Deep-learnt features for Twitter spam detection

Xinbo Ban[1], Chao Chen[1], Shigang Liu[1], Yu Wang[2], and Jun Zhang[1]

Abstract— **Twitter spam has become one of the most critical problems in recent years. Despite the efforts of researchers and security companies, the growing number of spam is not stopping. Machine learning is a very popular technology in network security and is also used for spam detection. An important step of applying machine learning for Twitter spam detection is feature engineering. Existing works mainly use URL based features, meta-data based features and social relation based features to detect spam tweets. All of the above mentioned works require human effort to extract features. More recently, deep learning has shed its light on automated feature engineering in extracting features from text. In this paper, we propose a new feature engineering mechanism based on a deep neural network trained using Bi-LSTM. We name the extracted features "deep-learnt features". We compare our feature set with word2vec features and statistical features in the experimental evaluation. The results show that machine learning models trained using deep-learnt features can detect Twitter spam more accurately than models trained using word2vec features and statistcal features.**

I. INTRODUCTION

Twitter, which is one of the most popular social networks, broadcasts over 400 million tweets produced by over 200 million Twitter users per day [17], [5]. It provides a platform for users to exchange messages and share ideas all over the world. Along with the convenience provided by Twitter, unfortunately, thre are a lot of spam message in this platform, which leads poor user online experience and even results in malicious activities. Spammers post unsolicited tweets with malicious links that direct to the websites designed for phishing, malware downloading and scams [1]. If one victim was lured to click the linked embedded in spam tweets, his account could be hacked or personal information being leaked. In September 2014, the Internet of New Zealand crashed due to the spread of twitter spam which contains malware downloading links [20]. Spammers used Hollywood stars' photos to attract victims to click the link, and download the malware. The malware was installed in victims' machines and performed a DDoS attack. Spam is a severe problem on Twitter, and is of great urgency to be solved.

Twitter, along with researchers, are working hard to combat spammers to maintain the cleanliness and security of Twitter. For example, Twitter employs Bot Maker, which checks whether the embedded tweet has a spam URL, to reduce spam on Twitter. [13]. Another example is Trend Micro's Web Reputation Technology system, which is a black listing service used to filter spam URLs for its users[19].

However, the time lag of blacklist filtering services usually leads to failure in protecting users from new spam URLs [12]. According to [24], 90% victims might have already visited the malicious link of new spam before blacklist service blocked it . To deal with the lagging issue of blacklisting-based spam detection mechanism, many machine learning techniques are applied for increasing the accuracy, the speed of detection while reducing manual work. Machine learning algorithm learns the knowledge through ground-truth and predicts new knowledge through the learnt knowledge. A number of works have applied machine learning techniques to detect Twitter spam [1], [9], [14]. An important step of applying machine learning for Twitter spam detection is feature engineering. Existing works mainly use URL based features [22], [14], meta-data based features [1], [25], social relation based features [28], [21] to detect spam tweets. All of the above mentioned works require human effort to extract features. More recently, deep learning has shed its light on automated feature engineering in extracting features from text [27]. Some preliminary works using word2vec to extract features.

In this paper, we propose a new feature engineering method for Twitter spam detection based on deep learning. Firstly, we convert textual tweets into the dense vector using Word2vec. Then, BiLSTM is used to automatically extract deep-learnt features to gather rich information from tweets. In the experimental evaluation, we also compare the performance of Twitter spam detection based on three kinds feature sets, which are statistical features, word2vec features, and deep-learnt features. Word2vec features are extracted from tweets by using Google's word2vec algorithm. Deep-learnt features are extracted from the deep neural network trained by Bi-LSTM. We run 100 times experiments for each dataset to compare the results from three different feature sets on six machine learning algorithms.

The rest of this paper is organised as follows. II reviews recent works of Twitter spam detection and machine learning. III presents our data collection and processing. We also introduce the word2vec based feature engineering and our proposed deep learning scheme for feature representation learning in this section. IV describes experimental settings and evaluation of three kinds of feature sets to detect Twitter spam. Finally, V concludes this work.

