Extracting actionable information from Security Forums

Joobin Gharibshah
jghar002@ucr.edu
University of California Riverside
Riverside, California

ABSTRACT

The goal of this work is to systematically extract information from hacker 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 the problem of analyzing text content in security forums. A key novelty is that we use user profiles and contextual features along with transfer learning approach and also embedding space to help us identify and refine information that we could not get from security forum with trivial analysis. We collect a wealth of data from 5 different security forums. The contribution of our work is twofold; (a) we develop a method to automatically identify through the forums malicious IP addresses (b) we also propose a systematic method to identify and classify user-specified threads of interest into four categories. We further showcase how this information can inform knowledge extraction from the forums. As the cyber-wars are becoming more intense, having early accesses to useful information becomes more imperative to remove the hackers first-move advantage, and our work is a solid step towards this direction.

KEYWORDS

Security forums, Text analysis, Word embedding, Transfer learning

ACM Reference Format:
Joobin Gharibshah. 2019. Extracting actionable information from Security Forums. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW '19 Companion), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3308560.3314197

1 INTRODUCTION

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. 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 potential malicious content such as IP addresses and threads of interest for the security analysts. In this research we are looking to analyze the text in order to identify and classify content of interests. Twofold of the interest here are IP addresses and user-specified security discussion threads. For example threads describing hacking tutorials or announcing emerging attacks.

In the first fold we identify and characterize IP addresses mentioned in text of security forums. 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.

Interestingly as we showed in our first research [4, 5], not all of the reported IP addresses are malicious, which makes the classification necessary. 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 its key novelty, our approach by utilizing a simple transfer learning technique, 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 [6].

In the second fold, we propose 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. Arguably, providing keywords is a relatively easy task for the user. 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. We use the term Identification to refer to this problem. A key challenge here is how to create a robust solution that is not overly sensitive to the omission of potentially important keywords.

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 word 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.

2 RELATED WORK

We summarize related work clustered into areas of relevance.

WWW '19 Companion, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3314197

a. Extracting and classifying entities from security forums. Recently there have been a few efforts focused on extracting entities of interest in security forums. There is a main efforts that study IP addresses and security forums [3]. In which, they focus on the spatiotemporal properties of Canadian IP addresses in forums without employing any identification and classification methods, which are the focus of our work. 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 [13, 15]. Another interesting work [18] uses a word embedding technique focusing identifying vulnerabilities and exploits. Other efforts study the users of security forums, group them into different classes, and identify their roles and social interactions [7, 16, 21].

b. Transfer learning. There is extensive literature on transfer learning [1, 2] and several good surveys [20], 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

c. NLP, Bag-of-Words, and Word Embedding techniques. Natural Language Processing is a vast field, and even the more recent approaches, such as word embedding have benefited from significant numbers of studies [10, 12, 17, 19].

3 METHODOLOGY

In this section we explain the proposed approaches to address each fold of our research.

3.1 Analyzing IP addresses

Here we present two problems regarding IP addresses in the security forums and explain our solutions to tackle them.

3.1.1 IP addresses Identification Problem. 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. For now, they are out of the scope of this research.

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. However, there is a non-trivial number of other numerical expressions, which can be misclassified as addresses such as software version and system logs.

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 described 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. We consider the frequency of the surrounding words, before and after the dot-decimal structure, in our classification.

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.

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

We saw that using *Mixed* outperforms two other feature sets.

3.1.2 IP addresses Characterization Problem. 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 3.1.3. We first identify the appropriate set of features which we discuss below. We then train a classifier.

Features sets for the Characterization problem. We consider and evaluate two 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 again to better estimate the discriminatory value of a word by considering its overall frequency.

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 our earlier work [4], with the rationale that profiling the author of a post can help us infer their intention and role and thus, improve the classification.

We saw that, by using *PostText* features on their own, we obtain slightly better results.

3.1.3 Transfer Learning with Cross-Seeding for Identification and Characterization . In both introduced 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 [2, 14]. We use the terms *source* and *target* domain to indicate the two forums with the target forum not having ground-truth available.

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.

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 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 [2]: 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 4.

c. Training a new classifier for the target forum. Having the seed information, we train the classifier directly.

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.

3.2 Analyzing threads of interest

We describe our approach toward analyzing the user-specified thread of interest first by identifying then classifying such thread.

3.2.1 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 phase, 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.

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

Step 1: The user provide a set or sets of keywords , 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. This can be defined by a threshold for each set of keywords.

Phase 2: Similarity-based selection

We propose an approach to extract additional relevant threads based on their similarity to existing relevant threads. In following steps, in which 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[12]. 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 a function of the vectors of its words and captures both the average and the maximum values of the vectors of its words[17].

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 with cosine similarity measure.

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"

3.2.2 Classifying threads of interest. We present our approach for classifying relevant threads into user-defined classes based on embeddings representation of words and also contextual features.

