methods to project posts in an embedding space [58, 70], which we will evaluate in the future. Our current approach gives sufficiently good results.

5.2.2 Step 2 and 3: Two iterative expansions

We propose an iterative approach for establishing similarity between: (a) words, and (b) posts. This is part of our novelty, since, to the best of our knowledge, no prior work used such iterative approaches within an embedding representation.

a. Word-space Iterative Expansion. We expand the initial keyword K_i into set K_s adding similar keywords iteratively. In each step, we include words whose average similarity to any of the words in K_s is above a **threshold** $\mathrm{Z_k}$. This threshold takes values in the range [0-1], with lower values leading to more selected words. We repeat this process, until we cannot identify any more words for inclusion. In Figure 5.1, we depict this expansion as the transition between panel 1 and panel 2.

b. Post-space Iterative Expansion. As mentioned earlier, we identify the posts, P_i, that contain keywords from the set K_s. We then apply a similar iterative process to expand P_i into the P_s set of posts. In each step, we add more relevant posts to P_s and we stop, when no more posts can be added using the same threshold as above. This process is represented by Panel 3 and Panel 4 in Figure 5.1.

Why an iterative approach makes sense: In Figure 5.2, we show the intuition behind our choice of an iterative approach. We depict a word in the initial set with a “black star”. and find relevant words shown with “black diamonds”. The iterative process leads to a chain-like selection of similar keywords. This way, the selected words are ”very” close

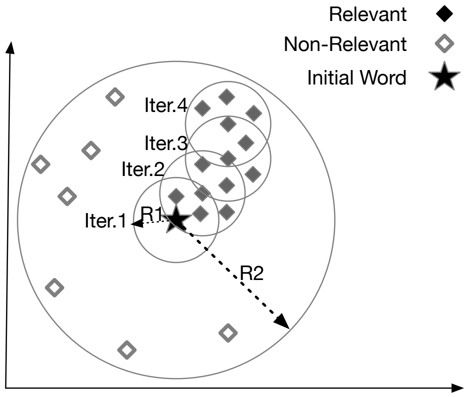


Figure 5.2: An intuitive explanation as to why the iterative approach gives better results.

to either the initial set of keywords, or words that were previously selected as similar.

Could we achieve the same by having a lower similarity threshold $\mathrm{Z_k}$? This would be equivalent to enlarging the radius of our "similarity" circle as shown in Figure 5.2. However, if we did that, we would run into the risk of including words that are typically far away from any of the initial or selected words.

This intuition is corroborated by our experiments. We vary the similarity threshold, $\mathrm{Z_k}$ and observe its effect on the quality of the keywords retrieved with the word-space iterative approach (focusing on the top 20 words) in Figure 5.3. We see that, by reducing the threshold $\mathrm{Z_k}$ from 0.95 to 0.8, we get 45% more irrelevant keywords in the word-space iterative expansion.

5.2.3 Step 4: Result Processing

Our approach introduces a processing stage to further refine the results in order to better respond to the user's needs. This is achieved with two main capabilities, which

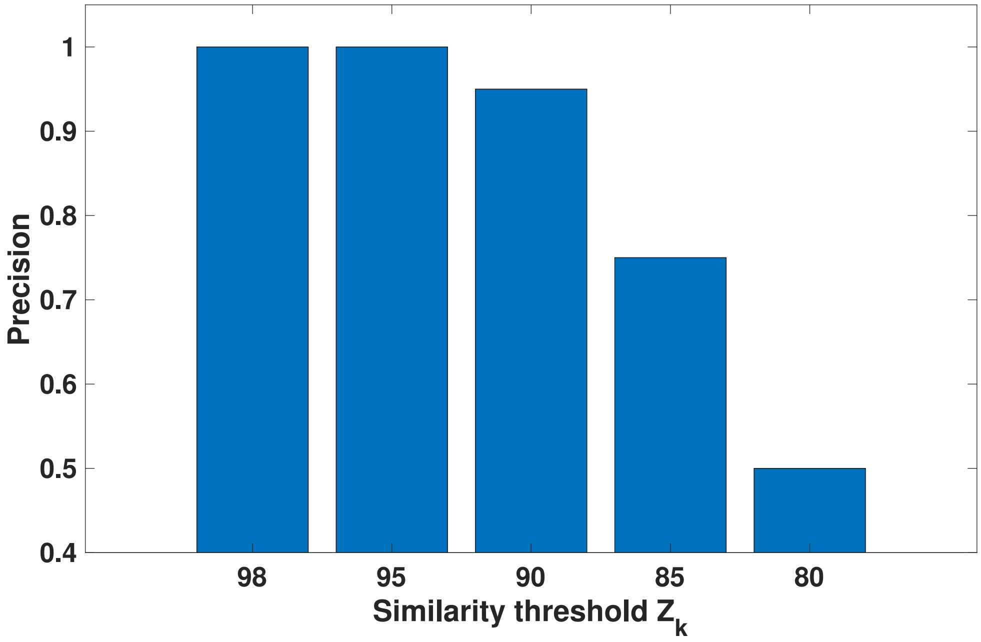


Figure 5.3: Precision of the top-20 retrieved words with the iterative approach in IKEA for different Z_k values.

we describe below.

(a) Filtering words: We have the ability to do an optional filtering on the words based on the user's needs. We can remove the keywords that appear in a dictionary or blacklists provided by the user. This provides the ability to remove words that the user knows are not of interest.

(b) Ranking words: An additional functionality is to rank the extracted keywords, in the order that is more likely to be of interest to the user. There are many different types of questions that the user may be trying to answer, and there are also various ways to rank words. As a proof of concept, we provide currently two options for ranking, as we discussed in the example in the introduction: a) **default**: where we rank words in terms of both popularity and their similarity to the words of interest, b) **jargon-focused:** where we prioritize ”jargon” words, such as names of malware, antivirus, and technical terms etc. We provide more details for our ranking below.

A. Metrics of importance and relevance. We use the following metrics to quantify aspects of the relevance, uniqueness and frequency of the words:

1. Word-Word similarity, $Sim_K(w, K_i)$**:** This functions captures the similarity of word w with the initial set of words provided by the user, K_i. We use the average cosine similarity between all words in K_i, which is a widely used metric in this space [33, 56].

2. Word-Post similarity, $Sim_P(w, P_s)$**:** This is a recently introduced metric, which captures the relevance and significance of keyword w to the posts that it appears [16]. Here, we use the metric to capture the “closeness” of word w to the set of posts P_s by calculating the average cosine similarity between word w and each post.