II. RELATED WORK

Spam is a critical problem on Twitter. Grier et al. found that 8% of crawled unique URLs obtained from tweets were

[1]X. Ban, S. Liu, C. Chen and J. Zhang is with the School of Software and Electrical Engineering, Swinburne University of technology, Australia. e-mail: 101918874@student.swin.edu.au, {shigangliu, chaochen, cchua}@swin.edu.au.

[2]Y. Wang is with the School of Computer Science, Guangzhou University, China e-mail: yuwang@gzhu.edu.cn.

spam in 2010 [12]. It means that 2 million out of 25 million URLs contained malicious links. Thomas et al. suggested that 80 million out of 1.8 billion were spam in 2011 [22]. Furthermore, it was reported by Chen et al., who collected and analysed over 600 million tweets which all contained URL links, that 1% of their collected tweets were spam [4].

Blacklist techniques were firtly used to filter the spam tweets. For example, Malicious URLs were detected by Oliver et al. through blacklist techniques [19]. Moreover, Ma et al. proposed a lightweight blacklisting method that consumes lower computing resource than other approaches [18]. However, Blacklist relies on the large amount of manually labelling work due to the growing number of spamming URLs. In addition, due to the time lag, it lacks the feasibility of detecting Twitter spam at an early stage [23].

Some early works applied heuristic rules to detect Twitter spam. Such as in [29], keyword detection and username pattern match were used to filter spam. However, since spammers are evolving to use complex strategies to generate spam [6], the basic string pattern match rules will miss a lot of spam tweets.

As a result, machine learning techniques are adopted by researchers for Twitter spam detection. The most important aspect of a successful machine learning model is the feature set. Various features are used in machine learning based Twitter spam detection approaches, such as URL based features, meta-data (user account and tweet information) based features, and social graph based features. Thomas et al. used features extracted from URLs such as path tokens and domain tokens, along with features obtained from domain information, DNS information and landing pages [22]. Meta-data based features were widely adopted by research community. Wang et al. proposed a method that was based on Bayesian model for Twitter spammers detection [25] by using user account based features. Benevenuto et al. introduced a Support Vector Machine algorithm based approach to detect both spams and spammers [1] based on account and teweet based features. However, it is easy to evade the features used in the above works by continuously posting tweets, buying off followers, and mixing normal and spam tweets together. Accordingly, robust feature proposed by researchers are used to avoid feature evasion based on the social graph. Song et al. combined these complex feature with the basic feature set and successfully improved the performance of a few classifiers, which is about 99% True Positive Rate and nearly 1% False Positive Rate [21]. In [28] Several robust spam features reported by Yang et al. include Bidirection Links Ratio, Betweebbess Contralisy and Local Clustering Coeffi-cient. Their work demonstrated that robust feature set can achieve outstanding performance compared with existing approaches.

More recently, deep learning has been applied for automated feature engineering in Twitter spam detection, as it requires little human effort to extract the features. [27] used word2vec to extract the features from tweet. Feng et al. also used word2vec embeddings feature set to train a Convolutional neural network to detect spam [8]. Differnt to the above mentioned works, we use Bi-LSTM for automated feature learning in this paper.

III. Data Processing and Feature Engineering

A. *Description of experimental Dataset*

One of the most important things in the area of Twitter spam detection is labelled dataset. In this paper, we collected tweets which contain URLs through Streaming API of Twitter during ten consecutive days. Although spam possibly dose not contains URLs, most spam tweets are embedded with URLs [7], [11]. After we have manually checked hundreds of spam tweets only a very limited number of tweets are considered as spam without URLs. Furthermore, it is more convenient for spammers to use tweets embedded URLs to redirect victims to their expected sites for some purposes such as scams, phishing and malware downloading [30]. Thus, we only use the tweets embedded URLs in this paper.