Given a thread, we calculate its projection in the embedding space based on the method proposed in the previous section. The embedding approaches have been used in the task of the text classification recently.

Using contextual features.

Apart from the words and their embeddings 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 provide 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 conjunction with word-based features.

4 EXPERIMENTAL RESULTS

4.1 Data Crawling and Labeling

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 1. 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.

For validating our classification method, we labeled data for each fold of the research as following:

In the IP addresses analysis, 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

	WildersSec.	OffensComm.	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 1: The basic statistics of our forums

uniformly randomly, and we used four different individuals for the labelling.

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.

In the thread of interest analysis, we need groundtruth to do both the training and the validation. We randomly selected 450 among the relevant threads from each forum as selected by the identification part. The labelling involved three manual evaluations based on the definitions and examples of the four classes, which we listed above.

4.2 Results in IP addresses analysis

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

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.

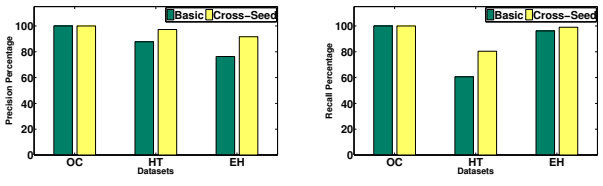
A. The IP Identification problem. In this problem, our classification approach exhibits 98% precision and 96% recall on average across all our sites, when we train with ground-truth for each forum.

Identification : 95% precision with Cross-Seeding and out performs Basic . 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 .

In figure 1, we show the results for precision and recall of cross-training using Basic (use the classifier from the source forum on the target forum) 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%. Cross-Seeding improves the precision by 8% and recall by 7% on average for the experiment shown in figure 1. The average precision increased from 88% to 95% and the average recall increased from 85% to 97%.

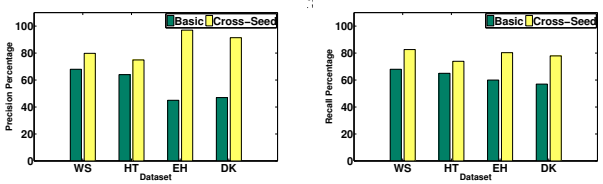
B. The IP Characterization problem. We evaluate our approach for solving the Characterization problem without per-forum training data. We can achieve 93% precision and 92% recall on average across all the forums, when we train with ground-truth for each forum.

a. Characterization : 88% precision on average with Cross-Seeding and outperforms Basic . Using OffensiveCommunity as source, and we classify WS , HackThisSite , and Darkode as shown



(a) Precision (b) Recall

Figure 1: Identification : Cross-Seeding improves both Precision and Recall. Using WildersSecurity to classify



(a) Precision (b) Recall

Figure 2: Characterization : Cross-Seeding improves both Precision and Recall. Using OffensiveCommunity as source.

in figure 2. Our Cross-Seeding approach can provide 88% precision and 82% recall on average.

we show thatby 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.

b. 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 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. We evaluated 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.

4.3 Results in threads of interest analysis

We present our experimental results and evaluation of our approach in analyzing threads of interest. We use the three forums (OffensiveCommunity , HackThisSite ,) that presented in Table 1 and the groundtruth, which we created as we explained in section 4.1.

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 300 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.

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 value.

Similarity threshold: The similarity threshold 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.

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.

Baseline methods. We evaluate our approach against three other state of the arts methods, which we briefly describe below. **Bag of Words (BOW)**: This methods users the word frequency (more accurately the TFIDF value) as its main feature [4, 8]. **Non-negative Matrix Factorization (NMF)**: This method users linear-algebra to represent high-dimensional data into low-dimensional space, in an effort to capture latent features of the data [11]. **FastText (FT)**: There is a family of methods that use the word2vec as their basis, and use a recently proposed method [9].

We present the results of our proposed methods.

Our similarity-based method is robust to the number of initial keywords. We evaluate the impact of the number of keywords to the similarity based method. In Figure 3, 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.

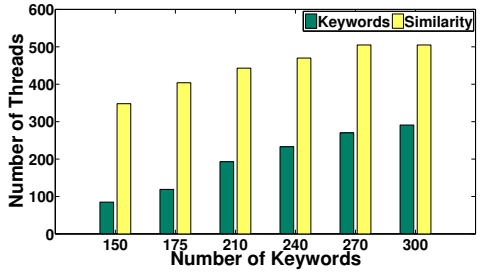


Figure 3: The robustness of the similarity approach to the initial keywords: number of relevant threads as a function of the number of keywords for OffensiveCommunity .

The features improves classification for all approaches. We briefly discusssed features in our classification section. We conduced experiments with and without these features for all four algorithms and we show the results in Figure 4 for OffensiveCommunity . Including the structural features in our classification improves the accuracy for all approaches (on average by 2.3%). The greatest beneficiary is the Bag-of-Words method whose accuracy improves by roughly 6%.