3. TFIDF, $TFIDF(w)$**:** Term Frequency-Inverse Document Frequency is a widely-used metric in information retrieval, which captures how important a word, w, is to a document within a collection of documents [75]. The intuition is that if a word is relatively rare overall, its appearance in a post is more significant compared to a word that appears in every post.

4. Inverse Document Frequency, $IDF(w)$**:** Inverse Document Frequency, $IDF(w)$ shows the reciprocal of frequency of posts containing word w in a corpus of documents, (the P_s set in our case). This metric measures the rarity of the word, and hence its discerning power and is widely used in this space [75].

Note that, TFIDF and IDF are metrics that capture the discerning power of a word given a set of documents/posts in a complementary way. IDF is particularly useful in the jargon-focused case.

B. Word ranking function. To rank the words, we combine the above metrics using a weighted function as follows in the default mode:

$$\begin{aligned} R_{Def}(w) = \alpha_1 * Sim_K(w, K_i) + \alpha_2 * Sim_P(w, P_s) + \\ + \alpha_3 * TFIDF(w) + \alpha_4 * IDF(w) \end{aligned} \tag{5.1}$$

where $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1$.

There are different ways to assign weights. Here, we use the rank exponent weight method [93], in which the normalized rank of each metric defines its weight. The weight α_{r_i} of rank r_i is given by:

$$\alpha_{r_i} = \frac{(N - r_i + 1)^\rho}{\sum_{k=1}^{N}(N - r_k + 1)^\rho} \tag{5.2}$$

where N=4. We find the best performance for ρ=2 experimentally. Intuitively, as we increase the value of ρ, we decrease the weight of the weight of the lower-ranked metrics. As we will see later, this approach seems to work well.

In the jargon-focused case (as defined earlier), we use the same function, but we change the order, and hence the α_i weights for each metric:

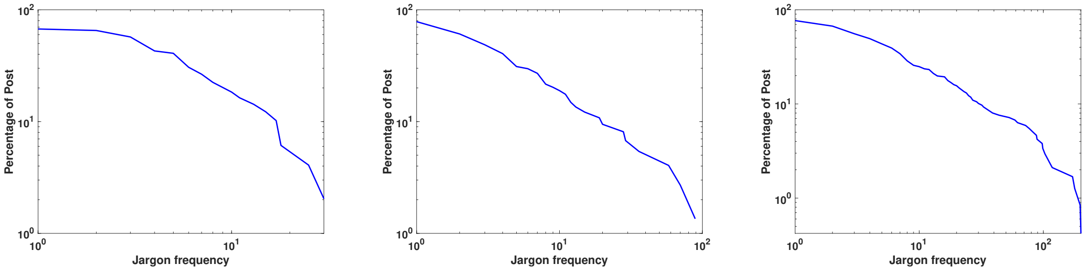
$$\begin{aligned} R_J(w) = \alpha_1 * IDF(w) + \alpha_2 * TFIDF(w) + \\ + \alpha_3 * Sim_K(w, K_i) + \alpha_4 * Sim_P(w, P_s) \end{aligned} \tag{5.3}$$

C. User preference: We enable the users to specify the preferred order of the keywords. We currently provide two options to the user:

a. Default preference: We use no filtering, and the ranking order in Eq. 5.1, which gives more weight to the similarity to the initial keywords.

b. Jargon-focused preference: We filter out English words, and use the ranking order in Eq. 5.3, to prioritize unusual words with highly discriminative power, which points us to jargon words.

Justification. Our choice of metrics and their order is motivated by two empirical observations. In Figure 5.4, we plot the CCDF of the number of jargon-focused keywords that appeared in the forums. Malware names are: (a) **rare**, as more than 63% of them appear in forums less than 6 times on average, averaging the results across the three forums, (b) **out-of-dictionary**, as we find that, in a random sample of 50 virus names from each forum, more than 88% of them are not in the dictionary.



(a) OffensiveCommunity (b) HackThisSite (c) EthicalHackers

Figure 5.4: CCDF of the frequency of the jargon based keywords (log-log scale).

D. Advanced user customization options. We have identified several opportunities to customize our approach. A sophisticated user can: (a) provide blacklists and dictionaries in the filtering stage, (b) introduce different ranking weights and functions. These are things that we will explore in the future as we discuss in section *Discussion.*

5.3 Experimental Results

In this section, we present the evaluation of our approach.

Data and queries. We use three security forums and 10 queries as we described in section *Background and Datasets.* When needed, we use a publicly available english dictionary (github.com/dwyl/english-words).

Defining our embedding space. We use a well-established skip-gram Word2Vec embedding approach [65]. First, we remove stop words, URLs and html tags and use Porter Stemmer for the stemming of words. In our Word2Vec model, we set the dimension of the embedding space for words and posts to 100. Experimentally, we opt for a value of 5 for the *training window size* parameter, which determines how much context around a word we consider during the training of the model [65]. In the iterative approach, we set the similarity threshold Z_k to 0.90.

Evaluation Metrics. We use the following metrics for evaluation: (a) precision, defined as the probability that an identified word is indeed relevant, (b) the mean average precision (MAP), when aggregating over multiple queries, and (c) the normalize discounted cumulative gain (NDCG). NDCG is a widely-used metric, which quantifies the quality of a ranking. We use the commonly-used paired t-test to evaluate the significance mean difference of the results [78, 33].

Reference methods. We evaluate our approach against other state-of-the-art methods for keyword expansion. We briefly describe them below, and we provide a detailed discussion and comparison with our approach in section 5.5.

Table 5.5: **Default**: Comparison of performance with mean average precision (MAP) and normalized discounted cumulative gain (NDCG). We use bold for the cases where IKEA exhibits the highest performance, and use the ”cross”(†) when a reference method matches that performance. The asterisk (∗) indicates the significance statistics using paired t-test with 95% confidence interval measure.