Currently, there are two ways to label the dataset for building ground-truth, i.e. blacklist filtering and manual inspection. Under the consideration of the real situation, manual inspection can not generate a large number of training samples, and it consumes too much human effort. Although labelling task can be helped with HIT(human intelligence task) websites, the results sometimes are questionable and it is costly as well [2]. Some researchers labelled spam tweets by applying provided blacklisting services, such as URIBL [10] and Google Safe Browsing. However, the limitation of APIs of these services makes it is not realistic to label a large number of spam tweets.

Trend Micro's Web Reputation Technology, that records a large dataset of URL reputation, is applied to identify which tweet is spam [4]. The reputation URL records in this system are obtained from opt-in URL filtering records of there customers. WRT system devotes to collect and analyse the latest and most popular URLs and then it will real-time protection for its users. Therefore, we can identify whether a URL is malicious by matching with the records in WRT system. The tweets, that contain identified malicious URLs, are deemed spam. In the testing report of AV Comparatives, it is stated that WRT system is reliable because of its 100% protection rate. Moreover, We have done a manual inspection of hundreds of tweets to confirm the reliability of WRT system. In our collected dataset, we labelled 1 million non-spam tweets and 1 million spam tweets, resulting 100k non-spam and 100k spam tweets per day. We randomly extract 50,000 non-spam tweets and 50,000 spam tweets. Two deep learning algorithms are used to extract features, word2vec and our approach (Bi-LSTM). We name them word2vec features and deep-learnt features. After we obtain the word2vec feature and deep-learnt feature, we used 50 thousand features of non-spam and 2632 features of spam as the 1:19 of imbalance ration (IR = non-spam tweets/ non-spam tweets), which is the same as in [3]. In order to conduct a comparative study from different angles, we also used the dataset published in [3]. This dataset can be retrieved from

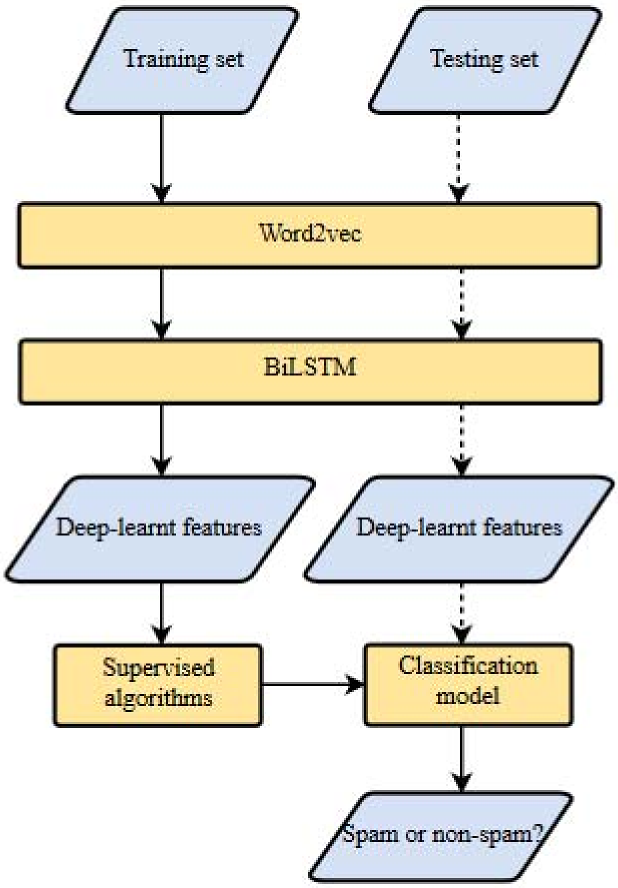

Fig. 1. Deep learning based framework

NSClab[1].

In next section, we will detail the deep learning framework for deep-learnt features extraction.

B. Feature Engineering

As mentioned before, we employ two popularly used approaches for automated feature learning from the text data. We consider each Tweet as a text document.