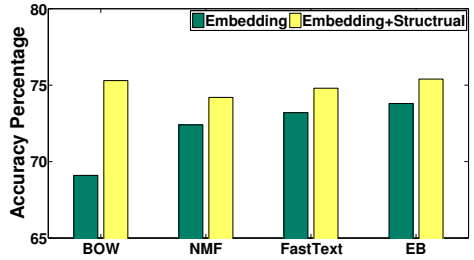


Figure 4: Classification accuracy for two different features sets in 10-fold cross validation in OffensiveCommunity .

EB outperforms the competition. Table 2 summarizes the comparison between the baseline methods and our Embedding Based approach (EB) for our three forums. EB consistently outperform other baseline method with at least 1.4 percentage point in accuracy and 0.7 percentage point in F1 score. Note that methods BOW and NMF did not assign any instances to the minority classes correctly, therefore the value of F1 score in Table 2 is reported as NA. Even though the EB outperforms other state-of-the-art methods, in the future work we are looking for possible approaches to involve the class labels in the embedding representation to improve the performance to the classification task.

Datasets	Metrics	BOW	NMF	FastText	EB
OffensComm.	Accuracy	75.3	74.2	74.8	75.4
	F1 Score	NA	NA	72.6	74.5
HackThisSite	Accuracy	66.6	72.4	69.7	74.6
	F1 Score	NA	70	65.7	72
EthicalHackers	Accuracy	59.9	58.2	59.9	61.1
	F1 Score	NA	57.2	58.9	59.5

Table 2: Classification: the performance of the four methods in classifying threads in 10-fold cross validation.

5 CONCLUSION AND FUTURE WORK

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 approaches to identify and classify IP addresses and threads of interest posted in security forums. In the future, we plan to extend our work by extracting other types of security information. Our future plans include:a) Considering class labels in embedding domain to improve the embedding performance in the classification. b) Developing a technique to automate keyword extraction from security forums to close the loop of identifying and classifying contents based on given keywords.

6 ACKNOWLEDGMENTS

I would like to thank to my PhD advisors, Professor Michalis Faloutsos for supporting me during these past four years.

REFERENCES

[1] Wenyuan Dai et al. 2007. Boosting for Transfer Learning *(ICML '07)*. New York, NY, USA, 193–200.
[2] Hal Daume III. 2007. Frustratingly Easy Domain Adaptation *(ACL '07)*.
[3] R. Frank et al. 2016. Location, Location, Location: Mapping Potential Canadian Targets in Online Hacker Discussion Forums *(EISIC '16)*.
[4] Joobin Gharibshah et al. 2017. InferIP: Extracting actionable information from security discussion forums *(ASONAM '17)*.
[5] Joobin Gharibshah et al. 2018. Mining actionable information from security forums: the case of malicious IP addresses. *CoRR* abs/1804.04800 (2018).
[6] Joobin Gharibshah et al. 2018. RIPEx: Extracting Malicious IP Addresses from Security Forums Using Cross-Forum Learning. In *PAKDD 2018*. Cham, 517–529.
[7] Thomas J Holt et al. 2012. Examining the social networks of malware writers and hackers. 6, 1 (2012), 891–903.
[8] Peng Jin et al. 2016. Bag-of-embeddings for text classification. *IJCAI International Joint Conference on Artificial Intelligence* 2016-Janua (2016), 2824–2830.
[9] Armand Joulin et al. 2017. Bag of Tricks for Efficient Text Classification. In *ACL 2017*. 427–431.
[10] Quoc Le et al. 2014. Distributed Representations of Sentences and Documents *(ICML'14)*. II–1188–II–1196.
[11] Daniel D Lee and H Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401 (oct 1999), 788.
[12] Tomas Mikolov et al. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013).

[13] Marti Motoyama et al. 2011. An Analysis of Underground Forums *(IMC '11)*. New York, NY, USA, 71–80.
[14] S. J. Pan and Q. Yang. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (2010), 1345–1359.
[15] Rebecca S. Portnoff et al. 2017. Tools for Automated Analysis of Cybercriminal Markets *(WWW '17)*. 10.
[16] S. Samtani et al. 2015. Exploring hacker assets in underground forums *(ISI '15)*.
[17] Dinghan Shen et al. 2018. Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms. In *ACL 2018*.
[18] Nazgol Tavabi et al. 2018. DarkEmbed: Exploit Prediction with Neural Language Models. In *IAAI2018*.
[19] Guoyin Wang et al. 2018. Joint Embedding of Words and Labels for Text Classification. In *ACL 2018*. 2321–2331.
[20] Karl Weiss et al. 2016. A survey of transfer learning. *Journal of Big Data* 3, 1 (28 May 2016), 9.
[21] Xiong Zhang et al. 2015. The classification of hackers by knowledge exchange behaviors. *Info. Systems Frontiers* 17, 6 (2015).