	IKEA		AQE		QM		
	MAP	NDCG	MAP	NDCG	MAP	NDCG	Datasest
@10	**0.957**	**0.968***	0.957†	0.951	0.957†	0.922	OffensiveCommunity
@20	**0.957***	**0.947***	0.871	0.926	0.843	0.896	
@30	**0.895***	**0.922***	0.833	0.899	0.767	0.889	
@40	**0.871***	**0.905**	0.811	0.903	0.746	0.879	
@50	**0.829***	0.891	0.803	**0.904**	0.714	0.882	
@10	**1***	**0.983***	0.971	0.965	0.957	0.895	HackThisSite
@20	**0.957***	**0.937***	0.929	0.924	0.9	0.902	
@30	**0.938**	**0.920**	0.928	0.918	0.866	0.901	
@40	**0.9**	0.893	0.89	**0.898**	0.818	0.876	
@50	**0.9**	**0.892**	0.90†	0.892†	0.814	0.883	
@10	**0.986***	**0.970***	0.971	0.951	0.571	0.642	EthicalHackers
@20	**0.9***	**0.921***	0.871	0.910	0.671	0.688	
@30	0.857	**0.885**	**0.861**	0.885†	0.695	0.699	
@40	**0.839**	**0.869**	0.839†	0.861	0.703	0.705	
@50	**0.826***	**0.858***	0.808	0.841	0.694	0.705	

- **Query expansion with maximum likelihood estimate(QM):** [56] This method uses Word2Vec embedding to find similar words to a given query in the word space language model.

- **Automatic Query Expansion (AQE):** [78] This method is a query expansion technique, where related words to a query are ranked using K-nearest neighbor approach.

- **Iterative thesaurus-based approach:** This is a straight-forward method, which returns synonyms of the initial keywords using a thesaurus capability of a dictionary (e.g. http://thesaurus.com). We obtain a list of synonyms to the initial keywords and expand them, similar to our iterative approach, until there are no new words added.

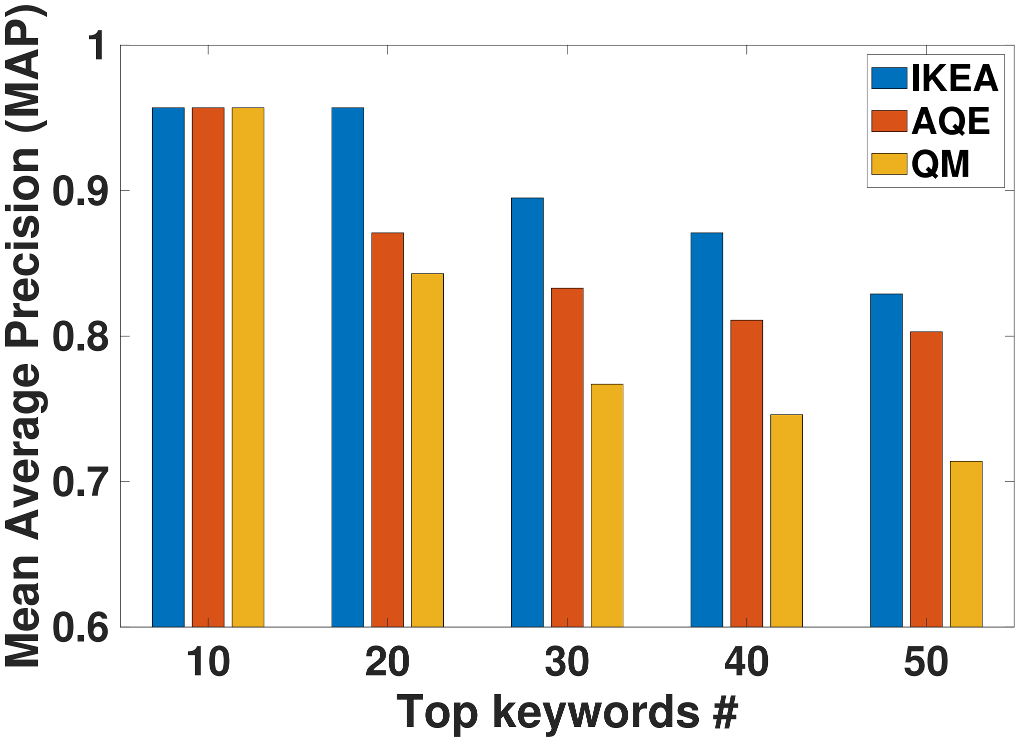


Figure 5.5: MAP comparison of IKEA and the reference algorithms in OffensiveCommunity forum for the top 10-50 retrieved keywords.

5.3.1 Default user preference

We show that IKEA outperforms the reference methods. In Table 5.5, we compare the three approaches using two metrics (MAP and NDCG) in our three forums. In our comparison, we vary the number of the top retrieved keywords: @10 means the top-10 keywords. In Table 5.6, we show the top-10 expanded keywords for the queries introduced in Table 5.2.

a. MAP: IKEA outperforms the competition in 14 out of 15 cases. As shown in Table 5.5, IKEA performs as well or better in the majority of the cases compared to the other methods with respect to MAP. We highlight (with a asterisk $*$) the cases where

Table 5.6: The expanded keyword sets with IKEA for our initial keyword sets. The underlined words are virus or malware names.

Initial	Retrieved top-10 ranked keywords
W_{MV}	malware, virus, trojan, infect, antivirus, malwarebytes, rootkit, worm, avg, spyware, investigate, keylogger
W_{HA}	hack, account, hotmail, facebook, twitter, gmail, bank, banking, learn, teach, instagram
W_{AV}	attack, vulnerability, phish, deface, defacement, ddos, dos, ftp, victim, credential, gmail, steal
W_{CC}	credit, card, bank, rfid, account, driver, wireless, sim, transfer, deposit, cash, subscription
W_{SB}	sell, buy, pay, cheaper, exchange, cash, tax, earn, purchase, reputation, fee, paypal
W_{VG}	video, game, youtube, play, vid, music, movies, watch, rpg, clip, mmorpg, fun
W_{TG}	tutorial, guide, tuts, beginner, noobie, teach, mentor, tip, help, recommend, explain, advice
J_{OC}	darkcomet, gingerbread, smp, hideman, cwm, msfconsole, adb, battlefield, casperspy, uniscan, urllib, fadias
J_{HT}	morris, slowloris, ugand, revolutinaryg, imf, lov, knoppix, openvms, teabag, joli, virtuawin
J_{EH}	chernobyl, zeroaccess, athcon, mersenne, duronio, dubuque, crypter, rustock, maricopa, wua, pornography, Gaobot

the paired t-test indicates statistically significant performance difference.