- Word2vec: this is widely used for large text datasets in natural language process, and it has been also successfully used in many research fields such as network security [27] and biomedical text processing [26]. Firstly, We remove the blank lines and non-alphabets, which means each processed tweet only contains spaces and words. Then, we employ Continuous Bag of Words (CBOW) to build the dictionary that includes every word exist in the dataset and its uniquely converted vectors. Through this dictionary, we calculate vector of each tweet using vector calculation method. The size of dimensionality is set to 100 because we found the supervised model performs well with 100 dimensions.
- BiLSTM: this model was previous used in [15] for feature selection. In this study, we re-use the idea from [15] in order to investigate the performance of deep learning in Twitter spam detection. We firstly train a deep neural network using Bi-LSTM, then we extract deep-learnt features through the last second layer of Bi-LSTM. Specifically, there are five layers in the architecture of our trained LSTM-based network. An embedding layer maps each element of the tweet to a dense vector of proper dimensionality as the first layer. With embedding layer, discrete inputs, that are words in tweets, are converted to a dense vector that is non-sparese. We employ CBOW to process the tokenized sequences in the embedding layer. The second and third layer contains two Bi-LSTM networks (i.e., two LSTM networks in a bi-directional form with each LSTM contains 100 LSTM units). The BiLSTM network is able to generate higher-level abstractions through learning dependent relationships between elements. The fourth layer is the global max pooling layers used to strengthen the signals. Then, the last two dense layers, which are "tanh" and "sigmoid" activation functions, are used to converge a 200 dimensionality output to a single dimensionality. In the period of the training phase, we try to minimise the loss function that is a binary cross entropy function in our study. In this study, the dimensionality of deep-learnt features is the same as word2vec features, that is 100-dimension vector. We choose the the features from the last second layer as deep-learnt feature representations.

The dataset with learnt features will be used to train some supervised classifiers. We then generate deep-learnt features for testing data, and test the performance of the trained classification model. The framework is shown in Figure 1.

IV. Experimental Evaluation

A. Experiments setup

The experiments are two-fold: feature learning and model building. **1) Feature learning:** we use Keras (version 2.0.8) to implement Bi-LSTM network with Tensotflow (version 1.3.0) back-end. For Word2vec Embedding, we use gensim package (version 3.0.1) with default setting except "size" (dimensionality of the word vectors) and "min_count" (words with total lowest frequency), which are 100 and 1. The computational system is a server running CentOS Linux 7 with two Physical Intel(R) Xeno(R) E5-2690 v3 2.6GHz CPUs and 96GB RAM. **2) Model buidling:** we use Weka for modeling building with the default setting.

We choose 6 machine learning algorithms: C4.5 decision tree (J48), random forest (RF), k-nearest neighbour (KNN), Naavie Bayes (NB), support vector machine (SVM), and neural network (NN), to perform the evaluation study. The dataset is randomly split into 50% training, 25% for evaluation and the rest 25% for testing. We repeat this for 100 times and report the average results.

B. Performance Metrics

To evaluate the performance of classifiers in various environments, we adopted the metrics such as Precision,