IKEA exhibits good performance: MAP is more than 0.826 across all the different experiments. In other words, IKEA returns a relevant word 82.6% of the time. We compare the MAP of the three methods in the OffensiveCommunity forum in Figure 5.5. We show the MAP of the three approaches for the top 10-50 retrieved keywords. We see that IKEA significantly outperforms the other two methods in the case of the top 20, 30, 40, and 50 retrieved keywords.

b. NDCG: IKEA outperforms the competition in 13 out 15 cases. Focusing on NDCG, IKEA again performs at least as well as the competition in 13 experiments. Furthermore, if we focus on the top-10 and top-20 extracted keywords, IKEA is better than the others method among all forums. Recall that NDCG is an indication of the ranking quality: a high NDCG value indicates superior ranking quality with the most relevant words near the top.

c. Thesaurus-based approach identifies only 16% words reported by IKEA. A natural question is: Does IKEA add more value than a simple thesaurus search? The answer is yes. In Figure 5.6, we plot the percentage of common keywords retrieved

Table 5.7: **Jargon-focused**: Precision of the top 50 and 100 ranked keywords.

	IKEA	R-Sim	R-Freq	Forum
@50	**0.62**	0.56	0.4	OffensiveCommunity
@100	**0.584**	0.495	0.426	
@50	**0.50**	0.45	0.2	HackThisSite
@100	**0.41**	0.38	0.36	
@50	**0.941**	0.902	0.902	EthicalHackers
@100	**0.901**	0.851	0.871	

by IKEA, AQE and QM compared to the thesaurus-based approach for the top 100 words. The thesaurus-based approach finds at most 16% of the keywords retrieved by IKEA and AQE, and much less ($\leq$6%) by QM.

We attribute the difference to the fact that a thesaurus-based approach is not domain-specific, but relies on word similarity broadly-defined at the dictionary level.

5.3.2 Jargon-focused user preference

We evaluate IKEA in the jargon-focused case.

IKEA finds jargon words with up to 0.94 precision. We show the performance of IKEA in the jargon-focused case in Table 5.7 for the top 50 and top 100 keywords. Further, we want to evaluate our combined four-metric ranking compared to using only one metric. We consider two alternative ranking functions using the first metric (with the most weight) from Eq. 5.1 and 5.3: (a) $R_{Sim}(w) = Sim_K(w, K_i)$, and (b) $R_{Freq}(w) = IDF(w)$. We see that our combination of the metrics, gives better results compared to single metric rankings.

Identifying emerging malware is a key concern in the security world, while jargon can include other technical terms. Upon manual investigation, we find that 35% of the

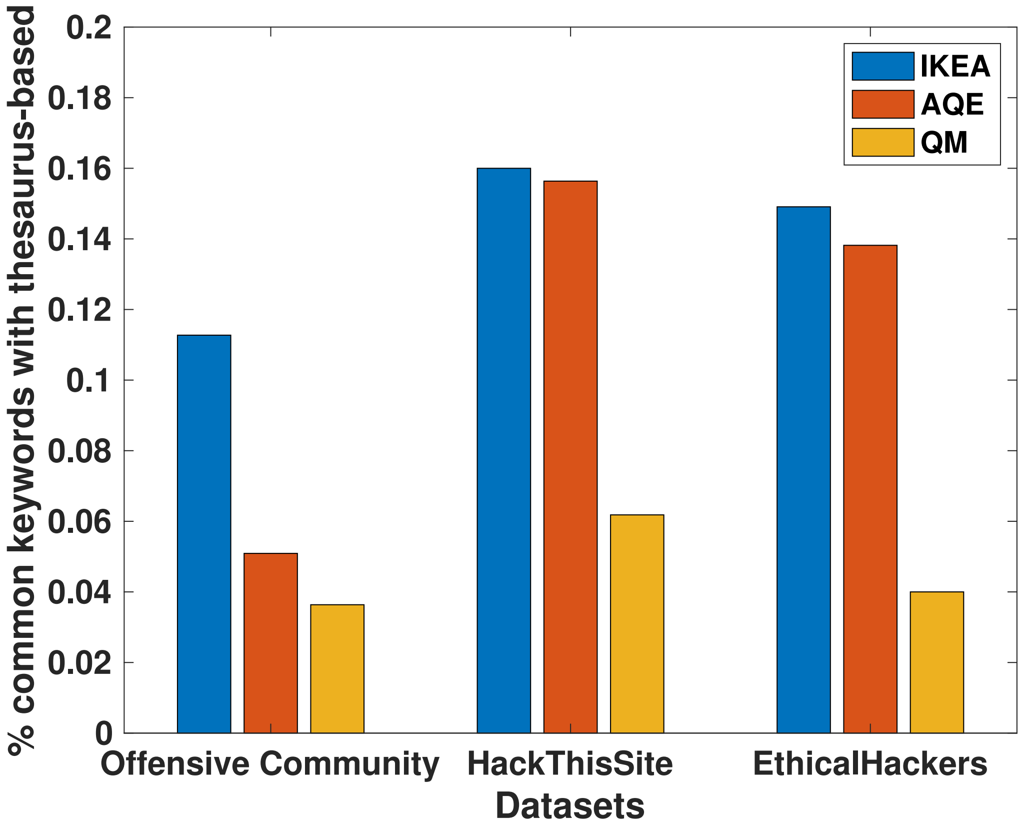


Figure 5.6: The percentage of common words between the retrieved keyword from the three methods and the thesaurus-based approach in our forums.

retrieved words with our approach are indeed malware and virus names.

5.3.3 IKEA: query expansion on the Fire dataset

As a case study, we evaluate our method within the context of informational retrieval focusing on the following problem. Given a set of documents and a query in natural language (NL), we want to find the related documents to that query. We use IKEA as a building block: (a) we extract keywords from the query, (b) we use IKEA, and (c) we used Lucene search engine for document retrieval and rank them accordingly [78]. To compare, we repeat the process replacing IKEA with the other two methods, AQM and QE, in the second step.

Table 5.8: **Query expansion**: Evaluating IKEA on the Fire dataset.

	IKEA		AQE		QM		No-Expansion	
	MAP	NDCG	MAP	NDCG	MAP	NDCG	MAP	NDCG
@10	**0.362**	**0.799***	**0.362**	0.713	0.261	0.637	0.203	0.483
@20	**0.358***	0.665	0.318	**0.681**	0.223	0.611	0.188	0.337
@30	**0.346***	0.598	0.326	**0.604**	0.236	0.529	0.188	0.261
@40	**0.335**	**0.579**	0.328	0.577	0.223	0.529	0.183	0.249
@50	**0.412***	**0.571***	0.268	0.555	0.162	0.529	0.177	0.249

IKEA outperforms or matches the other state-of-the-art methods as a building block in the document retrieval framework outlined above. We apply the three approaches to the set of queries in the Fire 2011 dataset, which we described in section *Background and Datasets.* In Table 5.8, we show the average over 10 sample queries in Fire 2011 using both precision (MAP) and ranking quality (NDCG).