[1] http://nsclab.org/nsclab/resources/

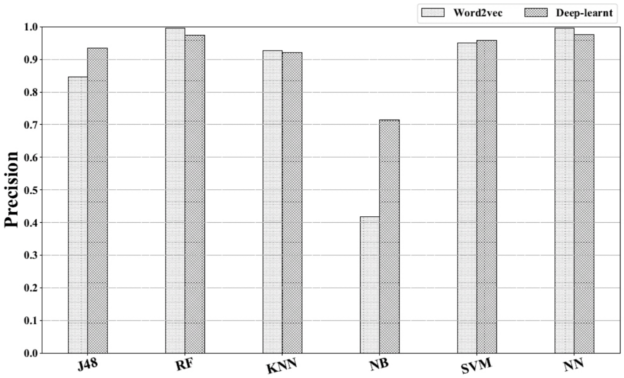

Fig. 2. Precision

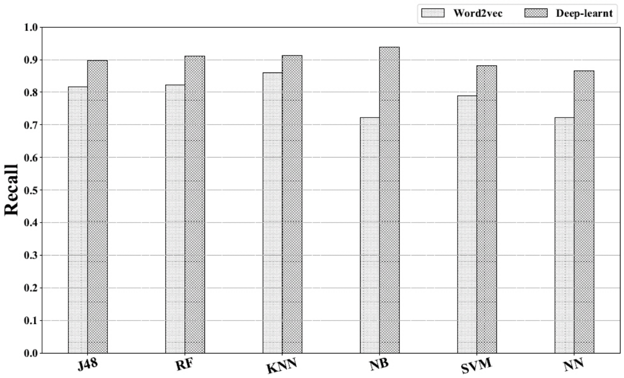

Fig. 3. Recall

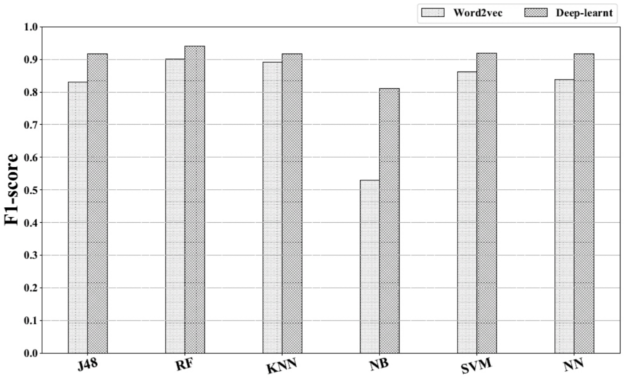

Fig. 4. F1-score

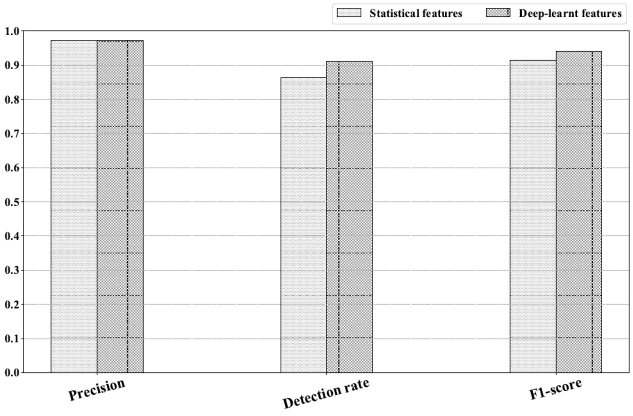

Fig. 5. RF

Recall, F1-score, generally used in information retrieval [16]. We treat Twitter spam as positive class and non-spam as a negative class in this study. These performance metrics can be calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

C. Evaluation 1: comparison based on word2vec features and deep-learnt features

Figure 2 shows the average results of spam detection based on word2vec features and deep-learnt features with 6 machine learning algorithms. We can see the comparison between word2vec features and deep-learnt features evaluated on a same algorithm. We can observe that precision of all algorithms are higher than 80% except NB for both feature sets. Figure 3 and Figure 4 show similar results, most of the algorithms can achieve over 80% for recall and F1-score. More importantly, we can see that the results based on deep-learnt features can further improve the performance of 10% to 30% higher in terms of Recall and F1-score. The reason is that deep-learnt features learned more essential details than word2vec features, which make classifiers to predict more accurately.

It can be reported that RF is the classifier which has the best performance among all machine learning algorithms in this study. Both types of features can achieve over 80% precision, recall, and F1-score. NB is the algorithm that performs worst. Except for recall, precision and F1-score of NB are around 50% for word2vec features and 70% for deep-learnt features. The performance of other algorithms are somewhere in between.

Generally speaking, RF could help to detect more spam based on deep-learnt features than word2vec features. Although RF based on word2vec features achieve nearly 99% of precision, which is about 2 percent higher than RF on deep-learnt features, its Recall is 10 percent less and the F1-score is about 5 percent less than RF based on deep learnt features. RF based on deep-learnt features achieves 96% precision and 91% recall, resulting a better F1-score of 97%. RF based on word2vec features reports less true negatives with low recall although it has high precision. Thus, it sacrifices true positive for higher precision which means it missed many spams. However, the performance of deep-learnt features is more acceptable than word2vec features due to overall consideration of both precision and recall.

D. Evaluation 2: comparison based on statistical features and deep-learnt features

We also compare the performance of multiple classifiers on statistical features [3] with our deep-learnt features.

Figure 5 shows the average results of Precision, Recall, F1-score of Random Forest trained with statistical features and deep-learnt features. Both RF models trained using statistical features and deep-learnt features can achieve over 96% in terms of precision. However, the model trained using deep-learnt features can achieve more than 90% recall and 93% F1-score, which is way better than that trained using statistical features. Therefore, we conclude that classification model based on deep-learnt features can help to detect more spam than that based on word2vec features. It is due to the reason that deep neurual networks like Bi-LSTM can learn the high representations behind the text of tweets.

V. Conclusions

In this paper, we propose a new automated feature engineering mechanism to extract features for Twitter spam detection. We train a deep neural network (i.e. Bi-LSTM) using labelled dataset, and extract features from the hidden layers of it to represent tweets. In order to evaluate the performance of deep-learnt features, we also compare our feature set with word2vec features and statistical features. In terms of F1-score, model trained on deep-learnt features can be 5% to 25% higher than that trained on word2vec features, depending on which algorithm is used to train the model. It is also 3% higher than model traind on statistical features. As a result, a machine learning model trained using the feature vector representation learnt from deep neural networks (Bi-LSTM in our work) can detect Twitter spam more accurately than models trained using word2vec features and statistcal features.

References

[1] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammer on twitter. In *Seventh Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, July 2010.

[2] C. Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 675–684, New York, NY, USA, 2011. ACM.

[3] C. Chen, Y. Wang, J. Zhang, Y. Xiang, W. Zhou, and G. Min. Statistical features-based real-time detection of drifted twitter spam. *IEEE Transactions on Information Forensics and Security*, 12(4):914–925, 2017.

[4] C. Chen, J. Zhang, X. Chen, Y. Xiang, and W. Zhou. 6 million spam tweets: A large ground truth for timely twitter spam detection. In *IEEE ICC 2015 - Communication and Information Systems Security Symposium (ICC'15 (11) CISS)*, pages 8689–8694, London, United Kingdom, June 2015.

[5] C. Chen, J. Zhang, Y. Xiang, and W. Zhou. Asymmetric self-learning for tackling twitter spam drift. In *Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on*, pages 208–213. IEEE, 2015.

[6] C. Chen, J. Zhang, Y. Xiang, W. Zhou, and J. Oliver. Spammers are becoming "smarter" on twitter. *IT Professional*, 18(2):66–70, Mar 2016.

[7] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna. Compa: Detecting compromised accounts on social networks. In *Annual Network and Distributed System Security Symposium*, 2013.

[8] B. Feng, Q. Fu, M. Dong, D. Guo, and Q. Li. Multistage and elastic spam detection in mobile social networks through deep learning. *IEEE Network*, 32(4):15–21, July 2018.

[9] H. Gao, Y. Chen, K. Lee, D. Palsetia, and A. Choudhary. Towards online spam filtering in social networks. In *NDSS*, 2012.

[10] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, IMC '10, pages 35–47, New York, NY, USA, 2010. ACM.

[11] H. Gao, Y. Yang, K. Bu, Y. Chen, D. Downey, K. Lee, and A. Choudhary. Spam ain't as diverse as it seems: throttling osn spam with templates underneath. In *Proceedings of the 30th Annual Computer Security Applications Conference*, pages 76–85. ACM, 2014.