5.4 Discussion

We discuss how IKEA could be used in practice, and touch upon its limitations and potential future directions.

A. IKEA as a practical tool. We argue that our approach could form an easy to use, yet powerful, tool that a user can customize for their needs. We intend to make our tool available as a tool to the research community.

a. Initial keywords: Minimal knowledge required. As we saw, our approach can work well with as little as two initial keywords, which in some cases, can be as much as a user starts with. With more keywords, user can only provide more context and help the approach narrow down fast on relevant keywords to a particular concept or topic.

b. Ready to use: no tuning required. Our approach comes with default values, which makes it easy to use even by a non-expert user or a user that explores a domain or a dataset for the first time.

c. Customization is possible. Our algorithm provides a sufficient number of parameters that can be used to finetune the results. We anticipate that this will be useful to: (a) an experienced user, or (b) a novice user in a trial-error exploration of a domain. We will provide easy to use interfaces or APIs for users to control several parameters, such as the similarity threshold Z_k, or the prioritization function for the ranking of the results, as we discuss below.

d. Result prioritization. We propose a method to prioritize the retrieved keywords, as we saw in section *Result Processing*. However, one can consider several ways to combine the different metrics of the keywords. In the final version of our tool, we intend to provide several flexible ways with which a user can rank the retrieved keywords.

B. A case-study of IKEA with a political focus. The default approach in IKEA can help an analyst identify the relevant keywords for a topic of interest. Let us consider a particular example outside the scope of security forums: an analyst wants to mine information regarding President Trump's impeachment from an online forum like Reddit.

The first step is to identify the right keywords to extract relevant posts. The analyst can easily provide a few initial keywords like "Trump" and "impeachment". At this point, IKEA is invoked to identify other similar keywords.

We have tested such scenario in collected data from Reddit and IKEA expands the two initial keywords by identifying related keywords. Specifically, the top ten identified keywords was the following set:

[trump, impeachment, acquit, formal, inquiry, nanci, nancy, pelosi, trial, convict]

It is interesting to note the prominent mispelling of “nanci”. These words are not obvious to someone without broader political context. We argue that this toy example provides an intuitive showcase of the value of our approach outside the scope of security.

C. Issues and limitations. We briefly discuss some potential limitations of our approach, which can also be seen as opportunities to further enhance the method.

How generalizable is the method? Our method was motivated by the analysis of online forums. We tested it on security forums, but it should work well with other online forums, as the initial results on Reddit and the Fire dataset suggest. However, we have not tried our approach to highly technical documents, such as legal or medical literature. Overall, we believe that our approach could apply to a fairly wide range of media and documents, which we will explore in the future.

For what type of keyword expansions does this work? This is a common question for any kind of keyword expansion method, as there is an unlimited number of questions and intentions behind a query. In our work, we tried to address this by considering various categories of questions as we saw in Table 5.2, the Fire dataset and our political-focused case-study. Clearly, there is a wide spectrum between broad or narrow queries (i.e. if the query is looking for specific information versus for general themes and topics). We argue that the number of initial keywords can play a factor in enabling a query to hone in on a narrow concept. In other words, how successful our approach can be for different types of queries could be affected by the number of initial keywords that are “necessary” for each type of query.

5.5 Related Work

We summarize related work in the following general areas.

a. Embedding approaches in query expansion. Several recent efforts leverage embedding approaches in extending a query, usually for structured documents, such as news reports [33, 57, 78].

We discuss the two most relevant studies and compare them with our approach. The QM approach [33, 56] uses embedding to identify similar words in the word-space only. The AQE approach [78] uses similarity expansion in the word-space selected from sudo relevant documents and they use a nearest neighbor technique to rank keywords. They differ from our work in that they do: (a) not use an iterative expansion in word and post domain (QM and AQE), (b) not do a similarity expansion in the post-space (QM), (c) not have a flexible ranking capability (QM and AQE). We argue that combination of all the above elements contributes to the superior performance and flexibility of our approach.

Following a different path, some methods rely on user-behavior and feedback to establish a statistical model of the word relevancy [10, 57].

b. No initial keyword set: extracting topics and keywords. Though related, these works address a fairly different problem than we do here: the goal is to identify the “important” words in a document without an initial keyword set. Several works identify keywords or key-phrases that capture the topic of a document [16, 105, 34].

Named-entity extraction is a tangentially related problem to our jargon-focused case with several supervised [62, 89] and unsupervised approaches [36, 69], which usually do not assume an initial keyword set.

c. NLP-based techniques for extracting specific information. A group of recent studies focus on retrieving specific information, such as: (a) prices and availability of malicious services in security forums [74, 68]; (b) extracting malicious IP addresses and discussions of interest in security forums [40]. A very recent work [44] uses embedding techniques to identify and classify threads of interest from an online forum, which is a related, but different problem.

Some research efforts use embedding techniques to identify vulnerabilities and study the evolution of cyber-security attacks [92, 97] using security and CVE reports, and also web-blogs and databases from the darkweb.

5.6 Conclusion

We propose an iterative keyword expansion approach (IKEA) based on embedding to identify keywords relevant to an initial set of keywords. The novelty of our approach is three-fold: (a) we use two similarity expansions, in the word-word and post-post spaces, (b) we use two iterative approaches for identifying similar words and posts, and (c) we provide a flexible processing that empowers the user to finetune its outcome.

We evaluate our method with real data and we show that: (a) our approach works well with a MAP above 0.82 and NDCG above 0.85 on average, (b) IKEA outperforms the state-of-art approaches in almost all cases and often with significant difference, and (c) IKEA exhibits superior performance as a component in a query expansion task using the Fire dataset.

We see our approach as an effective building block in the space of information retrieval. We intend to fully explore its capabilities in a wide range of: (a) queries for different user preferences, beyond the two we saw here, and (b) types of data and domains. For the latter, the nature of the data (e.g. legal forum or medical journals) can introduce both new challenges and opportunities in identifying the right keywords.

Chapter 6

Conclusions

In conclusion, in this thesis, we proposed a comprehensive set of solutions to harness the information in security forums. We proposed three approaches to extract useful information from security forums. The work consists of three main efforts: (a) we proposed a method to automatically identify malicious IP addresses; (b) we developed a systematic method to identify and classify user-specified threads of interest into different categories; and (c) we presented an iterative approach to expand the initial keywords of interest which are essential feeds in searching and retrieving information. We argue that our work provides fundamental novel and effective capabilities in analyzing and extracting information from security forums.

Our future plans include: (a) studying more entities and extracting more useful information in such security forums, (b) exploring the limits of proposed approaches in more detail, and (c) exploring more security forums. An overarching new direction is to apply the proposed work in a variety of other forums appropriately adjusted for their idiosyncrasies.

Bibliography

[1] Ashiyane. `http://www.ashiyane.org/forums/`.

[2] Geolite. `http://dev.maxmind.com/geoip/legacy/geolite/`.

[3] Nitol-botnet. `https://threatpost.com/tag/nitol-botnet/`.

[4] Offensive community. `http://www.offensivecommunity.net`.

[5] Virustotal. `http://www.virustotal.com`.

[6] Wilders security. `http://www.wilderssecurity.com`.

[7] A. Abbasi, W. Li, V. Benjamin, S. Hu, and H. Chen. Descriptive analytics: Examining expert hackers in web forums. In *2014 IEEE Joint Intelligence and Security Informatics Conference*, pages 56–63, Sep. 2014.

[8] Charu C. Aggarwal and ChengXiang Zhai. *A survey of text classification algorithms*, pages 163–222. Springer US, 8 2013.

[9] Luca Allodi. Economic factors of vulnerability trade and exploitation. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 1483–1499, New York, NY, USA, 2017. Association for Computing Machinery.

[10] Mohannad ALMasri, Catherine Berrut, and Jean-Pierre Chevallet. A comparison of deep learning based query expansion with pseudo-relevance feedback and mutual information. In *ECIR*, volume 9626, pages 709–715, 03 2016.

[11] Tim Althoff, Pranav Jindal, and Jure Leskovec. Online actions with offline impact: How online social networks influence online and offline user behavior. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pages 537–546, New York, NY, USA, 2017. ACM.

[12] Rie Kubota Ando and Tong Zhang. A high-performance semi-supervised learning method for text chunking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 1–9, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[13] Slobodan Beliga. Keyword extraction: a review of methods and approaches. *University of Rijeka, Department of Informatics, Rijeka*, pages 1–9, 2014.

[14] V. Benjamin and H. Chen. Securing cyberspace: Identifying key actors in hacker communities. In *2012 IEEE International Conference on Intelligence and Security Informatics*, pages 24–29, June 2012.

[15] V. Benjamin, W. Li, T. Holt, and H. Chen. Exploring threats and vulnerabilities in hacker web: Forums, irc and carding shops. In *2015 IEEE International Conference on Intelligence and Security Informatics (ISI)*, ISI '15, pages 85–90, May 2015.

[16] Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. Simple unsupervised keyphrase extraction using sentence embeddings. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 221–229, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[17] C. Blanco, J. Lasheras, R. Valencia-García, E. Fernández-Medina, A. Toval, and M. Piattini. A systematic review and comparison of security ontologies. In *2008 Third International Conference on Availability, Reliability and Security*, pages 813–820, March 2008.

[18] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 120–128, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[19] Robert A. Bridges, Corinne L. Jones, Michael D. Iannacone, and John R. Goodall. Automatic labeling for entity extraction in cyber security. *CoRR*, abs/1308.4941, 2013.

[20] Robert A. Bridges, Corinne L. Jones, Michael D. Iannacone, and John R. Goodall. Automatic labeling for entity extraction in cyber security. *CoRR*, abs/1308.4941, 2013.

[21] Emily Chen, Kristina Lerman, and Emilio Ferrara. Covid-19: The first public coronavirus twitter dataset, 2020.

[22] Minmin Chen, Kilian Q. Weinberger, and John C. Blitzer. Co-training for domain adaptation. NIPS'11, pages 2456–2464, USA, 2011.

[23] J. Cheng, M. Bernstein, C. Danescu-Niculescu-Mizil, and J. Leskovec. Anyone Can Become a Troll: Causes of Trolling Behavior in Online Discussions. *ArXiv e-prints*, February 2017.

[24] Gao Cong, Long Wang, Chin-Yew Lin, Young-In Song, and Yueheng Sun. Finding question-answer pairs from online forums. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 467–474, New York, NY, USA, 2008. ACM.

[25] Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Co-clustering based classification for out-of-domain documents. KDD '07, pages 210–219, USA, 2007.

[26] Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Transferring naive bayes classifiers for text classification. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 1*, AAAI'07, pages 540–545. AAAI Press, 2007.

[27] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. ICML '07, pages 193–200, New York, NY, USA, 2007.

[28] Rajarshi Das, Manzil Zaheer, and Chris Dyer. Gaussian lda for topic models with word embeddings. ACL'15, 2015.

[29] Hal Daume III. Frustratingly easy domain adaptation. ACL '07, 2007.

[30] Ashok Deb, Kristina Lerman, Emilio Ferrara, Ashok Deb, Kristina Lerman, and Emilio Ferrara. Predicting Cyber-Events by Leveraging Hacker Sentiment. *Information*, 9(11):280, nov 2018.

[31] P. Devineni, D. Koutra, M. Faloutsos, and C. Faloutsos. If walls could talk: Patterns and anomalies in facebook wallposts. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 367–374, 2015.

[32] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[33] Fernando Diaz, Bhaskar Mitra, and Nick Craswell. Query expansion with locally-trained word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 367–377, Berlin, Germany, August 2016. Association for Computational Linguistics.

[34] Adji B Dieng, Francisco J R Ruiz, and David M Blei. Topic modeling in embedding spaces. *arXiv preprint arXiv:1907.04907*, 2019.

[35] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996.

[36] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artif. Intell.*, 165(1):91–134, June 2005.

[37] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. ACL '05, pages 363–370, Stroudsburg, PA, USA, 2005.

[38] Corina Florescu and Cornelia Caragea. PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1115, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[39] R. Frank, M. Macdonald, and B. Monk. Location, location, location: Mapping potential canadian targets in online hacker discussion forums. EISIC '16, 2016.

[40] Joobin Gharibshah, Evangelos E. Papalexakis, and Michalis Faloutsos. Ripex: Extracting malicious ip addresses from security forums using cross-forum learning. In *PAKDD'18*. Springer International Publishing, 2018.