[12] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @spam: the underground on 140 characters or less. In *Proceedings of the 17th ACM conference on Computer and communications security*, CCS '10, pages 27–37, New York, NY, USA, 2010. ACM.

[13] R. Jeyaraman. Fighting spam with botmaker. *Twitter Engineering Blog*, August 2014.

[14] S. Lee and J. Kim. Warningbird: A near real-time detection system for suspicious urls in twitter stream. *IEEE Transactions on Dependable and Secure Computing*, 10(3):183–195, 2013.

[15] G. Lin, J. Zhang, W. Luo, L. Pan, and Y. Xiang. Poster: vulnerability discovery with function representation learning from unlabeled projects. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 2539–2541. ACM, 2017.

[16] S. Liu, Y. Wang, C. Chen, and Y. Xiang. An ensemble learning approach for addressing the class imbalance problem in twitter spam detection. In *Australasian Conference on Information Security and Privacy*, pages 215–228. Springer, 2016.

[17] S. Liu, J. Zhang, and Y. Xiang. Statistical detection of online drifting twitter spam. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, pages 1–10. ACM, 2016.

[18] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Learning to detect malicious urls. *ACM Trans. Intell. Syst. Technol.*, 2(3):30:1–30:24, May 2011.

[19] J. Oliver, P. Pajares, C. Ke, C. Chen, and Y. Xiang. An in-depth analysis of abuse on twitter. Technical report, Trend Micro, 225 E. John Carpenter Freeway, Suite 1500 Irving, Texas 75062 U.S.A., September 2014.

[20] C. Pash. The lure of naked hollywood star photos sent the internet into meltdown in new zealand. *Business Insider*, September 2014.

[21] J. Song, S. Lee, and J. Kim. Spam filtering in twitter using sender-receiver relationship. In *Proceedings of the 14th international conference on Recent Advances in Intrusion Detection*, RAID'11, pages 301–317, Berlin, Heidelberg, 2011. Springer-Verlag.

[22] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song. Design and evaluation of a real-time url spam filtering service. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, SP '11, pages 447–462, Washington, DC, USA, 2011. IEEE Computer Society.

[23] K. Thomas, C. Grier, and V. Paxson. Adapting social spam infrastructure for political censorship. In *Proceedings of the 5th USENIX conference on Large-Scale Exploits and Emergent Threats*, LEET'12, pages 13–13, Berkeley, CA, USA, 2012. USENIX Association.

[24] K. Thomas, C. Grier, D. Song, and V. Paxson. Suspended accounts in retrospect: an analysis of twitter spam. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, IMC '11, pages 243–258, New York, NY, USA, 2011. ACM.

[25] A. H. Wang. Don't follow me: Spam detection in twitter. In *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, pages 1–10, 2010.

[26] L. Wolf, Y. Hanani, K. Bar, and N. Dershowitz. Joint word2vec networks for bilingual semantic representations. *Int. J. Comput. Linguistics Appl.*, 5(1):27–42, 2014.

[27] T. Wu, S. Liu, J. Zhang, and Y. Xiang. Twitter spam detection based on deep learning. In *Proceedings of the Australasian Computer Science Week Multiconference*, ACSW '17, pages 3:1–3:8, New York, NY, USA, 2017. ACM.

[28] C. Yang, R. Harkreader, and G. Gu. Empirical evaluation and new design for fighting evolving twitter spammers. *Information Forensics and Security, IEEE Transactions on*, 8(8):1280–1293, 2013.

[29] S. Yardi, D. Romero, G. Schoenebeck, and D. Boyd. Detecting spam in a twitter network. *First Monday*, 15(1-4), January 2010.

[30] X. Zhang, S. Zhu, and W. Liang. Detecting spam and promoting campaigns in the twitter social network. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 1194–1199, 2012.