[41] Joobin Gharibshah, Zhabiz Gharibshah, Evangelos E. Papalexakis, and Michalis Faloutsos. An empirical study of malicious threads in security forums. In *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, page 176–182, New York, NY, USA, 2019. Association for Computing Machinery.

[42] Joobin Gharibshah, Tai Li, Maria Vanrell, Andre Castro, Konstantinos Pelechrinis, Evangelos Papalexakis, and Michalis Faloutsos. InferIP: Extracting actionable information from security discussion forums. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, ASONAM '17, pages 301–304, New York, NY, USA, 2017. ACM.

[43] Joobin Gharibshah, Tai Ching Li, Andre Castro, Konstantinos Pelechrinis, Evangelos E. Papalexakis, and Michalis Faloutsos. Mining actionable information from security forums: The case of malicious ip addresses. *From Security to Community Detection in Social Networking Platforms*, 2019.

[44] Joobin Gharibshah, Evangelos E. Papalexakis, and Michalis Faloutsos. Rest: A thread embedding approach for identifying and classifying user-specified information in security forums. *Proceedings of the International AAAI Conference on Web and Social Media*, 14(1):217–228, May 2020.

[45] H. Hang, A. Bashir, M. Faloutsos, C. Faloutsos, and T. Dumitras. "Infect-me-not": A user-centric and site-centric study of web-based malware. In *IFIP Networking*, pages 234–242, May 2016.

[46] Christopher R. Harshaw, Robert A. Bridges, Michael D. Iannacone, Joel W. Reed, and John R. Goodall. Graphprints: Towards a graph analytic method for network anomaly detection. In *Proceedings of the 11th Annual Cyber and Information Security Research Conference*, CISRC '16, pages 15:1–15:4, New York, NY, USA, 2016. ACM.

[47] Thomas J Holt, Deborah Strumsky, Olga Smirnova, and Max Kilger. Examining the social networks of malware writers and hackers. 6(1):891–903, 2012.

[48] Michael Iannacone, Shawn Bohn, Grant Nakamura, John Gerth, Kelly Huffer, Robert Bridges, Erik Ferragut, and John Goodall. Developing an ontology for cyber security knowledge graphs. CISR '15, pages 12:1–12:4, New York, NY, USA, 2015.

[49] Jing Jiang and Chengxiang Zhai. Instance weighting for domain adaptation in nlp. In *In ACL 2007*, pages 264–271, 2007.

[50] Peng Jin, Zhang Yue, Xingyuan Chen, and Yunqing Xia. Bag-of-embeddings for text classification. *IJCAI International Joint Conference on Artificial Intelligence*, 2016-Janua:2824–2830, 2016.

[51] Anjali Jivani. A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl.*, 2:1930–1938, 11 2011.

[52] Corinne L. Jones, Robert A. Bridges, Kelly M. T. Huffer, and John R. Goodall. Towards a relation extraction framework for cyber-security concepts. CISR '15, pages 11:1–11:4, 2015.

[53] Corinne L. Jones, Robert A. Bridges, Kelly M. T. Huffer, and John R. Goodall. Towards a relation extraction framework for cyber-security concepts. *CoRR*, abs/1504.04317, 2015.

[54] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics, 2017.

[55] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, March 1998.

[56] Saar Kuzi, Anna Shtok, and Oren Kurland. Query expansion using word embeddings. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 1929–1932, New York, NY, USA, 2016. ACM.

[57] Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, page 120–127, New York, NY, USA, 2001. Association for Computing Machinery.

[58] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1188–II–1196. JMLR.org, 2014.

[59] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788, oct 1999.

[60] Tai Ching Li, Joobin Gharibshah, Evangelos E. Papalexakis, and Michalis. Faloutsos. Trollspot: Detecting misbehavior in commenting platforms. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, ASONAM '17, 2017.

[61] W. Li and H. Chen. Identifying top sellers in underground economy using deep learning-based sentiment analysis. In *2014 IEEE Joint Intelligence and Security Informatics Conference*, pages 64–67, Sep. 2014.

[62] Ying Luo, Hai Zhao, and Junlang Zhan. Named entity recognition only from word embeddings. 2019.

[63] E. Marin, J. Shakarian, and P. Shakarian. Mining key-hackers on darkweb forums. In *2018 1st International Conference on Data Intelligence and Security (ICDIS)*, pages 73–80, April 2018.

[64] Andrew McCallum and Kamal Nigam. A comparison of event models for naive bayes text classification, 1998.

[65] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[66] Brad Miller, Alex Kantchelian, Sadia Afroz, Rekha Bachwani, Edwin Dauber, Ling Huang, Michael Carl Tschantz, Anthony D. Joseph, and J.D. Tygar. Adversarial active learning. In *Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop*, AISec '14, pages 3–14, New York, NY, USA, 2014. ACM.

[67] Marti Motoyama, Damon McCoy, Kirill Levchenko, Stefan Savage, and Geoffrey M. Voelker. An analysis of underground forums. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, pages 71–80, New York, NY, USA, 2011. ACM.

[68] Marti Motoyama, Damon McCoy, Kirill Levchenko, Stefan Savage, and Geoffrey M. Voelker. An analysis of underground forums. IMC '11, pages 71–80, New York, NY, USA, 2011.

[69] David Nadeau, Peter D. Turney, and Stan Matwin. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. pages 266–277, 2006.

[70] Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, NAACL'18, pages 528–540, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[71] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[72] Evangelos E Papalexakis, Nicholas D Sidiropoulos, and Rasmus Bro. From k-means to higher-way co-clustering: Multilinear decomposition with sparse latent factors. *IEEE transactions on signal processing*, 61(2):493–506, 2013.

[73] Sergio Pastrana, Daniel R. Thomas, Alice Hutchings, and Richard Clayton. Crimebb: Enabling cybercrime research on underground forums at scale. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 1845–1854, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee.

[74] Rebecca S. Portnoff, Sadia Afroz, Greg Durrett, Jonathan K. Kummerfeld, Taylor Berg-Kirkpatrick, Damon McCoy, Kirill Levchenko, and Vern Paxson. Tools for automated analysis of cybercriminal markets. WWW '17, 2017.

[75] Anand Rajaraman and Jeffrey Jure Leskovec, Jure Ullman D. *Data Mining*, page 1–17. Cambridge University Press, 2011.

[76] Juan Ramos. Using TF-IDF to determine word relevance in document queries. ICML '03, 2003.

[77] R. Rawassizadeh, E. Momeni, C. Dobbins, J. Gharibshah, and M. Pazzani. Scalable daily human behavioral pattern mining from multivariate temporal data. *IEEE Transactions on Knowledge and Data Engineering*, 28(11):3098–3112, 2016.

[78] Dwaipayan Roy, Debjyoti Paul, Mandar Mitra, and Utpal Garain. Using word embeddings for automatic query expansion. *ArXiv*, abs/1606.07608, 2016.

[79] Carl Sabottke, Octavian Suciu, and Tudor Dumitras. Vulnerability disclosure in the age of social media: Exploiting twitter for predicting real-world exploits. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 1041–1056, Washington, D.C., 2015. USENIX Association.

[80] S. Samtani, R. Chinn, and H. Chen. Exploring hacker assets in underground forums. In *IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 31–36, May 2015.

[81] Sagar Samtani, Ryan Chinn, and Hsinchun Chen. In *2015 IEEE International Conference on Intelligence and Security Informatics: Securing the World through an Alignment of Technology, Intelligence, Humans and Organizations, ISI 2015*, pages 31–36. Institute of Electrical and Electronics Engineers Inc., jul 2015. 13th IEEE International Conference on Intelligence and Security Informatics, ISI 2015 ; Conference date: 27-05-2015 Through 29-05-2015.

[82] Enrico Santus, Hongmin Wang, Emmanuele Chersoni, and Yue Zhang. A Rank-Based Similarity Metric for Word Embeddings. pages 552–557, 2018.

[83] A. Sapienza, A. Bessi, S. Damodaran, P. Shakarian, K. Lerman, and E. Ferrara. Early warnings of cyber threats in online discussions. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 667–674, Nov 2017.

[84] Anna Sapienza, Sindhu Kiranmai Ernala, Alessandro Bessi, Kristina Lerman, and Emilio Ferrara. Discover: Mining online chatter for emerging cyber threats. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pages 983–990,

Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee.

[85] Harrisen Scells and Guido Zuccon. Generating better queries for systematic reviews. In *The 41st International ACM SIGIR Conference on Research &#38; Development in Information Retrieval*, SIGIR '18, pages 475–484, New York, NY, USA, 2018. ACM.

[86] D. Sculley, Matthew Eric Otey, Michael Pohl, Bridget Spitznagel, John Hainsworth, and Yunkai Zhou. Detecting adversarial advertisements in the wild. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 274–282, New York, NY, USA, 2011. ACM.

[87] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[88] Jana Shakarian, Andrew T. Gunn, and Paulo Shakarian. *Exploring Malicious Hacker Forums*, pages 259–282. Springer International Publishing, Cham, 2016.

[89] Jingbo Shang, Liyuan Liu, Xiang Ren, Xiaotao Gu, Teng Ren, and Jiawei Han. Learning named entity tagger using domain-specific dictionary. In *EMNLP*, 2018.

[90] Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 440–450. Association for Computational Linguistics, 2018.

[91] Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) ACL'18*, pages 440–450. Association for Computational Linguistics, 2018.

[92] Yun Shen et al. Attack2vec: Leveraging temporal word embeddings to understand the evolution of cyberattacks. In *28th USENIX Security Symposium (USENIX Security 19)*, USENIX Security'19, pages 905–921, Santa Clara, CA, August 2019. USENIX Association.

[93] William G. Stillwell, David A. Seaver, and Ward Edwards. A comparison of weight approximation techniques in multiattribute utility decision making. *Organizational Behavior and Human Performance*, 28(1):62 – 77, 1981.

[94] Brett Stone-Gross et al. The underground economy of spam: A botmaster's perspective of coordinating large-scale spam campaigns. LEET'11, pages 4–4, 2011.

[95] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification *. Technical report.

[96] Jian Tang, Meng Qu, and Qiaozhu Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1165–1174, New York, NY, USA, 2015. ACM.

[97] Nazgol Tavabi, Palash Goyal, Mohammed Almukaynizi, Paulo Shakarian, and Kristina Lerman. Darkembed: Exploit prediction with neural language models. In *AAAI*, pages 7849–7854. AAAI Press, 2018.

[98] Johan Ugander, Brian Karrer, Lars Backstrom, and Cameron Marlow. The anatomy of the facebook social graph. *CoRR*, abs/1111.4503, 2011.

[99] K. Veeramachaneni, I. Arnaldo, V. Korrapati, C. Bassias, and K. Li. ai^2: Training a big data machine to defend. In *2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity)*, pages 49–54, April 2016.

[100] Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. Joint embedding of words and labels for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2321–2331, 2018.

[101] Qiu-Hong Wang, Wei T. Yue, and Kai-Lung Hui. *Do Hacker Forums Contribute to Security Attacks?*, pages 143–152. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[102] Shuai Wang, Zhiyuan Chen, Bing Liu, and Sherry Emery. Identifying Search Keywords for Finding Relevant Social Media Posts. *Proceedings of the Thirthieth AAAI Conference on Artificial Intelligence (AAAI-16)*, pages 3052–3058, 2016.

[103] Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3(1):9, May 2016.

[104] Weidi Xu and Ying Tan. Semi-supervised Target-level Sentiment Analysis via Variational Autoencoder. 2018.

[105] Guangxu Xun, Yaliang Li, Wayne Xin Zhao, Jing Gao, and Aidong Zhang. A correlated topic model using word embeddings. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 4207–4213. AAAI Press, 2017.

[106] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[107] Xin Ye, Hui Shen, Xiao Ma, Razvan Bunescu, and Chang Liu. From word embeddings to document similarities for improved information retrieval in software engineering. *Proceedings of the 38th International Conference on Software Engineering - ICSE '16*, pages 404–415, 2016.

[108] Hamed Zamani and W. Bruce Croft. Relevance-based word embedding. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 505–514, New York, NY, USA, 2017. ACM.

[109] Amy X. Zhang, Bryan Culbertson, and Praveen Paritosh. Characterizing Online Discussion Using Coarse Discourse Sequences. *Proceedings of the International Conference on Weblogs and Social Media*, pages 357–366, 2017.

[110] Xiong Zhang, Alex Tsang, Wei T. Yue, and Michael Chau. The classification of hackers by knowledge exchange behaviors. *Information Systems Frontiers*, 17(6):1239–1251, December 2015.

[111] Xiong Zhang, Alex Tsang, Wei T. Yue, and Michael Chau. The classification of hackers by knowledge exchange behaviors. *Info. Systems Frontiers*, 17(6):1239–1251, 2015.